IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 2, 2018, accepted January 9, 2019, date of publication January 18, 2019, date of current version April 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2893531

A Trace Lasso Regularized Robust Nonparallel Proximal Support Vector Machine for Noisy Classification

WEI-JIE CHEN[1,2], KAI-LI YANG[3], YUAN-HAI SHAO[4], YU-JUAN CHEN[5], JU ZHANG[1], AND JING-JING YAO[1]

[1]Zhijiang College, Zhejiang University of Technology, Hangzhou 310024, China
[2]Centre for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW 2007, Australia
[3]College of Science, Zhejiang University of Technology, Hangzhou 310024, China
[4]School of Economics and Management, Hainan University, Haikou 570228, China
[5]School of Data Sciences, Zhejiang University of Finance and Economics, Hangzhou 310018, China

Corresponding author: Wei-Jie Chen (wjcper2008@126.com) and Yu-Juan Chen (chenyj@zufe.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61603338, Grant 11871183, Grant 61866010, Grant 61703370, and Grant 11426202, in part by the Natural Science Foundation of Zhejiang Province under Grant LQ17F030003, Grant LY17F030015, Grant LY15F030013, and Grant LQ13F030010, in part by the Natural Science Foundation of Hainan Province under Grant 118QN181, in part by the Scientific Research Fund of the Zhejiang Provincial Education Department under Grant Y201840622, and in part by the Foundation of China Scholarship Council under Grant 201708330179.

ABSTRACT Generalized eigenvalue proximal support vector machine (GEPSVM) and its improvement IGEPSVM are excellent nonparallel classification methods due to their excellent generalization. However, all of them adopt the square L_2-norm metric to implement their empirical risk or penalty, which is sensitive to noise and outliers. Moreover, in many real-world learning tasks, it is a significant challenge for GEPSVMs when the data appears highly correlated. To alleviate the above issues, in this paper, we propose a novel trace lasso regularized robust nonparallel proximal support vector machine (RNPSVM) for noisy classification. Compared with GEPSVMs, our RNPSVM enjoys the following advantages. First, the empirical risk of RNPSVM is implemented by the robust L_1-norm metric with a maximum margin criterion. Namely, it aims to maximize the L_1-norm inter-class distance dispersion while minimizing the L_1-norm intra-class distance dispersion simultaneously. Second, to capture the sparsity and the underlying correlation of data, a trace lasso (adaptive norm-based training data) is further introduced to regularize RNPSVM. Third, an iterative algorithm is designed to solve the maximization optimization problem of RNPSVM, whose convergence is guaranteed theoretically. The extensive experimental results on both synthetic and real-world noisy datasets demonstrate the effectiveness of RNPSVM.

INDEX TERMS Support vector machines, L_1-norm, regularization, robustness, classification algorithms.

I. INTRODUCTION

Support Vector Machine (SVM) [1], [2] is an excellent maximum-margin learning method for pattern recognition, which originates from statistical learning theory. The central idea of SVM is to construct an optimal separating hyperplane by optimizing the soft margin using the hinge loss and a regularization term, such that the margin between two parallel support hyperplanes is maximized, while the instances are pushed as far as possible away from this margin. With the help of structural risk minimization (SRM) principle, the upper bound on the generalized error of SVM is guaranteed theoretically. Formally, the optimization problem of SVM for finding a max-margin separation can be stated as a quadratic programming problem (QPP), whose global optimum can be achieved. During the last decades, SVM has already achieved good performances in various practical application domains, such as bioinformatics [3], [4], computer vision [5], fault diagnosis [6], and so on [7], [8].

However, there are two major bottlenecks in SVM. One is the training procedure of SVM needs to optimize a larger QPP, and another is SVM cannot capture the underlying

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Callico.

distribution of heterogeneous data well, such as the "XOR" problem. To alleviate above issues, so far, many efficient optimization algorithms and tools have been proposed to improve the learning efficiency of SVM, e.g. SMO [9], SVMlight [10], LIBSVM [11], LIBLINEAR [12] and Pegasos [13]. On the other hand, a series of excellent SVM models has been put forward, such as least square SVM (LSSVM) [14], proximal SVM (PSVM) [15], smooth SVM (SSVM) [16] and Lagrangian SVM [17].

Different from the parallelism condition in the original SVM, recently, a novel nonparallel hyperplane learning (NHL) paradigm has been proposed [18]–[20]. The goal of NHL is to seek an optimal nonparallel hyperplane for each class, such that each hyperplane is closest to its own class while as far as possible from the other class. The pioneering work of NHL can be dated back to the generalized eigenvalues proximal support vector machine (GEPSVM), which was proposed by Mangasarian and Wild [19]. It relaxes the requirement of hyperplanes generated by SVM should be parallel, and aims to construct a pair of nonparallel proximal hyperplanes by solving two generalized eigenvalue problems instead of a large scale QPP. The results in [19] demonstrate the effectiveness of GEPSVM, especially on the "XOR" problem.

Recently, the advantages of GEPSVM has brought many efforts to its various improvements. Ye and Ne [21] presented a new method via singular value decomposition (SVD) to overcome the singularity problem that may encounter in GEPSVM. To improve the stability of GEPSVM, Guarracino *et al.* [22] proposed a *regularized general eigenvalue classifier* (ReGEC) by introducing a new regularization term. Subsequently, Shao *et al.* [23] proposed an *improved generalized eigenvalue SVM* (IGEPSVM) according to the maximum margin criterion [24]. Specifically, IGEPSVM reformulates the optimization problems of GEPSVM by replacing the "ratio" formulations with "difference" ones, and an extra meaningful parameter is introduced. As a result, IGEPSVM owns the better generalization than GEPSVM theoretically. Additionally, GEPSVM has also been extended to deal with the semi-supervised learning [25] and multi-view learning [26]. For more related works on extensions of GEPSVM and NHL, we refer the readers to [27]–[36].

However, it is worth noting that all the aforementioned GEPSVMs are utilized the L_2-norm metric to measure their loss function or penalty, resulting in sensitivity to outliers. The reason is that the L_2-norm will magnify the effect of outliers by square operation, which leads to the bias classification result. Statistically speaking, the L_1-norm is usually deemed as a more robust way than the L_2-norm, since the absolute value operation in the L_1-norm will mitigate the impact of outliers compared with the L_2-norm [37]–[42]. Therefore, to improve the model robustness, Li *et al.* [43] proposed a *L_1-norm nonparallel proximal support vector machine* (L1NPSVM) for noisy classification. A gradient ascending (GA) iterative algorithm was further proposed to solve the L_1-norm ratio optimization problem of L1NPSVM. However, both the need of the non-convex surrogate function and the difficult selection of step-size in GA may not guarantee the optimum solution.

On the other hand, in practice, difference datasets may have difference correlation structure. However, the existing GEPSVMs ignore this underlying correlation information at all, which may degrade their performance. Although the L_2-norm regularization is considered to control their model complexity, such regularizer cannot automatically satisfy the data distribution [44]–[47]. That is, the L_2-norm regularizer is blind to exact correlation structure of data, since it is not uniformly required for all features and is not adaptive for correlation of features.

The above analysis motives us to improve the performance of GEPSVM [19], [23], and propose a novel robust nonparallel proximal SVM via trace lasso for noisy classification, termed as RNPSVM. Compared with the existing GEPSVMs, our RNPSVM owns the following merits:

1) The empirical risk of RNPSVM is implemented by the L_1-norm measurement with maximum margin criterion. Namely, it aims to maximize the L_1-norm inter-class distance dispersion while minimize the L_1-norm intra-class distance dispersion simultaneously. The L_1-norm formulation makes RNPSVM enjoy the robustness to outliers.
2) To further improve the performance, a trace lasso is introduced to regularize RNPSVM by considering the sparsity and correlation of data simultaneously. Trace lasso is also known as an adaptive norm, which can automatically balance the L_1-norm and L_2-norm regularization based on the data distribution. To our best knowledge, RNPSVM is the first NHL classifier that considers the data correlation, which is a useful extension of GEPSVM.
3) Our RNPSVM can avoid the singularity problem effectively, which may encounter such problem in GEPSVM by solving eigenvalue problems.
4) An efficient iterative algorithm is designed to solve the corresponding L_1-norm maximization problem with trace lasso regularization, whose convergence is guaranteed theoretically.
5) Last but not the least, extensive experimental results on both synthetic and real-world noisy datasets demonstrate that, compare with its peers, our RNPSVM can effectively suppress the impact of the outliers and achieve better performance.

The rest of this paper is organized as follows: Section II briefly dwells on GEPSVM and IGEPSVM methods. Section III proposes our RNPSVM approach, and the feasibility of algorithmic procedure is also theoretically analyzed. Experimental results are described in Section IV and concluding remarks are given in Section V.

II. BACKGROUNDS

In this section, we first describe the notations used throughout the paper. Then, briefly introduce GEPSVM [19] and IGEPSVM [23].

A. NOTATIONS

Upper (lower) bold face letters are used for matrices (column vectors). All vectors will be column vectors unless transformed to row vectors by a prime superscript $(\cdot)'$. A vector of zeros of arbitrary dimensions is represented by $\mathbf{0}$. In addition, $\boldsymbol{e}$ is denoted as a vector of ones and $\boldsymbol{I}$ as an identity matrix of arbitrary dimensions.

Consider a binary classification problem in the n-dimensional space $\mathbb{R}^n$. Denote the set of training data as

$$\mathcal{T} = \{(\boldsymbol{x}_i, y_i) | 1 \le i \le m\} \in (\mathcal{X} \times \mathcal{Y})^m, \tag{1}$$

where $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^n$ is an input instance with its associated label $y_i \in \mathcal{Y} = \{+1, -1\}$. For simplification, denote $\mathcal{I}_k$ as the set of indices such that if an instance $\boldsymbol{x}_i$ belongs to the k-th class, i.e., $i \in \mathcal{I}_k$, where $k = 1$ or 2 corresponds to the positive or negative class. Otherwise, $j \in \mathcal{I}_{\bar{k}}$ means that an instance $\boldsymbol{x}_j$ does not belong to the k-th class. Moreover, suppose that matrix $\boldsymbol{A} = \{\boldsymbol{x}_i\}_{i \in \mathcal{I}_1}$ with size of $m_1 \times n$ represents instances of class 1 (class $+1$), while matrix $\boldsymbol{B} = \{\boldsymbol{x}_j\}_{j \in \mathcal{I}_2}$ with size of $m_2 \times n$ represents instances of class 2 (class -1). Denote $\boldsymbol{X} = \left[\boldsymbol{A}' \; \boldsymbol{B}'\right]'$ as all the training instances, where $m_1 + m_2 = m$. Furthermore, define a diagonal conversion operation $\text{Diag}(\cdot)$ that converts a vector $\boldsymbol{d} \in \mathbb{R}^n$ into a diagonal matrix $\boldsymbol{D} \in \mathbb{R}^{n \times n}$, whose diagonal elements $D_{ii} = d_i$.

