L1-Norm Distance Minimization-Based Fast Robust Twin Support Vector k-Plane Clustering

Qiaolin Ye, *Member, IEEE*, Henghao Zhao, Zechao Li, Xubing Yang, Shangbing Gao, *Member, IEEE*, Tongming Yin, and Ning Ye

***Abstract*— Twin support vector clustering (TWSVC) is a recently proposed powerful k-plane clustering method. It, however, is prone to outliers due to the utilization of squared L2-norm distance. Besides, TWSVC is computationally expensive, attributing to the need of solving a series of constrained quadratic programming problems (CQPPs) in learning each clustering plane. To address these problems, this brief first develops a new k-plane clustering method called L1-norm distance minimization-based robust TWSVC by using robust L1-norm distance. To achieve this objective, we propose a novel iterative algorithm. In each iteration of the algorithm, one CQPP is solved. To speed up the computation of TWSVC and simultaneously inherit the merit of robustness, we further propose Fast RTWSVC and design an effective iterative algorithm to optimize it. Only a system of linear equations needs to be computed in each iteration. These characteristics make our methods more powerful and efficient than TWSVC. We also conduct some insightful analysis on the existence of local minimum and the convergence of the proposed algorithms. Theoretical insights and effectiveness of our methods are further supported by promising experimental results.**

***Index Terms*— Iterative algorithm, k-plane clustering, L1-norm distance, linear equations, twin support vector clustering (TWSVC).**

I. Introduction

Clustering, as one of the fundamental topics in machine learning and pattern classification, has widely been applied to various areas, such as text mining, web analysis, and bioinformatics [1]–[3]. The target of clustering is to group similar samples into the same cluster while dissimilar samples into different clusters, such that the meaningful structures of data are well discovered [4]. There are many clustering techniques in the literature. For example, [5] and [6] considered kernel-based clustering, [7] and [9] used max-margin constraint in the clustering, and [9]–[11] proposed point-based central clustering techniques, e.g., k-mean, k-median, and fuzzy c-mean.

In recent years, there has been increasing interest in k-plane clustering. k-plane clustering changes the entity of the center from being a point to that of being a plane [12]. kPC [12], proximal plane clustering (PPC) [13], and twin support vector clustering (TWSVC) [14] are three typical k-plane clustering techniques. kPC only considers the similarities among the samples in a cluster plane and ignores the dissimilarities. PPC is proposed to address the problem. In fact, PPC is based on multisurface proximal support vector machine classification via generalized eigenvalues (GEPSVM) [15], while TWSVC is an extension to twin support vector machine (TWSVM) [16], [17]. TWSVM is a milestone in the development of the multiplane-based classification, which can yield solid theoretical results and outperforms GEPSVM and support vector machine (SVM) in terms of classification performance. Among kPC, PPC, and TWSVC, TWSVC gains the best clustering result. However, this method formulates the objective using squared L2-norm distance in a cluster plane, which could exaggerate the effect of outliers. Furthermore, TWSVC is computationally expensive, since it requires solving a series of constrained quadratic programming problems (CQPPs) to determine each of the k-cluster planes. This is an open problem raised in section "conclusion" of [14].

In this brief, we aim to develop fast robust k-plane clustering methods. For this purpose, a novel L1-norm distance minimization-based robust*TWSVC* (RTWSVC) method is first proposed. It is well known that L1-norm distance is more robust to outliers than the L2-norm one, since it does not magnify the effect of outliers [18], [19]. Solving the objective of RTWSVC is very challenging, because it is not only nonsmooth but also nonconvex. As one of the important theoretical contributions of this brief, we present a novel iterative algorithm for the derivation of the clustering planes. However, based on the algorithm, RTWSVC, like TWSVC, determines the k-cluster planes by solving a series of CQPPs, leading to the expensive computational cost. To reduce the computational costs of TWSVC and RTWSVC and inherit the merit of the robustness of RTWSVC, fast RTWSVC (FRTWSVC) is further developed. Likewise, an effective iterative algorithm is proposed to solve FRTWSVC. In each iteration of the algorithm, only a system of linear equations needs to be solved. In addition, some insightful analysis on the existence of local minimum and the convergence of the proposed algorithms are conducted. Theoretical studies and extensive experimental results on several benchmark data sets verify the effectiveness and applicability of our methods.