B. GEPSVM

GEPSVM [19] is an excellent classifier for classification, which relaxes the requirement of hyperplanes generated by SVM should be parallel and attempts to seek a pair of non-parallel proximal hyperplanes

$$\boldsymbol{w}_k'\boldsymbol{x} + b_k = 0, \quad k = \{1, 2\} \tag{2}$$

where $\boldsymbol{w}_k \in \mathbb{R}^n$ is the normal vector, and $b_k \in \mathbb{R}$ is the bias term. The optimization goal of GEPSVM is that each hyperplane in (2) should be closest to its class while as far as possible from the other class simultaneously. GEPSVM utilizes the "ratio" criterion to implement its empirical risk, leading to the following optimization problem for the k-th class hyperplane

$$\min_{(\boldsymbol{w}_k, b_k) \neq \mathbf{0}} \frac{\sum_{i \in \mathcal{I}_k} \|\boldsymbol{w}_k'\boldsymbol{x}_i + b_k\|_2^2}{\sum_{j \in \mathcal{I}_{\bar{k}}} \|\boldsymbol{w}_k'\boldsymbol{x}_j + b_k\|_2^2}. \tag{3}$$

For sake of simplicity, we augment the input space $\mathcal{X}$ from $\mathbb{R}^n$ to $\mathbb{R}^{n+1}$: $\tilde{\boldsymbol{x}} = [\boldsymbol{x}' \; 1]'$ and define $z_k = \left[\begin{smallmatrix} w_k \\ b_k \end{smallmatrix}\right]$, then the corresponding proximal hyperplanes (2) can be expressed as $z_k'\tilde{\boldsymbol{x}} = 0$.

Note that, solving problem (3) may suffer the singularity problem. Therefore, a Tikhonov regularization term is further introduced in (3) to improve the stability of GEPSVM. It yields

$$\min_{z_k \neq \mathbf{0}} \frac{z_k'\left(\sum_{i \in \mathcal{I}_k} \tilde{\boldsymbol{x}}_i\tilde{\boldsymbol{x}}_i'\right) z_k + \delta\|z_k\|_2^2}{z_k'\left(\sum_{j \in \mathcal{I}_{\bar{k}}} \tilde{\boldsymbol{x}}_j\tilde{\boldsymbol{x}}_j'\right) z_k}, \tag{4}$$

where δ is a small positive parameter. The above minimization problem (4) is exactly Rayleigh quotient [19], whose solution can be obtained via the following generalized eigenvalue problem

$$\left(\sum_{i \in \mathcal{I}_k} \tilde{\boldsymbol{x}}_i\tilde{\boldsymbol{x}}_i' + \delta\boldsymbol{I}\right) z_k = \lambda \left(\sum_{j \in \mathcal{I}_{\bar{k}}} \tilde{\boldsymbol{x}}_j\tilde{\boldsymbol{x}}_j'\right) z_k. \tag{5}$$

The optimal solution to problem (4) is the eigenvector corresponding to smallest eigenvalue of problem (5). Once the solution z_k is obtained, hyperplanes (2) of GEPSVM are constructed. For an unseen instance $\boldsymbol{x} \in \mathbb{R}^n$, its class label is assigned according to which of the hyperplanes (2) it is closer to, i.e.,

$$\text{Class } k = \arg\min_{k=1,2} \frac{|\boldsymbol{w}_k'\boldsymbol{x} + b_k|}{\|\boldsymbol{w}_k\|_2}, \tag{6}$$

where $|\cdot|$ is the absolute value.

C. IMPROVEMENT OF GEPSVM (IGEPSVM)

It is observed from (3) that GEPSVM utilizes the "ratio" criterion to measure the differences of distances between instances of two classes and k-th class hyperplane, which may encounter the possible singularity. To alleviate this, in light of the maximum margin criterion [24], Shao *et al.* [23] proposed an improved version of GEPSVM, termed as IGEPSVM. Specifically, its empirical risk is implemented in the "minus" form instead of "ratio", which yields the following optimization problem for the k-th class hyperplane

$$\min_{z_k} \nu z_k'\left(\sum_{i \in \mathcal{I}_k} \tilde{\boldsymbol{x}}_i\tilde{\boldsymbol{x}}_i'\right) z_k - z_k'\left(\sum_{j \in \mathcal{I}_{\bar{k}}} \tilde{\boldsymbol{x}}_j\tilde{\boldsymbol{x}}_j'\right) z_k, \tag{7}$$

where $\nu > 0$ is a penalty parameter that determines the trade-off between the two loss terms in (7) and $k = \{1, 2\}$. Compared with GEPSVM, this meaningful parameter ν allows IGEPSVM to have a bias factor for different data class.

Let $\boldsymbol{A}_k = \{\tilde{\boldsymbol{x}}_i\}_{i \in \mathcal{I}_k}$ and $\boldsymbol{B}_k = \{\tilde{\boldsymbol{x}}_j\}_{j \in \mathcal{I}_{\bar{k}}}$, then problem (7) can be rewritten in the matrix formulation as

$$\min_{z_k} \nu z_k'\boldsymbol{M}_k z_k - z_k'\boldsymbol{H}_k z_k, \tag{8}$$

where the symmetry matrices $\boldsymbol{M}_k$ and $\boldsymbol{H}_k$ are defined by

$$\boldsymbol{M}_k = \boldsymbol{A}_k'\boldsymbol{A}_k = \sum_{i \in \mathcal{I}_k} \tilde{\boldsymbol{x}}_i\tilde{\boldsymbol{x}}_i', \tag{9}$$

$$\boldsymbol{H}_k = \boldsymbol{B}_k'\boldsymbol{B}_k = \sum_{j \in \mathcal{I}_{\bar{k}}} \tilde{\boldsymbol{x}}_j\tilde{\boldsymbol{x}}_j'. \tag{10}$$

Similar to GEPSVM, a regularization term $\|z_k\|_2^2$ is introduced to control the norm of the problem variable z_k, then the primal problem of IGEPSVM can be formulated as

$$\min_{z_k} \nu z_k'\boldsymbol{M}_k z_k - z_k'\boldsymbol{H}_k z_k + \delta\|z_k\|_2^2, \tag{11}$$

where $\delta > 0$ is a regularization parameter. Minimize problem (11) is equal to solve the following related eigenvalue problem (EP)

$$(\nu\boldsymbol{M}_k - \boldsymbol{H}_k + \delta\boldsymbol{I}) z_k = \lambda z_k, \tag{12}$$

whose solution is the eigenvector corresponding to smallest eigenvalue of problem (12). Note that, the generalized eigenvalue decomposition (5) in GEPSVM is replaced by the standard eigenvalue decomposition (12) in IGEPSVM, resulting in simpler optimization problem without the possible singularity.

III. ROBUST NONPARALLEL PROXIMAL SUPPORT VECTOR MACHINE VIA TRACE LASSO

A. PROBLEM FORMULATION

Obviously, the optimization problem (11) of IGEPSVM can be rewritten in its equivalence maximum formulation according to (9) and (10) as

$$\max_{z_k} \|\boldsymbol{B}_k z_k\|_2^2 - \nu\|\boldsymbol{A}_k z_k\|_2^2 - \delta\|z_k\|_2^2, \tag{13}$$

Remark 1: From (13)*, we can see that IGEPSVM utilizes the L_2-norm criterion to implement its empirical risk to find a pair of optimal hyperplane* (2)*. For k-th class proximal hyperplane ($k = \{1, 2\}$), it aims to maximize the L_2-norm distances from instances of the other class to this hyperplane (the first term in* (13)*), meanwhile minimize the L_2-norm distances for its own class instances (the second term in* (13)*). In fact, such an L_2-norm measurement is a square operation, which causes it sensitive to outliers. Fig. 1 illustrates that the employment of L_2-norm tends to exaggerate the effect of outliers, which leads to the large loss penalty dominating the sum in IGEPSVM.*

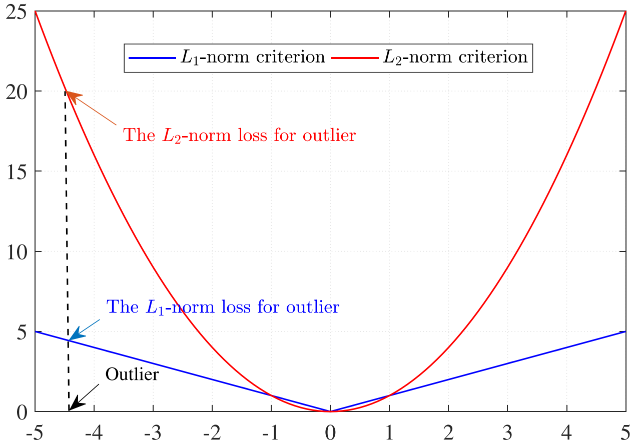


FIGURE 1. Illustration of the exaggeration effect of the L_2-norm versus the L_1-norm.

To improve model robustness in the presence of outliers, the L_1-norm measurement is usually considered as a more efficient way than the L_2-norm [37]. From Fig. 1, it can be observed that the L_1-norm can reduce the influence (loss penalty) of outliers compared with the L_2-norm.

The above analysis motives us to propose a robust nonparallel proximal SVM model with trace lasso regularization, termed as RNPSVM. In particular, our RNPSVM utilizes the robustness L_1-norm measurement to implement its empirical risk, which leads to the following optimization problem for k-th class hyperplane

$$\max_{z_k} \|\boldsymbol{B}_k z_k\|_1 - \nu\|\boldsymbol{A}_k z_k\|_1 - \frac{\delta}{2}\|z_k\|_2^2 - \eta\|\tilde{\boldsymbol{X}}\boldsymbol{D}_z\|_*, \tag{14}$$

where $\nu, \delta, \eta > 0$ are parameters, $k = \{1, 2\}$, $\|\cdot\|_1$ is the L_1-norm measurement, $\|\cdot\|_*$ is the nuclear norm,[1] $\tilde{\boldsymbol{X}} = [\boldsymbol{X}'\ \boldsymbol{e}]'$ is the augmented input matrix, and $\boldsymbol{D}_z = \mathrm{Diag}(z_k)$ is a diagonal matrix with its diagonal elements z_k.

To deliver the mechanism of RNPSVM, we now carry out the analysis and intuitive explanation for optimization problem (14):

- For the first and second terms in problem (14), the L_1-norm measurement is employed to implement the empirical risk of RNPSVM, which is robust to outliers.
- Maximizing the first L_1-norm based term $\|\boldsymbol{B}_k z_k\|_1 = \sum_{j\in\mathcal{I}_{\bar{k}}} |z_k'\tilde{\boldsymbol{x}}_j|$ aims to push instances $\tilde{\boldsymbol{x}}_{j\in\mathcal{I}_{\bar{k}}}$ of the other class as far as possible away from the k-th class proximal hyperplane $z_k'\tilde{\boldsymbol{x}} = 0$.
- Minimizing the second L_1-norm based term $\|\boldsymbol{A}_k z_k\|_1 = \sum_{i\in\mathcal{I}_k} |z_k'\tilde{\boldsymbol{x}}_i|$ hopes to make instances $\tilde{\boldsymbol{x}}_{i\in\mathcal{I}_k}$ of the k-th class as close as possible to this hyperplane.
- The third term in objective function is the L_2-norm of z_k, which is utilized to control the model complexity of RNPSVM and obtain a more appropriate model.
- The nuclear norm of matrix $\tilde{\boldsymbol{X}}\boldsymbol{D}_z$ in last term is utilized to capture the underlying information of data. Moreover, this term is known as trace lasso regularization [44]–[47], which makes our RNPSVM consider the sparsity and correlation of data simultaneously. Trace lasso naturally clusters the highly correlated data together.

B. MODEL OPTIMIZATION OF RNPSVM

In this subsection, we discuss how to optimize our RNPSVM. Due to the non-smooth L_1-norm loss and nuclear norm terms, it is difficult to achieve the global optimum of (14) directly by traditional optimization algorithms. Therefore, to optimize problem (14), we now consider its equivalent formulation by Proposition 1.

Proposition 1: Problem (14) *is equivalent to*

$$\max_{z_k,\boldsymbol{S}} \|\boldsymbol{B}_k z_k\|_1 - \nu\|\boldsymbol{A}_k z_k\|_1 - \frac{\delta}{2}\|z_k\|_2^2 - \frac{\eta}{2}z_k'\boldsymbol{D}_s z_k - \frac{\eta}{2}\,\mathrm{tr}(\boldsymbol{S}), \tag{15}$$

where $\boldsymbol{D}_s$ is a diagonal matrix with its diagonal elements extracted from the corresponding diagonal elements of matrix $\tilde{\boldsymbol{X}}'\boldsymbol{S}^{-1}\tilde{\boldsymbol{X}}$.

Proof: To prove it, we first introduce Lemma 1 [44].

Lemma 1: For any matrix $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$, the following variational equality for the nuclear norm of $\boldsymbol{Q}$ holds:

$$\|\boldsymbol{Q}\|_* = \frac{1}{2}\inf_{\boldsymbol{S}\succ\boldsymbol{0}} \left\{\mathrm{tr}(\boldsymbol{Q}'\boldsymbol{S}^{-1}\boldsymbol{Q}) + \mathrm{tr}(\boldsymbol{S})\right\}, \tag{16}$$

[1] The nuclear norm of matrix $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$ is the sum of its singular values $\sigma(\boldsymbol{Q})$, which is defined as $\|\boldsymbol{Q}\|_* = \mathrm{tr}\left(\sqrt{\boldsymbol{Q}'\boldsymbol{Q}}\right) = \sum_{i=1}^{\min\{m,n\}} \sigma_i(\boldsymbol{Q})$.

and its infimum is achieved at $\boldsymbol{S} = \sqrt{\boldsymbol{Q}\boldsymbol{Q}'}$, *where* $\mathrm{tr}(\cdot)$ *is the trace operator of a matrix.*

Let $\boldsymbol{Q} = \tilde{\boldsymbol{X}}\boldsymbol{D}_z$ in Lemma 1, then the nuclear norm $\|\tilde{\boldsymbol{X}}\boldsymbol{D}_z\|_*$ in problem (14) can be formulated as

$$\begin{aligned}\|\tilde{\boldsymbol{X}}\boldsymbol{D}_z\|_* &= \frac{1}{2}\inf_{\boldsymbol{S}>\boldsymbol{0}}\left\{\mathrm{tr}(\boldsymbol{D}_z'\tilde{\boldsymbol{X}}'\boldsymbol{S}^{-1}\tilde{\boldsymbol{X}}\boldsymbol{D}_z)+\mathrm{tr}(\boldsymbol{S})\right\}\\ &= \frac{1}{2}\inf_{\boldsymbol{S}>\boldsymbol{0}}\left\{z_k'\boldsymbol{D}_s z_k + \mathrm{tr}(\boldsymbol{S})\right\}, \qquad (17)\end{aligned}$$

where $\boldsymbol{D}_s$ is defined as a diagonal matrix with its corresponding diagonal elements extracted from the diagonal elements of $\tilde{\boldsymbol{X}}'\boldsymbol{S}^{-1}\tilde{\boldsymbol{X}}$ and the infimum of (17) is obtained at $\boldsymbol{S} = \sqrt{\tilde{\boldsymbol{X}}\boldsymbol{D}_z^2\tilde{\boldsymbol{X}}'}$.

Finally, substituting (17) into problem (14), it yields

$$\begin{aligned}&\max_{z_k}\ \|\boldsymbol{B}_k z_k\|_1 - \nu\|\boldsymbol{A}_k z_k\|_1 - \frac{\delta}{2}\|z_k\|_2^2\\ &-\frac{\eta}{2}\left(\inf_{\boldsymbol{S}>\boldsymbol{0}}\left\{z_k'\boldsymbol{D}_s z_k + \mathrm{tr}(\boldsymbol{S})\right\}\right),\\ &\Rightarrow \max_{z_k,\boldsymbol{S}}\|\boldsymbol{B}_k z_k\|_1 - \nu\|\boldsymbol{A}_k z_k\|_1 - \frac{\delta}{2}\|z_k\|_2^2\\ &-\frac{\eta}{2}z_k'\boldsymbol{D}_s z_k - \frac{\eta}{2}\,\mathrm{tr}(\boldsymbol{S}).\end{aligned}$$

The proof is established. ■

Remark 2: Although problem (15) *is convex, it is challenging to optimize variables* z_k *and* $\boldsymbol{S}$ *simultaneously. However, from Lemma 1, we can see that, when fix* z_k*, the objective function of problem* (15) *achieves maximum on the condition of* $\boldsymbol{S} = \sqrt{\tilde{\boldsymbol{X}}\boldsymbol{D}_z^2\tilde{\boldsymbol{X}}'}$.

The above analysis motives us to design an iterative algorithm to optimize z_k and $\boldsymbol{S}$ alternatively, summarized in Algorithm 1. More specifically, once obtaining z_k, we update $\boldsymbol{S}$ according to the latest z_k (Step 3). On the other hand, once updating $\boldsymbol{S}$, we compute $\boldsymbol{D}_s$ and then solve z_k by the following L_1-norm problem (Step 4)

$$\max_{z_k}\ \|\boldsymbol{B}_k z_k\|_1 - \nu\|\boldsymbol{A}_k z_k\|_1 - \frac{\delta}{2}\|z_k\|_2^2 - \frac{\eta}{2}z_k'\boldsymbol{D}_s z_k. \qquad (18)$$

Note that, due to the convexity of problem (15) w.r.t. z_k and $\boldsymbol{S}$, fixing one and maximizing another in Algorithm 1 will guarantee the increasing of its objective function. Therefore, the optimal solution z_k to problem (15) can be achieved via Algorithm 1, i.e., the updating operation in Step 3 and 4 iterates alternately until it converges.

Remark 3: In Algorithm 1, $\boldsymbol{S} = \sqrt{\tilde{\boldsymbol{X}}\boldsymbol{D}_z^2\tilde{\boldsymbol{X}}'}$ *can be calculated easily via eigenvalue decomposition of* $\tilde{\boldsymbol{X}}\boldsymbol{D}_z^2\tilde{\boldsymbol{X}}'$*. However, as for* z_k*, problem* (18) *is non-smooth due to the* L_1*-norm loss terms* $\|\boldsymbol{B}_k z_k\|_1$ *and* $\|\boldsymbol{A}_k z_k\|_1$*. As a result, it is difficult to obtain its optimal solution* z_k *directly by traditional gradient-based optimization techniques.*

In what follows, we focus on the optimization of problem (18). Note that the L_1-norm loss terms in problem (18) can be unfolded as $\|\boldsymbol{B}_k z_k\|_1 = \sum_{j\in\bar{\mathcal{I}}_k}|z_k'\tilde{\boldsymbol{x}}_j|$ and $\|\boldsymbol{A}_k z_k\|_1 = \sum_{i\in\mathcal{I}_k}|z_k'\tilde{\boldsymbol{x}}_i|$. Inspired by [37], we present an efficient iterative algorithm to maximize problem (18) via Algorithm 2, which updates the solution z_k iteratively until it converges.

Algorithm 1 The Procedure of RNPSVM for k-th Class Hyperplane

```
Input: The training data matrices $A_k = \{\tilde{x}_i\}_{i\in\mathcal{I}_k}$ and $B_k = \{\tilde{x}_j\}_{j\in\bar{\mathcal{I}}_k}$, and parameters $\mu, \delta, \mu > 0$.
1: Initialize $z_k \in \mathbb{R}^n$ as a random vector and then normalize it with unit length, i.e., $z_k = z_k/\|z_k\|_2$.
2: while not converged do
3:     Update $S$ according to $z_k$ by
           $S = U\,\mathrm{diag}(\sqrt{\lambda})U'$,   (19)
       where eigenvalues $\lambda$ and eigenvectors $U$ are computed by the eigenvalue decomposition of $\tilde{X}D_z^2\tilde{X}'$, and $D_z = \mathrm{diag}(z_k)$.
4:     Update $z_k$ according to $S$ by solving problem (18) via Algorithm 2, where $D_s$ is the diagonal elements of matrix $D_s = \tilde{X}'S^{-1}\tilde{X}$ and $S^{-1} = U\,\mathrm{diag}(1/\sqrt{\lambda})\,U'$.
5:     Check the convergence condition: the objective function value of problem (14) does not increase anymore, then terminate the loop.
6: end while
Output: The optimal solution $z_k^*$ to problem (14).
```

Algorithm 2 The Procedure for Solving Problem (18)

```
Input: The training data matrices $A_k = \{\tilde{x}_i\}_{i\in\mathcal{I}_k}$ and $B_k = \{\tilde{x}_j\}_{j\in\bar{\mathcal{I}}_k}$, parameters $\mu, \delta, \mu > 0$, and the matrix $D_s$.
1: Set the iterator $t = 0$, and initialize $z_k \in \mathbb{R}^n$ as a random vector and then normalize it with unit length, i.e., $z_k = z_k/\|z_k\|_2$.
2: while $\|z_k(t+1) - z_k(t)\| > 10^{-4}$ do
3:     Calculate two polarity functions $p_i(t)$ and $q_i(t)$ to unclose the absolute value operations in problem (18) according to $z_k(t)$
           $p_j(t) = \begin{cases} 1, & z_k(t)'\tilde{x}_j \geq 0 \\ -1, & z_k(t)'\tilde{x}_j < 0 \end{cases} \quad (j \in \bar{\mathcal{I}}_k)$,   (20)
       and
           $q_i(t) = \begin{cases} 1, & z_k(t)'\tilde{x}_i \geq 0 \\ -1, & z_k(t)'\tilde{x}_i < 0 \end{cases} \quad (i \in \mathcal{I}_k)$.   (21)
4:     Update $z_k(t+1)$ by
           $z_k(t+1) = (\nu V(t) + \delta I + \eta D_s)^{-1}u(t)$,   (22)
       where $V(t)$ and $u(t)$ are calculated as
           $V(t) = \sum_{i\in\mathcal{I}_k}\frac{\tilde{x}_i\tilde{x}_i'}{|z_k(t)'\tilde{x}_i|}$ and $u(t) = \sum_{j\in\bar{\mathcal{I}}_k}p_j\tilde{x}_j$.   (23)
5: end while
Output: The optimal solution $z_k^*$ for problem (18).
```

Remark 4: In very few cases, the denominator of $\boldsymbol{V}(t)$ *in* (23) *may become zero. Thus, similar to [37], to ensure the well-defined of* $\boldsymbol{V}(t)$*, we will set* $|z_k(t)'\tilde{\boldsymbol{x}}_i| = \sqrt{(z_k(t)'\tilde{\boldsymbol{x}}_i)^2 + \epsilon}$

when this situation occurs (ϵ is a very small positive float). Namely, when $\epsilon \approx 0$, it will approximate the original value.