II. Related Work

A. Notations

Suppose that $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^T \in \mathbf{R}^{m\times n}$ is the data set with m samples of n dimensions. The L2-norm of a vector is denoted

Manuscript received December 25, 2015; revised June 15, 2016, March 23, 2017, August 8, 2017, and August 18, 2017; accepted August 27, 2017. Date of publication October 3, 2017; date of current version August 20, 2018. This work was supported in part by the National Science Foundation of China under Grant 61401214, Grant 61773210, Grant 61603184, Grant 61772275, and Grant 61402192, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20171453, Grant K20140058, Grant BK20170033, and Grant BK20140794, in part by the Jiangsu Key Laboratory for Internet of Things and Mobile Internet Technology, and in part by the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety. (*Corresponding authors: Qiaolin Ye; Shangbing Gao.*)

Q. Ye, H. Zhao, X. Yang, and N. Ye are with the College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China, with the Laboratory for Internet of Things and Mobile Internet Technology of Jiangsu Province, Huaiyin Institute of Technology, Nanjing 223003, China, and also with the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: yqlcom@njfu.edu.cn; zzhaohenhao@163.com; xbyang@njfu.edu.cn; yening2004@gmail.com).

Z. Li is with the College of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: zechao.li@njust.edu.cn).

S. Gao is with the College of Computer Engineering, Huaiyin Institute of Technology, Huai'an, China (e-mail: luxiaofen_2002@126.com).

T. Yin is with the College of Forestry, Nanjing Forestry University, Nanjing 210037, China (e-mail: tmyin@njfu.com.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2749428

by $||\cdot||$. Let $\text{sgn}(\cdot)$ be a sign function with $\text{sgn}(\cdot) = -1$ if $(\cdot)$ is a negative value and $\text{sgn}(\cdot) = 1$ otherwise. The primary task of clustering is to partition $\mathbf{X}$ into k clusters. We represent the samples in the i-th cluster by $\mathbf{X}_i \in \mathbf{R}^{m_i \times n}$ and those in the rest clusters by $\overline{\mathbf{X}}_i \in \mathbf{R}^{(m-m_i)\times n}$, where m_i denotes the number of samples in the i-th cluster ($i = 1, \ldots, k$). Note that $\mathbf{X}_i$ and $\overline{\mathbf{X}}_i$ are two subsets of $\mathbf{X}$ and $\mathbf{X} = [\mathbf{X}_i, \overline{\mathbf{X}}_i]^T$. A column vector of 1s of arbitrary dimension is denoted by $\mathbf{e}$, and the weight and bias of the i-th clustering plane are denoted by $\mathbf{w}_i \in \mathbf{R}^n$ and $b_i \in \mathbf{R}$, respectively. Define three augmented matrices: $\mathbf{z}_i = [\mathbf{w}_i^T \;\; b_i]^T \in \mathbf{R}^{n+1}$, $\mathbf{G}_i = [\mathbf{X}_i \;\; \mathbf{e}] \in \mathbf{R}^{m_i \times (n+1)}$, and $\mathbf{H}_i = [\overline{\mathbf{X}}_i \;\; \mathbf{e}] \in \mathbf{R}^{(m-m_i)\times(n+1)}$.

B. kPC

kPC [12] targets to cluster $\mathbf{X}$ into k clusters, and the samples of the corresponding cluster get clustered around the cluster plane $\mathcal{P}_i$

$$\mathcal{P}_i = \mathbf{w}_i^T \mathbf{x} + b_i = 0 \quad i = 1, \ldots, k \tag{1}$$

kPC first initializes the cluster assignment of $\mathbf{X}$. For the current $\mathbf{X}_i$, the k cluster planes are determined by minimizing the following problem with $i = 1, 2, \ldots, k$:

$$\min_{\mathbf{w}_i,\, b_i} \; 0.5||\mathbf{X}_i \mathbf{w}_i + \mathbf{e}b_i||^2, \quad \text{s.t. } \mathbf{w}_i^T \mathbf{w}_i = 1. \tag{2}$$

The solution to (2) can be found by solving a standard eigenvalue problem [12]. Then, each sample $\mathbf{x}$ can be relabeled by

$$\text{Cluster}(\mathbf{x}) = \underset{i=1,\ldots,\,k}{\text{argmin}} \; ||\mathbf{w}_i^T \mathbf{x} + b_i||^2. \tag{3}$$

From (3), the corresponding clusters of the samples are updated. Then, the new clustering planes are computed by (2). The process continues till some terminate conditions are satisfied.