C. CONVERGENCE ANALYSIS

In this subsection, we will focus on the convergence of Algorithm 2. Denote the objective function of (18) as

$$\boldsymbol{J}(z_k) = \|\boldsymbol{B}_k z_k\|_1 - \nu\|\boldsymbol{A}_k z_k\|_1 - \frac{\delta}{2}\|z_k\|_2^2 - \frac{\eta}{2} z_k' \boldsymbol{D}_s z_k. \quad (24)$$

The updating rule (22) guarantees the convergence of Algorithm 2, i.e., $\boldsymbol{J}(z_k(t+1)) \geq \boldsymbol{J}(z_k(t))$, which is justified by Theorem 1. To prove it, we introduce Lemma 2 [48].

Lemma 2: For any vector $\boldsymbol{v} = (v_1, \cdots, v_n)' \in \mathbb{R}^n$, the following variational equality holds

$$\|\boldsymbol{v}\|_1 = \min_{\tilde{x}\in\mathbb{R}^n_+} \left\{ \frac{1}{2}\sum_{k=1}^{n}\left(\frac{v_k^2}{u_k}\right) + \frac{1}{2}\|\boldsymbol{u}\|_1 \right\}. \quad (25)$$

and the minimum is uniquely achieved at $u_k = |v_k|$, where $\boldsymbol{u} = (u_1, \cdots, u_n)'$.

Theorem 1: Algorithm 2 monotonically non-decreases the objective function $\boldsymbol{J}(z_k)$ in each iteration, namely, $J(z_k(t+1)) \geq J(z_k(t))$.

Proof: Suppose that $z_k(t)$ is the optimal solution obtained in the t-th iteration, and the corresponding objective function (24) can be expressed as

$$\boldsymbol{J}(z_k(t)) = \|\boldsymbol{B}_k z_k(t)\|_1 - \nu\|\boldsymbol{A}_k z_k(t)\|_1 - \frac{\delta}{2}\|z_k(t)\|_2^2 - \frac{\eta}{2} z_k(t)' \boldsymbol{D}_s z_k(t). \quad (26)$$

The first and second L_1-norm loss terms in (26) can be further rewritten as

$$\begin{aligned}\|\boldsymbol{B}_k z_k(t)\|_1 &= \sum_{j\in\mathcal{I}_{\bar{k}}} |z_k(t)'\tilde{\boldsymbol{x}}_j| \\ &= z_k(t)' \sum_{j\in\mathcal{I}_{\bar{k}}} p_j(t)\tilde{\boldsymbol{x}}_j = z_k(t)'\boldsymbol{u}(t), \quad (27)\end{aligned}$$

and

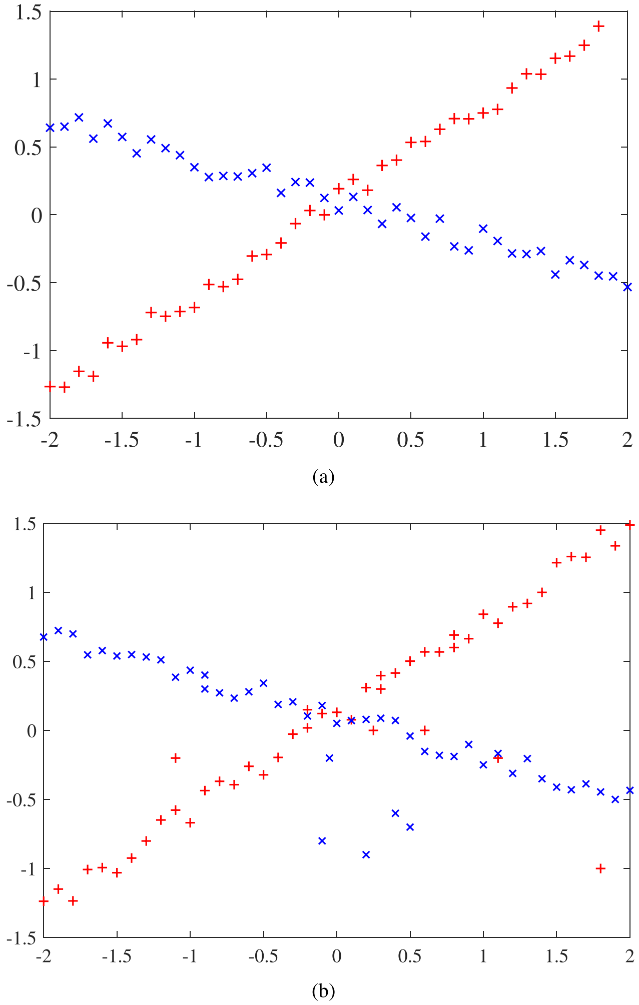
$$\begin{aligned}\|\boldsymbol{A}_k z_k(t)\|_1 &= \sum_{i\in\mathcal{I}_k} |z_k(t)'\tilde{\boldsymbol{x}}_i| \\ &= \frac{1}{2} z_k(t)' \left(\sum_{i\in\mathcal{I}_k} \frac{\tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i'}{|z_k(t)'\tilde{\boldsymbol{x}}_i|} \right) z_k(t) + \frac{1}{2}\|\boldsymbol{A}_k z_k(t)\|_1 \\ &= \frac{1}{2} z_k(t)' \boldsymbol{V}(t) z_k(t) + \frac{1}{2}\|\boldsymbol{a}(t)\|_1, \quad (28)\end{aligned}$$

where $\boldsymbol{u}(t)$ and $\boldsymbol{V}(t)$ are defined in (23), and vector $\boldsymbol{a}(t) = \boldsymbol{A}_k z_k(t)$. Substituting (27) and (28) into (26), it yields

$$J(z_k(t)) = z_k(t)'\boldsymbol{u}(t) - \frac{\nu}{2} z_k(t)'\boldsymbol{V}(t) z_k(t) - \frac{\nu}{2}\|\boldsymbol{a}(t)\|_1 - \frac{\delta}{2}\|z_k\|_2^2 - \frac{\eta}{2} z_k' \boldsymbol{D}_s(t) z_k. \quad (29)$$

However, it is difficult to directly calculate the derivative of function $J(z_k(t))$ due to the non-smooth L_1-norm. Therefore, we introduce the following surrogate function as

$$\boldsymbol{L}_t(\boldsymbol{\xi}) = \boldsymbol{\xi}'\boldsymbol{u}(t) - \frac{\nu}{2}\boldsymbol{\xi}'\boldsymbol{V}(t)\boldsymbol{\xi} - \frac{\nu}{2}\|\boldsymbol{a}(t)\|_1 - \frac{\delta}{2}\boldsymbol{\xi}'\boldsymbol{\xi} - \frac{\eta}{2}\boldsymbol{\xi}'\boldsymbol{D}_s(t)\boldsymbol{\xi}. \quad (30)$$



(a)

(b)

FIGURE 2. Synthetic "XOR" dataset (a) without and (b) with outliers. The red "+" and blue "×" scatter plot of the instance from Class 1 and Class 2, respectively. (a) "XOR" without outliers. (b) "XOR" with outliers.

It is worth noting that $\boldsymbol{L}_t(\boldsymbol{\xi})$ has only one variable $\boldsymbol{\xi}$ with fixed $\boldsymbol{u}(t)$, $\boldsymbol{v}(t)$, $\boldsymbol{a}(t)$ and $\boldsymbol{D}_s(t)$. Therefore, to obtain the maximum of $\boldsymbol{L}_t(\boldsymbol{\xi})$, we set the derivative of $\boldsymbol{L}_t(\boldsymbol{\xi})$ with respect to $\boldsymbol{\xi}$ be zero

$$\frac{\partial \boldsymbol{L}_t(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \boldsymbol{u}(t) - \nu\boldsymbol{V}(t)\boldsymbol{\xi} - \delta\boldsymbol{\xi} - \eta\boldsymbol{D}_s\boldsymbol{\xi} = \boldsymbol{0}. \quad (31)$$

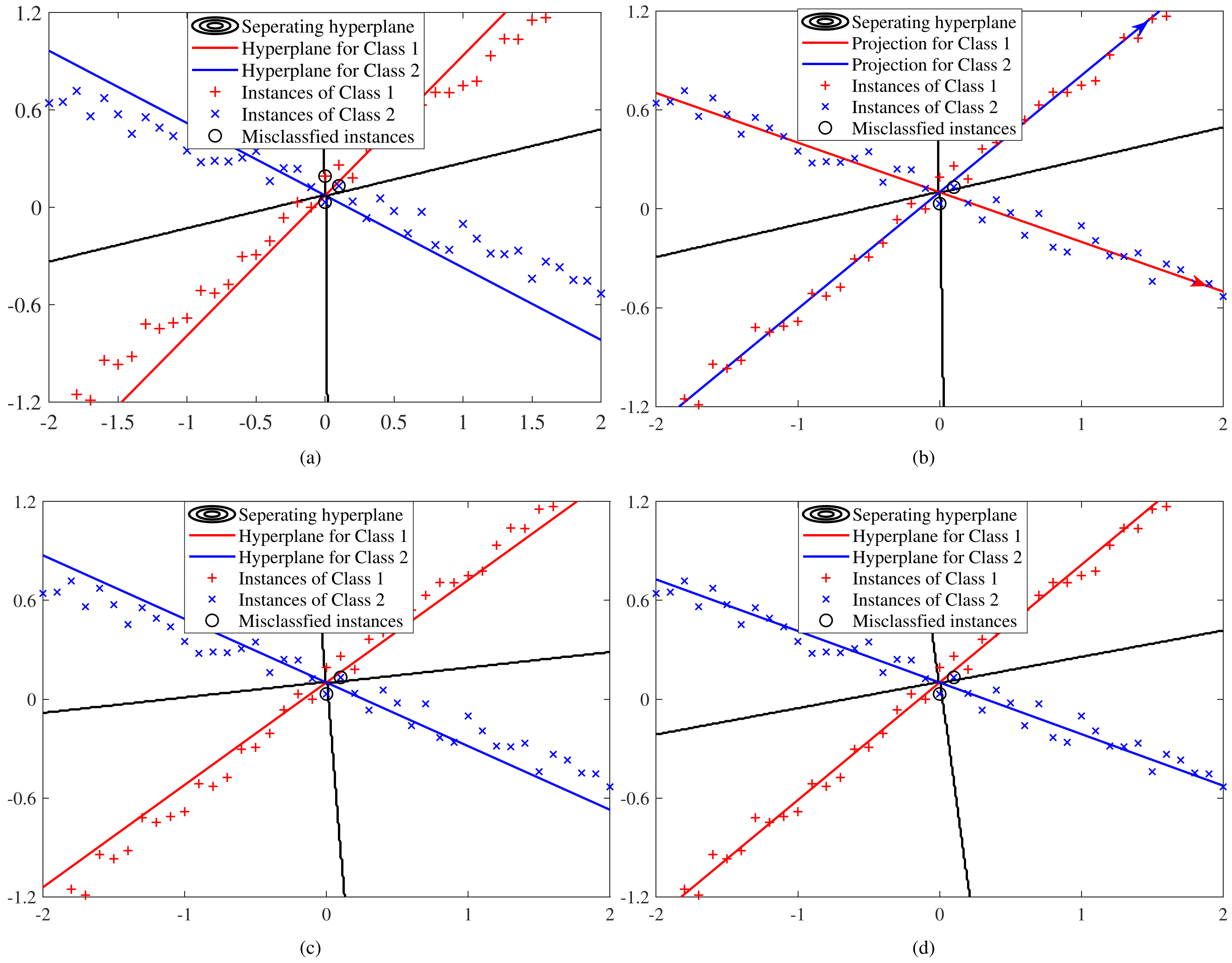
From (31), we have

$$\boldsymbol{\xi} = (\nu\boldsymbol{V}(t) + \delta\boldsymbol{I} + \eta\boldsymbol{D}_s(t))^{-1}\boldsymbol{u}(t). \quad (32)$$