C. PPC

kPC considers the intracluster similarities among the samples in a cluster plane but overlooks the intercluster separation. PPC [13] aims to overcome this problem.

With the initial cluster assignment of $\mathbf{X}$, like kPC, PPC iteratively updates the cluster planes and the corresponding clusters of the samples by

$$\min_{\mathbf{w}_i,\, b_i} \; ||\mathbf{X}_i \mathbf{w}_i + \mathbf{e}b_i||^2 - c||\overline{\mathbf{X}}_i \mathbf{w}_i + \mathbf{e}b_i||^2, \quad \text{s.t. } \mathbf{w}_i^T \mathbf{w}_i = 1 \tag{4}$$

and (3), respectively, till some terminate conditions are satisfied. In (4), c is a regularization parameter balancing different contributions of the two terms. Clearly, PPC is a k-plane clustering extension to GEPSVM [15].

D. TWSVC

Recently, Wang *et al.* [14] proposed a more powerful k-plane clustering method than kPC and PPC, called TWSVC, which is based on TWSVM [17]. With the initial cluster assignment of $\mathbf{X}$, TWSVC finds the k_cluster planes of (1) by solving the following problem with $i = 1, \ldots, k$:

$$\begin{aligned} \min_{\mathbf{w}_i, b_i} \; & 0.5||\mathbf{X}_i \mathbf{w}_i + \mathbf{e}b_i||^2 + c\mathbf{e}^T \boldsymbol{\xi}_i \\ \text{s.t. } & |\overline{\mathbf{X}}_i \mathbf{w}_i + \mathbf{e}b_i| + \boldsymbol{\xi}_i \geq \mathbf{e}, \quad \boldsymbol{\xi}_i \geq 0 \end{aligned} \tag{5}$$

where $\boldsymbol{\xi}_i$ is a symmetric *Hinge* loss function. Then, we use (3) to update the corresponding clusters of the samples. Based on the updated clusters, we proceed to update the k_cluster planes of (1). The process continues till some terminate conditions are satisfied. Note that the cluster-updating process is the same as that of kPC or PPC. Rewrite (5) as

$$\begin{aligned} \min_{\mathbf{w}_i,\, b_i} \; & 0.5||\mathbf{X}_i \mathbf{w}_i + \mathbf{e}b_i||^2 + c\mathbf{e}^T \boldsymbol{\xi}_i \\ \text{s.t. } & \mathbf{U}_i(\overline{\mathbf{X}}_i \mathbf{w}_i + \mathbf{e}b) + \boldsymbol{\xi}_i \geq \mathbf{e}, \quad \boldsymbol{\xi}_i \geq 0 \end{aligned} \tag{6}$$

where $\mathbf{U}_i = \text{diag}(\text{sign}(\overline{\mathbf{X}}_i \mathbf{w}_i + \mathbf{e}b_i))$ is a diagonal matrix. The constraint of the problem is nonconvex. TWSVC solves (6) using the constrained concave–convex procedure (CCCP) [20], [21], where $\mathbf{U}_i$ is viewed as a variable that depends on $\mathbf{w}_i$ and b_i (or $\mathbf{z}_i$). Specifically, compute $\mathbf{U}_i$ based on the current $\mathbf{z}_i$ obtained in the last iteration and then update $\mathbf{z}_i$ by solving the Wolfe dual problem of (6). Suppose that $\mathbf{z}_i^{(p)}$ is the solution of the pth iteration, and $\mathbf{z}_i^{(p+1)}$ is the one of the $(p+1)$th iteration, where $\mathbf{z}_i^{(p+1)} = (\mathbf{G}_i^T \mathbf{G}_i)^{-1} \mathbf{M}_i^{(p)^T} \boldsymbol{\alpha}_i^{(p+1)}$. Here, $\mathbf{M}_i^{(p)} = \mathbf{U}_i^{(p)} \mathbf{H}_i$ and $\boldsymbol{\alpha}_i^{(p+1)}$ denote the updated Lagrangian multiplier vector that is defined as