Let $z_k(t+1) = (\nu\boldsymbol{V}(t) + \delta\boldsymbol{I} + \eta\boldsymbol{D}_s(t))^{-1}\boldsymbol{u}(t)$ as the updating rule (22) of z_k in Algorithm 2. Based on the above derivation, we will justify that $\boldsymbol{J}(z_k)$ monotonically non-decreases with this updating rule.

Since the maximum of $\boldsymbol{L}_t(\boldsymbol{\xi})$ is attained for $\boldsymbol{\xi} = z_k(t+1)$ in the t-th iteration, we have $\boldsymbol{L}_t(z_k(t+1)) \geq \boldsymbol{L}_t(\boldsymbol{\xi})$ for any $\boldsymbol{\xi}$. Thus, we derive

$$\begin{aligned}&\boldsymbol{L}_t(z_k(t+1)) \\ &= z_k(t+1)'\boldsymbol{u}(t) - \frac{\nu}{2} z_k(t+1)'\boldsymbol{V}(t) z_k(t+1)\end{aligned}$$



(a)

(b)

(c)

(d)

FIGURE 3. The learning results on "XOR" dataset without outliers: hyperplanes are learned by (a) IGEPSVM, (c) L1NPSVM and (d) RNPSVM, and projections are learned by (b) RPTSVM.

$$
\begin{aligned}
&-\frac{\nu}{2}\|\boldsymbol{a}(t)\|_1-\frac{\delta}{2}\|z_k(t+1)\|_2^2-\frac{\eta}{2}z_k(t+1)'\boldsymbol{D}_s z_k(t+1))\\
&\geq z_k(t)'\boldsymbol{u}(t)-\frac{\nu}{2}z_k(t)'\boldsymbol{V}(t)z_k(t)-\frac{\nu}{2}\|\boldsymbol{a}(t)\|_1\\
&\quad-\frac{\delta}{2}\|z_k(t)\|_2^2-\frac{\eta}{2}z_k(t)'\boldsymbol{D}_s z_k(t))\\
&=\boldsymbol{L}_t(z_k(t))=\boldsymbol{J}(z_k(t)). \qquad (33)
\end{aligned}
$$

Now, we proof $\boldsymbol{J}(z_k(t+1)) \geq \boldsymbol{L}_t(z_k(t+1))$ as follows.

Since $p_j(t+1)$ is the sign of $z_k(t+1)'\tilde{\boldsymbol{x}}_j$, we can conclude that, for any $j \in \mathcal{I}_{\bar{k}}$, it always has $p_j(t+1)z_k(t+1)'\tilde{\boldsymbol{x}}_j \geq 0$. However, for some $j \in \mathcal{I}_{\bar{k}}$, the corresponding $p_j(t)z_k(t+1)'\tilde{\boldsymbol{x}}_j$ may be negative. That is,

$$
\begin{aligned}
\|\boldsymbol{B}_k z_k(t+1)\|_1 &= z_k(t+1)'\sum_{j\in\mathcal{I}_{\bar{k}}} p_j(t+1)\tilde{\boldsymbol{x}}_j\\
&\geq z_k(t+1)'\sum_{j\in\mathcal{I}_{\bar{k}}} p_j(t)\tilde{\boldsymbol{x}}_j = z_k(t+1)'\boldsymbol{u}(t)
\end{aligned}
\qquad (34)
$$

On the other hand, from Lemma 2, we have

$$
\begin{aligned}
&\|\boldsymbol{A}_k z_k(t+1)\|_1\\
&\quad=\sum_{i\in\mathcal{I}_k}|z_k(t+1)'\tilde{\boldsymbol{x}}_i|\\
&\quad=\min_{\boldsymbol{u}\in\mathbb{R}_+^n}\left\{\frac{1}{2}\sum_{k=1}^{n}\left(\frac{(z_k(t+1)'\tilde{\boldsymbol{x}}_i)^2}{u_k}\right)+\frac{1}{2}\|\boldsymbol{u}\|_1\right\}\\
&\quad\leq\frac{1}{2}\sum_{i\in\mathcal{I}_k}\frac{(z_k(t+1)'\tilde{\boldsymbol{x}}_i)^2}{|\boldsymbol{a}_i(t)|}+\frac{1}{2}\|\boldsymbol{a}(t)\|_1\\
&\quad=\frac{1}{2}z_k(t+1)'\boldsymbol{V}(t)z_k(t+1)+\frac{1}{2}\|\boldsymbol{a}(t)\|_1, \qquad (35)
\end{aligned}
$$

Combining (34) and (35), yields

$$
\begin{aligned}
&\boldsymbol{J}(z_k(t+1))\\
&=\|\boldsymbol{B}_k z_k(t+1)\|_1-\nu\|\boldsymbol{A}_k z_k(t+1)\|_1\\
&\quad-\frac{\delta}{2}\|z_k(t+1)\|_2^2-\frac{\eta}{2}z_k(t+1)'\boldsymbol{D}_s z_k(t+1)
\end{aligned}
$$

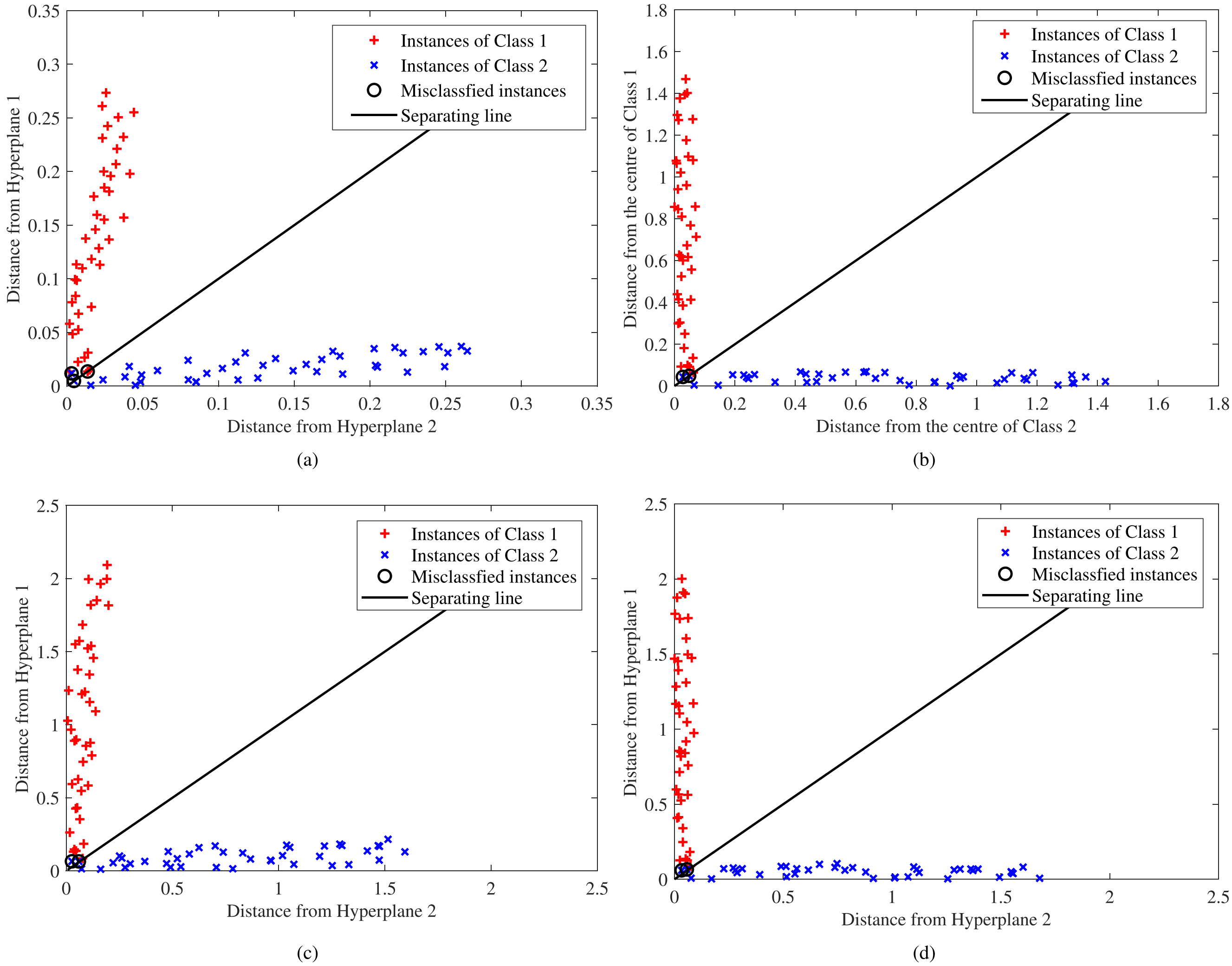


FIGURE 4. The distance scatter of instances on "XOR" dataset without outliers: (a) IGEPSVM, (b) RPTSVM, (c) L1NPSVM, and (d) RNPSVM. (a) IGEPSVM. (b) RPTSVM. (c) L1NPSVM. (d) RNPSVM.

$$\geq z_k(t+1)'\boldsymbol{u}(t) - \frac{\nu}{2} z_k(t+1)'\boldsymbol{V}(t) z_k(t+1) - \frac{\nu}{2}\|\boldsymbol{a}(t)\|_1$$

$$- \frac{\delta}{2}\|z_k(t+1)\|_2^2 - \frac{\eta}{2} z_k(t+1)'\boldsymbol{D}_s z_k(t+1))$$

$$= \boldsymbol{L}_t(z_k(t+1)). \tag{36}$$

Then, by using (33), we have

$$\boldsymbol{J}(z_k(t+1)) \geq \boldsymbol{J}(z_k(t)) \tag{37}$$

Thus, the objective function $\boldsymbol{J}(z_k)$ non-decreases via each iteration, which establishes the proof. ■

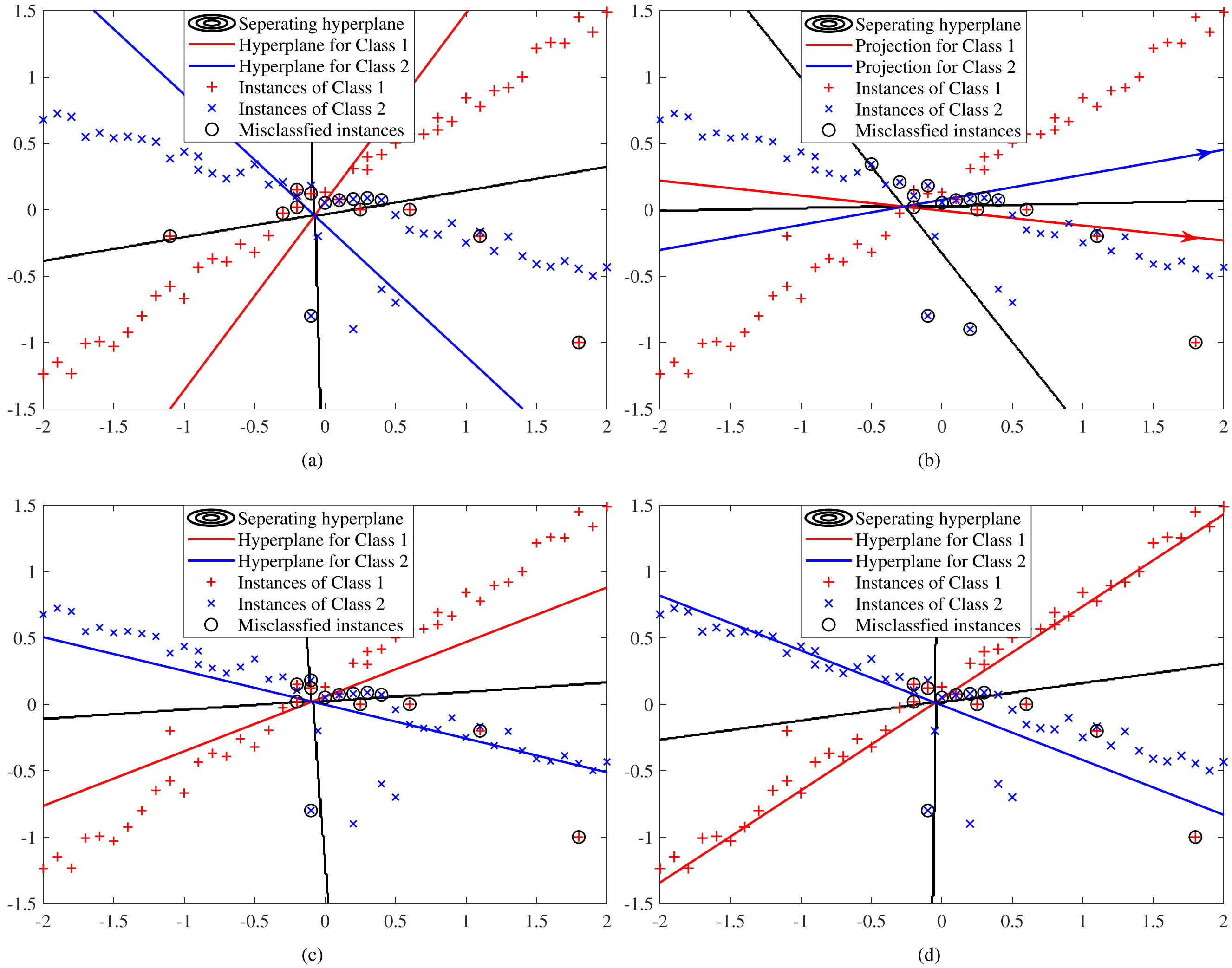
Note that, the objective function $\boldsymbol{J}(z_k)$ of (18) has a lower bound. Hence, Theorem 1 indicates that z_k will converge to a local optimal solution of problem (18) by the proposed Algorithm 2.

IV. EXPERIMENTAL RESULTS

To evaluate the robustness of the proposed RNPSVM, we investigate its classification accuracy[2] and efficiency[3] on both noisy synthetic and real-world datasets. In our experiments, we carry out comparisons between RNPSVM and three nonparallel SVMs, including IGEPSVM [23], RPTSVM [49] and L1NPSVM [43]. All the experiments are implemented by Matlab (2017b) on a personal computer (PC) with an Intel Core-i7 processor (2.9 GHz) and 32 GB random-access memory (RAM). The eigenvalue problem in IGEPSVM is solved by Matlab function "eig(·)". For RPTSVM, we resort Matlab function "quadprog(·)" to

[2] Classification accuracy (%) is defined as: Acc $= \frac{TP+TN}{TP+FP+TN+FN}$, where TP, TN, FP and FN are the number of true positive, true negative, false positive and false negative, respectively.

[3] We use the learning time (not include the parameters tuning time) to represent the training CPU time (s.) for each algorithm.



(a)

(b)

(c)

(d)

FIGURE 5. The learning results on "XOR" dataset with outliers: hyperplanes are learned by (a) IGEPSVM, (c) L1NPSVM and (d) RNPSVM, and projections are learned by (b) RPTSVM.

solve QPP. With regard to the parameter selection, we employ the standard ten-fold cross-validation technique.[4] Similar to [19], [27], [28], we use grid-based approach to obtain the optimal parameters for classifiers. The parameters δ, c_1, c_2 in GEPSVM, RPTSVM and RNPSVM are selected from $\{2^i | i = -5, -4, \ldots, 5\}$, while the learning rate in L1NPSVM is chosen from the set $\{0.0005, 0.001, 0.005, 0.01, 0.05\}$. Once selected, we returned them to learn the final decision function.

A. EXPERIMENTS ON SYNTHETIC DATASETS

To investigate the robustness of RNPSVM, in this subsection, we construct two types of synthetic "XOR" dataset, as shown in Fig.2, which is usually used to demonstrate the effectiveness of nonparallel SVM [19], [20], [29]. One is the original "XOR" dataset, which is generated by perturbing points from two intersecting lines

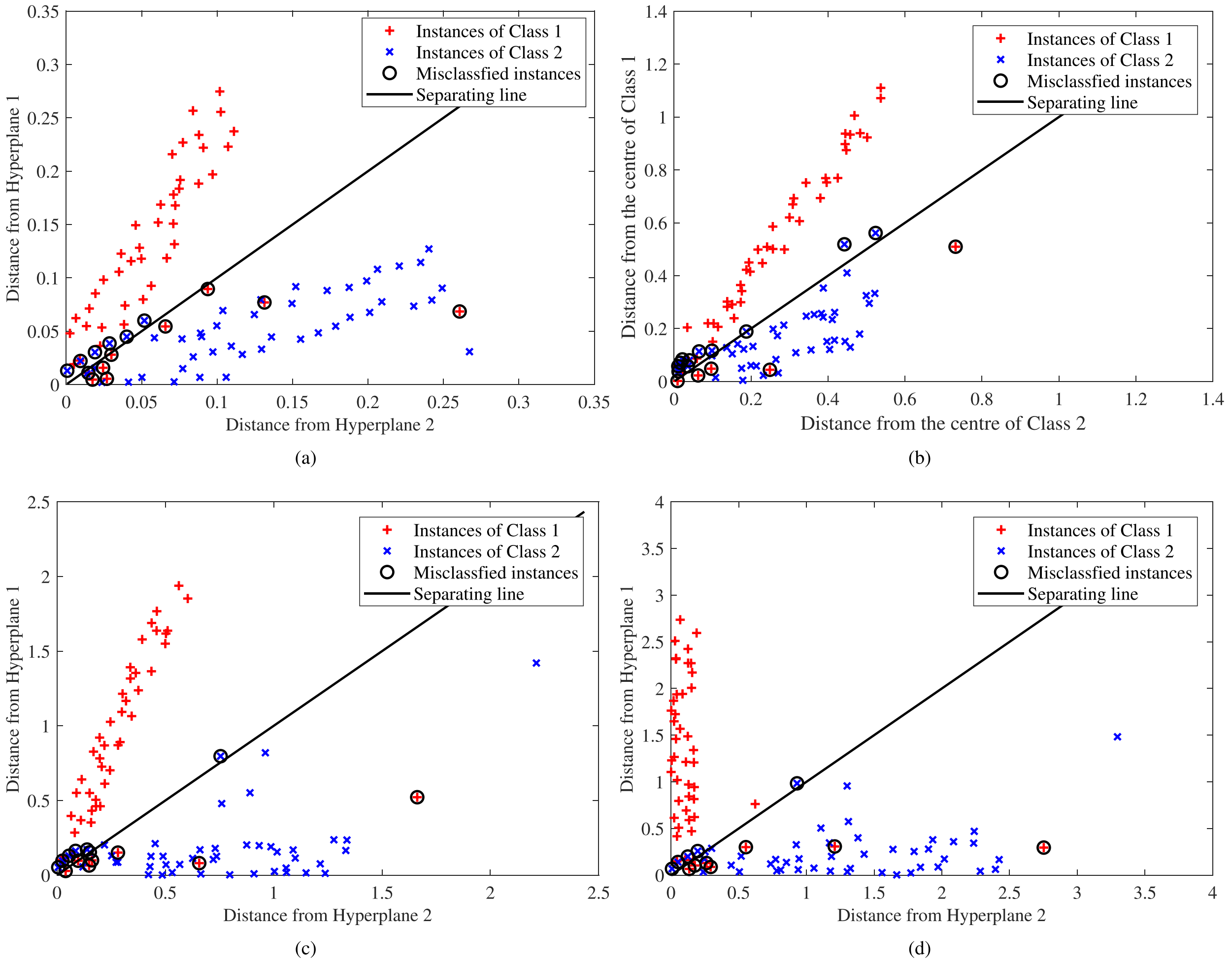
$$\begin{aligned} \text{Class 1: } y_i &= 0.7 \times x_i + \xi, \\ \text{Class 2: } y_i &= -0.3 \times x_i + \xi, \end{aligned}$$

where the Gaussian noise $\xi \sim \mathcal{N}(0, 0.2)$ is randomly added to each instance. Another is the contaminated "XOR" dataset, which is polluted by some outliers.

The learning results of each classifier on the original "XOR" dataset are illustrated in Fig.3. It can be seen that all classifiers can capture the underlying "XOR" distribution and obtain the optimal nonparallel hyperplanes/projections (the red and blue planes or directions in Fig.3) for non-outliers case. Moreover, we record the distance of each instance from two hyperplanes/projections learned by four classifiers, as shown in Fig.4. It reveals that all classifiers could separate the two classes well and obtain good performance.

In what follows, we turn to compare the performance of RNPSVM with other classifiers on "XOR" dataset with outliers. The learning results and the distance scatter of instances

[4] In detail [1], each dataset is partitioned into ten subsets with similar sizes and distributions. Then, the union of nine subsets is used as the training set while the remaining subset is used as the test set. The experiment is repeated 10 times such that every subset is used once as a test set.