$$\begin{aligned} \boldsymbol{\alpha}_i^{(p+1)} = \underset{\boldsymbol{\alpha}_i}{\text{argmin}} \; & 0.5\boldsymbol{\alpha}_i^T \mathbf{M}_i^{(p)} \left(\mathbf{G}_i^T \mathbf{G}_i\right)^{-1} \mathbf{M}_i^{(p)^T} \boldsymbol{\alpha}_i - \mathbf{e}^T \boldsymbol{\alpha}_i \\ & \text{s.t. } \mathbf{0} \leq \boldsymbol{\alpha}_i \leq c\mathbf{e}. \end{aligned} \tag{7}$$

The results of [17] demonstrate the promising performance of TWSVC.

III. L1-Norm Distance Minimization-Based Fast Robust Twin Support Vector k-Plane Clustering

In this section, we first propose RTWSVC, a new k-plane clustering method. Then, an FRTWSVC method is developed.

A. Linear RTWSVC and FRTWSVC

It has been well known that squared L2-norm distance measurement is nonrobust to outliers, which means that TWSVC may not obtain the desired solution. In the literature, the L1-norm distance is usually applied to handle this problem [18], [19]. Illuminated by this, we propose a new method for k-plane clustering called RTWSVC. Same as TWSVC, we first partition the samples of $\mathbf{X}$ into k clusters, and then use

$$\begin{aligned} \min_{\mathbf{w}_i, b_i} \; & 0.5||\mathbf{X}_i \mathbf{w}_i + \mathbf{e}b_i||_1 + c\mathbf{e}^T \boldsymbol{\xi}_i \\ \text{s.t. } & |\overline{\mathbf{X}}_i \mathbf{w}_i + \mathbf{e}b_i| + \boldsymbol{\xi}_i \geq \mathbf{e}, \quad \boldsymbol{\xi}_i \geq 0 \end{aligned} \tag{8}$$

to update the k_cluster planes. Update the corresponding clusters of the samples using (3). The process continues till some terminate conditions are satisfied.

As observed, in each cluster update, TWSVC minimizes the similar problem of TWSVC in (5) using the L1-norm distance rather than the squared L2-norm one. Rewrite (8) as

$$\min_{\mathbf{z}_i} \; 0.5||\mathbf{G}_i \mathbf{z}_i||_1 + c\mathbf{e}^T \boldsymbol{\xi}_i, \; \text{s.t. } |\mathbf{H}_i \mathbf{z}_i| + \boldsymbol{\xi}_i \geq \mathbf{e}, \quad \boldsymbol{\xi}_i \geq 0. \tag{9}$$

Solving the problem is very difficult, because it involves nonsmooth L1-norm terms and the nonconvex constraints. Recently, there have been a lot of algorithms to solve nonsmooth problems, such as [19], [22] and [23]. Zhong [19] proposed a gradient-ascending iterative procedure to minimize the L1-norm distance maximization–minimization problem. The research [22] proposed a randomized block-coordinate variant of the classic Frank–Wolfe algorithm to solve the dual structural SVM problem involving convex objective. In [23], generalized conditional gradient with gradient sliding is proposed to solve nonsmooth unconstrained composite optimization problems. Clearly, our objective has a different formulation. Next, we propose an iterative algorithm to solve (9).

Note that $||\mathbf{G}_i\mathbf{z}_i||_1 = \sum_j |\mathbf{g}_{i,j}\mathbf{z}_i|$, where $\mathbf{g}_{i,j}^T \in \mathbf{R}^n$ denotes the j-th row of $\mathbf{G}_i$. Solving the problem is very challenging, since it contains nonconvex constraints and absolute value operations. We derive its solution as follows.

Rewrite (9) with the following formulation:

$$\min_{\mathbf{z}_i} \ 0.5\mathbf{z}_i^T \sum_j \left(\mathbf{g}_{i,j}^T\mathbf{g}_{i,j}/|\mathbf{g}_{i,j}\mathbf{z}_i|\right)\mathbf{z}_i + c\mathbf{e}^T\boldsymbol{\xi}_i$$
$$\text{s.t. } \operatorname{diag}(\operatorname{sign}(\mathbf{H}_i\mathbf{z}_i))(\mathbf{H}_i\mathbf{z}_i) + \boldsymbol{\xi}_i \geq \mathbf{e}, \quad \boldsymbol{\xi}_i \geq 0. \tag{10}$$

Let $d_{j,j} = 1/|\mathbf{g}_{i,j}\mathbf{z}_i|$ and construct the diagonal matrices $\mathbf{D}_i$ with its j-th diagonal entry as d_{jj}. To the end, (10) becomes

$$\min_{\mathbf{z}_i} \ 0.5\mathbf{z}_i^T\mathbf{G}_i^T\mathbf{D}_i\mathbf{G}_i\mathbf{z}_i + c\mathbf{e}^T\boldsymbol{\xi}_i, \ \text{s.t. } \mathbf{F}_i(\mathbf{H}_i\mathbf{z}_i) + \boldsymbol{\xi}_i \geq \mathbf{e}, \quad \boldsymbol{\xi}_i \geq 0 \tag{11}$$

where $\mathbf{F}_i = \operatorname{diag}(\operatorname{sign}(\mathbf{H}_i\mathbf{z}_i))$. $\mathbf{D}_i$ and $\mathbf{F}_i$ are dependent on $\mathbf{z}_i$ and thus are two latent variables. We propose an iterative algorithm described in Algorithm 1 to obtain the solution $\mathbf{z}_i$.

Algorithm 1 Efficient Iterative Algorithm to Solve Problem (9)

Input: Input data matrices $\mathbf{X}_i$ and $\overline{\mathbf{X}}_i$.
Set $p = 0$. Initialize $\mathbf{z}_i^{(p)}$.
Construct the matrices $\mathbf{G}_i = [\mathbf{X}_i \quad \mathbf{e}]$ and $\mathbf{H}_i = [\overline{\mathbf{X}}_i \quad \mathbf{e}]$.
While not converge **do**

1. Compute the diagonal matrix $\mathbf{D}_i^{(p)}$ with its j-th diagonal entry $d_{jj}^{(p)} = 1/|\mathbf{g}_{i,j}\mathbf{z}_i^{(p)}|$, where $\mathbf{g}_{i,j} \in \mathbf{R}^{1\times n}$ denotes the j-th row of $\mathbf{G}_i$.
2. Compute the diagonal matrix $\mathbf{F}_i^{(p)}$=diag(sign($\mathbf{H}_i\mathbf{z}_i^{(p)}$)).
3. Compute $\mathbf{z}_i^{(p+1)}$ by solving

$$\mathbf{z}_i^{(p+1)} = \arg\min_{\mathbf{z}_i} \ 0.5\mathbf{z}_i^T\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i\mathbf{z}_i + c\mathbf{e}^T\boldsymbol{\xi}_i,$$
$$\text{s.t. } \mathbf{F}_i^{(p)}(\mathbf{H}_i\mathbf{z}_i) + \boldsymbol{\xi}_i \geq \mathbf{e}, \ \boldsymbol{\xi}_i \geq 0.$$

5. $p = p + 1$.

End while
Output: The learned $\mathbf{w}_i$ and b_i from $\mathbf{z}_i = [\mathbf{w}_i^T \quad b_i]^T$.

As seen, in each iteration, $\mathbf{z}_i$ is computed with current $\mathbf{D}_i$ and $\mathbf{F}_i$, and then $\mathbf{D}_i$ and $\mathbf{F}_i$ are updated with currently computed $\mathbf{z}_i$. The iteration continues until the algorithm converges. There is no doubt that the problem in step 3 of Algorithm 1 is convex, which can be solved by the following Wolfe dual formation:

$$\boldsymbol{\beta}_i^{(p+1)} = \arg\min_{\boldsymbol{\beta}_i} \ 0.5\boldsymbol{\beta}_i^T\mathbf{F}_i^{(p)}\mathbf{H}_i\left(\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i\right)^{-1}\mathbf{H}_i^T\mathbf{F}_i^{(p)^T}\boldsymbol{\beta}_i - \mathbf{e}^T\boldsymbol{\beta}_i$$
$$\text{s.t. } \mathbf{0} \leq \boldsymbol{\beta}_i \leq c\mathbf{e} \tag{12}$$

where $\boldsymbol{\beta}_i$ is the Lagrangian multiplier vector. Once $\boldsymbol{\beta}_i^{(p+1)}$ is known, the updated $\mathbf{z}_i^{(p+1)}$ in the $(p+1)$th iteration is defined as $\mathbf{z}_i^{(p+1)} = (\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i)^{-1}\mathbf{H}_i^T\mathbf{F}_i^{(p)^T}\boldsymbol{\beta}_i^{(p+1)}$.