(a)

(b)

(c)

(d)

FIGURE 6. The distance scatter of instances on "XOR" dataset with outliers: (a) IGEPSVM, (b) RPTSVM, (c) L1NPSVM, and (d) RNPSVM.

TABLE 1. Accuracy (%) and learning time (s) of each classifier on "XOR" dataset with or without outliers case.

Datasets	Classifiers			
	IGEPSVM	RPTSVM	L1NPSVM	RPNSVM
without outliers	96.34	**97.56**	**97.56**	**97.56**
	0.0042	0.0152	0.0107	0.0083
with outliers	84.54	83.50	85.56	**87.63**
	0.0056	0.0239	0.0136	0.0117

are illustrated in Fig.5 and Fig.6, respectively. It can be seen that the L_2-norm based IGEPSVM and RPTSVM cannot capture the "XOR" distribution well, and their proximal hyperlanes/projections are affected greatly by outliers. On the contrary, thanks to the L_1-norm loss criterion, L1NPSVM and our RPNSVM are less sensitive to the outliers. RNPSVM can discover the more discriminate information from the contaminated "XOR" than others.

For better comparisons, we also give the accuracy and learning time of each classifier on the above two "XOR" datasets in Table.1. The results show that all classifiers perform well without outliers. However, when the dataset is polluted by outliers, the performance of L_2-norm based IGEPSVM and RPTSVM decreases obviously compared

TABLE 2. Statistics for UCI datasets used in experiments.

Datasets	Instances	Features	Training	Testing
BUPA	345	6	241	104
Ionosphere	351	34	245	106
Titanic	2201	3	1540	661
Creadit	690	15	482	208
Hepatitis	155	19	108	47
Australian	690	14	482	208
PimaIndian	768	8	537	231
German	1000	24	700	300
Diabetes	768	8	537	231
Monks3	432	6	302	130
CMC	1473	9	1031	442
Hypothyroid	3163	25	2214	949

TABLE 3. The average learning results of each classifier on UCI datasets with slight noisy-level $m = 5\%$, in terms of testing accuracy (Acc) and learning time (Time).

Datasets	IGEPSVM	RPTSVM	L1NPSVM	RNPSVM
	Acc (%)	Acc (%)	Acc (%)	Acc (%)
	Time (s)	Time (s)	Time (s)	Time (s)
BUPA	66.42±3.06	65.17±3.44	66.87±2.96	**67.59±1.90**
	0.0964	0.2487	0.1298	0.1674
Ionosphere	**87.09±3.69**	85.18±2.58	86.39±4.13	86.93±3.13
	0.1386	0.3103	0.1854	0.2058
Titanic	74.51±2.40	73.72±2.32	**75.81±1.67**	75.52±2.64
	1.7175	4.4072	2.5746	2.0811
Creadit	83.94±2.39	83.06±4.07	84.98±2.83	**85.40±1.92**
	0.6787	2.6343	1.2458	1.7454
Hepatitis	82.88±3.69	80.45±2.92	83.28±3.48	**83.76±2.58**
	0.0328	0.1071	0.0523	0.0641
Australian	77.43±2.37	76.21±2.45	77.87±3.34	**78.62±2.26**
	1.4576	3.8192	1.8604	1.5329
PimaIndian	67.42±3.34	69.54±2.89	69.21±1.98	**71.63±2.64**
	0.1664	0.5137	0.3237	0.2818
German	71.42±2.67	69.46±4.03	**74.27±2.38**	73.95±1.61
	3.1910	7.0939	4.2076	5.2557
Diabetes	70.53±3.26	72.44±2.42	72.71±2.45	**74.04±2.69**
	2.5540	5.6462	2.2878	3.0433
Monks3	**87.20±3.25**	85.57±3.27	86.33±2.90	86.68±2.74
	0.8436	2.9501	0.9210	1.2217
CMC	71.82±4.49	70.49±2.94	**73.51±3.12**	73.26±2.81
	4.1647	8.3818	4.5556	5.2646
Hypothyroid	94.49±2.34	93.83±2.59	95.02±2.62	**95.16±1.94**
	8.5845	17.5920	6.2318	8.3478
Ave. Acc	77.92	77.09	78.85	79.37
Ave. Time	1.9688	4.4754	2.0480	2.4343
Ave. Rank	2.83	3.75	2.0	1.42

with the L_1-norm based L1NPSVM and RNPSVM. Moreover, our RNPSVM obtain the best performance among classifiers. As for learning time, RNPSVM is a bit slower than IGEPSVM, but faster than L1NPSVM and RPTSVM. The above results illustrate the robustness of our RNPSVM.

B. EXPERIMENTS ON UCI DATASETS

To further validate the generalization performance of the proposed RNPSVM, we consider twelve real-world datasets from the UCI machine learning repository,[5] whose statistics are listed in Table 2. These datasets represent a wide range of fields (include pathology, bioinformatics, finance and so on), sizes (from 155 to 3163) and features (from 9 to 34). The setting of our experiments is given as follows. Firstly, the features of all datasets are normalized to zero mean and unit deviation. Then, we divide each dataset into two subsets: 70% for training and 30% for testing. Afterwards,

[5] The UCI datasets are available at `http://archive.ics.uci.edu/ml`

TABLE 4. **The average learning results of each classifier on UCI datasets with heavy noisy-level $m = 20\%$, in terms of testing accuracy (Acc) and learning time (Time).**

Datasets	IGEPSVM	RPTSVM	L1NPSVM	RNPSVM
	Acc (%)	Acc (%)	Acc (%)	Acc (%)
	Time (s)	Time (s)	Time (s)	Time (s)
BUPA	62.09±5.75	61.51±6.24	64.41±2.89	**65.24±2.56**
	0.1287	0.2650	0.1341	0.1970
Ionosphere	82.96±6.34	80.32±9.74	84.61±3.82	**85.37±4.42**
	0.1310	0.3311	0.1899	0.2134
Titanic	71.79±3.95	71.40±3.77	**73.59±3.25**	73.37±3.01
	1.8692	5.1804	2.7936	2.3621
Creadit	80.18±4.13	79.64±5.25	82.08±4.14	**83.52±2.49**
	0.7331	2.5466	1.3137	1.9185
Hepatitis	79.07±7.20	78.38±4.53	81.71±3.71	**82.45±5.42**
	0.0484	0.1126	0.0580	0.0595
Australian	73.26±4.15	75.20±5.62	74.46±1.63	**76.77±2.75**
	1.2868	4.0356	1.5669	1.6243
PimaIndian	63.54±6.83	65.80±4.03	69.02±3.86	**70.66±3.28**
	0.1547	0.5873	0.3512	0.2974
German	68.92±5.79	65.86±5.65	71.19±3.74	**72.39±3.23**
	3.6979	6.2412	4.6425	4.7162
Diabetes	70.01±4.62	69.83±6.88	**72.86±7.54**	72.28±4.52
	2.6053	5.3420	3.4368	2.9110
Monks3	84.42±5.72	82.15±3.64	84.70±5.25	**85.92±2.82**
	0.7337	2.3100	0.7809	0.9688
CMC	69.60±5.43	68.78±4.74	71.83±4.49	**72.49±3.34**
	4.0626	7.9235	5.7672	4.1749
Hypothyroid	91.50±4.09	90.69±6.92	**94.56±2.94**	94.25±3.24
	7.0122	15.6658	10.9325	8.9196
Ave. Acc	74.77	74.13	77.08	77.89
Ave. Time	1.8720	4.2118	2.6639	2.3636
Ave. Rank	3.16	3.75	1.83	1.25

we randomly select m ratio of instances of training subset, and polluted their features with Gaussian noise to generate outliers. In this experiment, we consider two kind of situations: slight noisy-level $m = 5\%$ and heavy noisy-level $m = 20\%$. Finally, we transform them into the noisy classification tasks. Each experimental setting is repeated 10 times.

Table 3 and 4 list the average learning results of four classifiers on UCI datasets with the noisy-level 5% and 20%, respectively. The best performance is highlighted in bold. From results, we can learn that the classification performance for all classifiers will deteriorate generally with the aggravation of noisy-level. When datasets are polluted with the slight noisy-level (5%), our RNPSVM obtains the comparable or better performance than other three classifiers, as reported in Table 3. More specifically, RNPSVM gains the best accuracy on 7 of 12 datasets, while IGEPSVM and L1NPSVM only achieves that on 2 and 3 of 12 respectively.

When the noisy-level of datasets becomes more serious, as shown in Table 4, the performance of L_2-norm IGEPSVM and RPTSVM are dramatically worse. For example, on the

Wpbc dataset, IGEPSVM obtains the best accuracy (75.23%), while its performance reduced to 67.15% when the ratio of outliers is 20%. On the other hand, with the help of L_1-norm technique, RNPSVM and L1NPSVM are less sensitive to outliers than IGEPSVM and RPTSVM. Observing from Table 4, they can achieve satisfied performance even in the heavy noisy-level case. Although both RNPSVM and L1NPSVM are utilized the L_1-norm measurement for their empirical risks, the performance of RNPSVM is better than that of L1NPSVM in the most case. That is, our RNPSVM gains the best accuracy on 9 of 12 datasets. The reason may be that the trace lasso assists our RNPSVM to capture more underlying information, which makes it consider the sparsity and correlation of data simultaneously. Another reason may be that the efficient algorithm guarantees RNPSVM to optimize the convex sub-problem iteratively. Meanwhile, L1NPSVM utilizes the GA algorithm to optimize its L_1-norm ratio optimization problem, and both the need of the non-convex surrogate function and the difficult selection of step-size in GA may not guarantee to obtain the optimum solution.

As for the learning efficiency, we can observe from Table 3 and 4 that the noisy-level has little impact on the learning time of each classifier. Compared with IGEPSVM, our RNPSVM need spend a bit more time to solve the optimization problem with L_1-norm and trace lasso. In addition, although RNPSVM is slower than IGEPSVM, it is faster than RPTSVM.

To provide more statistical evidence [30], [50], we employ the Friedman's test to check whether there are significant differences between RNPSVM and other classifiers on the whole datasets, according to the testing accuracies in Table 3 and 4. The bottom of Table 3 and 4 lists the average rank of classifiers obtained by Friedman's test. It can be seen that the proposed RNPSVM is ranked first in both slight and heavy noisy-level situations, followed by L1NPSVM successively. These results confirmed the robustness of our RNPSVM against the outliers.

V. CONCLUSION

In this paper, we propose a novel robust nonparallel proximal SVM with trace lasso regularization for noisy classification, termed as RNPSVM. To improve the robustness of RNPSVM, the L_1-norm metric is utilized to implement its empirical risk. In detail, our RNPSVM aims to maximize the L_1-norm inter-class distance dispersion and minimize the L_1-norm intra-class distance dispersion simultaneously. Moreover, trace lasso [44]–[47] is further adopted as an adaptive norm to balance the L_1-norm and the L_2-norm regularization for our model. This elegance formulation allows RNPSVM to avoid the singularity problem effectively, which may encounter in GEPSVM. To optimize the non-smooth maximum problem of RNPSVM, an efficient iterative algorithm is further designed, whose convergence is guaranteed theoretically. Finally, the effectiveness and robustness of RNPSVM is confirmed by extensive experimental results on both synthetic and real-world noisy dataset.

One of our future work is to extend our model to deal with the nonlinear noisy classification tasks. The extensions to multi-class classification, regression, and semi-supervised learning are also interesting and under our consideration.

REFERENCES

[1] N. Deng, Y. Tian, and C. Zhang, *Support Vector Machines: Optimization Based Theory, Algorithms, and Extensions*. Philadelphia, PA, USA: CRC Press, 2013.

[2] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[3] A. Subasi, "Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders," *Comput. Biol. Med.*, vol. 43, no. 5, pp. 86–576, 2013.

[4] B. Gaonkar, R. T. Shinohara, and C. Davatzikos, "Interpreting support vector machine models for multivariate group wise analysis in neuroimaging," *Med. Image Anal.*, vol. 24, no. 1, pp. 190–204, 2015.

[5] H. Yin, X. Jiao, Y. Chai, and B. Fang, "Scene classification based on single-layer SAE and SVM," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3368–3380, 2015.

[6] S. Ma, B. Cheng, Z. Shang, and G. Liu, "Scattering transform and LSPTSVM based fault diagnosis of rotating machinery," *Mech. Syst. Signal Process.*, vol. 104, pp. 155–170, May 2018.

[7] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVM^{perf}," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1857–1863, 2015.

[8] S. Ding, H. Jia, M. Du, and Y. Xue, "A semi-supervised approximate spectral clustering algorithm based on HMRF model," *Inf. Sci.*, vol. 429, pp. 215–228, Mar. 2018.

[9] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods—Support Vector Learning*, B. Scholkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999.

[10] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods—Support Vector Learning*, B. Scholkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999.

[11] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[13] S. Shalev-Shwartz, Y. Singer, A. Cotter, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 1–27.

[14] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[15] O. L. Mangasarian and G. Fung, "Proximal support vector machine classifiers," in *Proc. Knowl. Discovery Data (KDD)*, 2001, pp. 77–86.

[16] Y.-J. Lee and O. L. Mangasarian, "SSVM: A smooth support vector machine for classification," *Comput. Optim. Appl.*, vol. 20, no. 1, pp. 5–22, 2001.

[17] O. L. Mangasarian and D. R. Musicant, "Lagrangian support vector machines," *J. Mach. Learn. Res.*, vol. 1, pp. 161–177, Sep. 2001.

[18] S. Ding, X. Hua, and J. Yu, "An overview on nonparallel hyperplane support vector machine algorithms," *Neural Comput. Appl.*, vol. 25, no. 5, pp. 975–982, 2014.

[19] O. Mangasarian and E. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 69–74, Jan. 2006.

[20] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, May 2007.

[21] Q. Ye and N. Ye, "Improved proximal support vector machine via generalized eigenvalues," in *Proc. Int. Joint Conf. Comput. Sci. Optim.*, 2009, pp. 705–709.

[22] M. R. Guarracino, C. Cifarelli, O. Seref, and P. M. Pardalos, "A classification method based on generalized eigenvalue problems," *Optim. Methods Softw.*, vol. 22, no. 1, pp. 73–81, 2007.

[23] Y.-H. Shao, N.-Y. Deng, W.-J. Chen, and Z. Wang, "Improved Generalized Eigenvalue Proximal Support Vector Machine," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 213–216, Mar. 2013.

[24] H. R. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 157–165, Feb. 2006.

[25] W.-J. Chen, Y.-H. Shao, D.-K. Xu, and Y.-F. Fu, "Manifold proximal support vector machine for semi-supervised classification," *Appl. Intell.*, vol. 40, no. 4, pp. 623–638, 2014.

[26] S. Sun, X. Xie, and C. Dong, "Multiview learning with generalized eigenvalue proximal support vector machines," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 688–697, Feb. 2018.

[27] X. Chen, J. Yang, Q. Ye, and J. Liang, "Recursive projection twin support vector machine via within-class variance minimization," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2643–2655, 2011.

[28] Q. Ye, N. Ye, and T. Yin, "Enhanced multi-weight vector projection support vector machine," *Pattern Recogn. Lett.*, vol. 42, pp. 91–100, 2014.

[29] Y. H. Shao, C. H. Zhang, X. B. Wang, and N. Y. Deng, "Improvements on twin support vector machines," *IEEE Trans. Neural Netw.*, vol. 22, no. 6, pp. 962–968, May 2011.

[30] Y.-H. Shao, W.-J. Chen, and N.-Y. Deng, "Nonparallel hyperplane support vector machine for binary classification problems," *Inf. Sci.*, vol. 263, pp. 22–35, Apr. 2014.

[31] Y. Tian, Z. Qi, X. Ju, Y. Shi, and X. Liu, "Nonparallel support vector machines for pattern classification," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1067–1079, Jul. 2014.

[32] W.-J. Chen, Y.-H. Shao, C.-N. Li, and N.-Y. Deng, "MLTSVM: A novel twin support vector machine to multi-label learning," *Pattern Recognit.*, vol. 52, pp. 61–74, Apr. 2016.

[33] Z. Qi, Y. Tian, and Y. Shi, "Structural twin support vector machine for classification," *Knowl.-Based Syst.*, vol. 43, no. 2, pp. 74–81, May 2013.

[34] W.-J. Chen, Y.-H. Shao, N.-Y. Deng, and Z.-L. Feng, "Laplacian least squares twin support vector machine for semi-supervised classification," *Neurocomputing*, vol. 145, pp. 465–476, Dec. 2014.

[35] Y.-H. Shao, W.-J. Chen, J.-J. Zhang, Z. Wang, and N.-Y. Deng, "An efficient weighted Lagrangian twin support vector machine for imbalanced data classification," *Pattern Recognit.*, vol. 47, no. 9, pp. 3158–3167, 2014.

[36] W.-J. Chen, Y.-H. Shao, and N. Hong, "Laplacian smooth twin support vector machine for semi-supervised classification," *Int. J. Mach. Learn.*, vol. 5, no. 3, pp. 459–468, 2014.

[37] N. Kwak, "Principal component analysis based on L_1-norm maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1672–1680, Sep. 2008.

[38] D. Meng, Q. Zhao, and Z. Xu, "Improve robustness of sparse PCA by L_1-norm maximization," *Pattern Recognit.*, vol. 45, no. 1, pp. 487–497, 2012.

[39] F. Zhong and J. Zhang, "Linear discriminant analysis based on ℓ_1-norm maximization," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3018–3027, Aug. 2013.

[40] F. Zhong, J. Zhang, and D. Li, "Discriminant locality preserving projections based on L_1-norm maximization," *IEEE Trans. Neural Netw. Lear. Syst.*, vol. 25, no. 11, pp. 2065–2074, Nov. 2014.

[41] Y. Pang and Y. Yuan, "Outlier-resisting graph embedding," *Neurocomputing*, vol. 73, nos. 4–6, pp. 968–974, 2010.

[42] H. Wang, X. Lu, Z. Hu, and W. Zheng, "Fisher discriminant analysis with ℓ_1-norm," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 828–842, Jun. 2014.

[43] C.-N. Li, Y.-H. Shao, and N.-Y. Deng, "Robust L_1-norm non-parallel proximal support vector machine," *Optim.*, vol. 65, no. 1, pp. 169–183, 2015.

[44] G. R. Obozinski, E. Grave, and F. R. Bach, "Trace Lasso: A trace norm regularization for correlated designs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2011, pp. 2187–2195.

[45] C. Lu, J. Feng, Z. Lin, and S. Yan, "Correlation adaptive subspace segmentation by trace lasso," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2014, pp. 1345–1352.

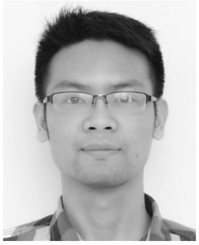
[46] J. Wang, C. Lu, M. Wang, P. Li, S. Yan, and X. Hu, "Robust face recognition via adaptive sparse representation," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2368–2378, Dec. 2014.

[47] G.-F. Lu, J. Zou, and Y. Wang, "L_1-norm and maximum margin criterion based discriminant locality preserving projections via trace lasso," *Pattern Recognit.*, vol. 55, pp. 207–214, Jul. 2016.

[48] R. Jenatton, G. Obozinski, and F. Bach, "Structured sparse principal component analysis," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 1–8.

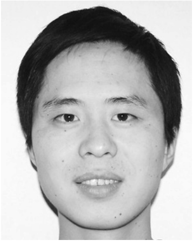
[49] Y.-H. Shao, Z. Wang, W.-J. Chen, and N.-Y. Deng, "A regularization for the projection twin support vector machine," *Knowl.-Based Syst.*, vol. 37, pp. 203–210, Jan. 2013.

[50] Z. Yang, K.-T. Fang, and S. Kotz, "On the Student's t-distribution and the t-statistic," *J. Multivariate Anal.*, vol. 98, no. 6, pp. 1293–1304, 2007.



WEI-JIE CHEN received the B.S. degree in electrical engineering and automation and the Ph.D. degree in control science and engineering from the Zhejiang University of Technology, China, in 2006 and 2011, respectively. From 2017 to 2018, he was a Visiting Scholar with the Centre for Artificial Intelligence, University of Technology Sydney, Australia (with supervisor Prof. I. Tsang). He is currently an Associate Professor with the Zhijiang College, Zhejiang University of Technology. He has published over 40 refereed papers. His research interests include pattern recognition, intelligence computation, and manifold learning.

KAI-LI YANG is currently pursuing the M.Sc. degree with the Zhejiang University of Technology, China. Her research interests include data mining and machine learning.



YUAN-HAI SHAO received the B.S. degree in information and computing science from the College of Mathematics, Jilin University, in 2006, and the master's degree in applied mathematics and Ph.D. degree in operations research and management from the College of Science, China Agricultural University, China, in 2008 and 2011, respectively. He is currently a Professor with the School of Economics and Management, Hainan University. His research interests include data mining, machine learning, and optimization methods. He has published over 80 refereed papers on these areas.

YU-JUAN CHEN received the master's degree in statistics from Jiangsu University, China, in 2004, and the Ph.D. degree in statistics from Zhejiang Gongshang University, in 2013. She is currently an Associate Professor with the School of Data Sciences, Zhejiang University of Finance and Economics. Her research interests include decision making, comprehensive evaluation and uncertainty, and data mining.

JU ZHANG received the B.S. degree in mechanical engineering from the Zhejiang University of Technology, in 1994, and the Ph.D. degree in automatic control engineering from Zhejiang University, in 2005. From 2005 to 2006, he was a Visiting Scholar with Stuttgart University, Germany. From 2014 to 2015, he was a Visiting Scholar with Michigan State University, USA. He is currently a Professor with the Zhejiang University of Technology. His research interests include model predictive control, hybrid systems, and linear parameter varying systems.

JING-JING YAO received the master's degree in engineering from Northwest Normal University, China, in 2012. She is currently a Lecturer with the Zhijiang College, Zhejiang University of Technology. Her research interests include education data mining and machine learning.