Recall that TWSVC is based on TWSVM for classification [16], [17]. Similar to the iterative algorithm of TWSVC, Algorithm 1 requires solving a series of CQPPs, leading to the expensive computational cost. Previous efforts have shown that an effective method, which not only speeds up the computation but also loses no performance of TWSVM, is to convert the original CQPP into a least-squares problem [24] by replacing the inequality constraints with equality ones. Inspired by this, we promote the computational efficiency of RTWSVC by reformulating problem (8) in each cluster updated as follows:

$$\min_{\mathbf{w}_i,,b_i} ||\mathbf{X}_i\mathbf{w}_i + \mathbf{e}b_i||_1 + c||\boldsymbol{\xi}_i||_1, \quad \text{s.t. } |\overline{\mathbf{X}}_i\mathbf{w}_i + \mathbf{e}b_i| + \boldsymbol{\xi}_i = \mathbf{e}. \tag{13}$$

Simply speaking, our problem is based on the efficient least-squares version of TWSVM [24]. We call the reformulation FRTWSVC. We can rewrite (13) with the following problem:

$$\min_{\mathbf{z}_i} \ ||\mathbf{G}_i\mathbf{z}_i||_1 + c||\boldsymbol{\xi}_i||_1, \ \text{s.t. } |\mathbf{H}_i\mathbf{z}_i| + \boldsymbol{\xi}_i = \mathbf{e}. \tag{14}$$

Rewrite (14) with the following equivalent formulation:

$$\min_{\mathbf{z}_i} \ \mathbf{z}_i^T \sum_j \left(\mathbf{g}_{i,j}^T\mathbf{g}_{i,j}/|\mathbf{g}_{i,j}\mathbf{z}_i|\right)\mathbf{z}_i + c\sum_j ((\boldsymbol{\xi}_{i,j})^2/|\boldsymbol{\xi}_{i,j}|)$$
$$\text{s.t. } \operatorname{diag}(\operatorname{sign}(\mathbf{H}_i\mathbf{z}_i))(\mathbf{H}_i\mathbf{z}_i) + \boldsymbol{\xi}_i = \mathbf{e} \tag{15}$$

where $\boldsymbol{\xi}_{i,j}$ the j-th element of $\boldsymbol{\xi}_i$. Let $a_{j,j} = 1/|\boldsymbol{\xi}_{i,j}|$ and construct the diagonal matrix $\mathbf{A}_i$ with its j-th diagonal entry as a_{jj}. To the end, (15) becomes

$$\min_{\mathbf{z}_i} \ \mathbf{z}_i^T\mathbf{G}_i^T\mathbf{D}_i\mathbf{G}_i\mathbf{z}_i + c\boldsymbol{\xi}_i^T\mathbf{A}_i\boldsymbol{\xi}_i, \ \text{s.t. } \mathbf{F}_i(\mathbf{H}_i\mathbf{z}_i) + \boldsymbol{\xi}_i = \mathbf{e}. \tag{16}$$

For the definitions of $\mathbf{D}_i$ and $\mathbf{F}_i$ [see (11)]. $\mathbf{A}_i$, $\mathbf{D}_i$, and $\mathbf{F}_i$ are dependent on $\mathbf{z}_i$. We can propose the similar iterative algorithm of Algorithm 1 to obtain the solution $\mathbf{z}_i$. The algorithm is described in Algorithm 2.

Algorithm 2 Efficient Iterative Algorithm to Solve Problem (14)

Input: Input data matrices $\mathbf{X}_i$ and $\overline{\mathbf{X}}_i$.
Initialize $\mathbf{z}_i^{(p)}$, and set $p = 0$.
Construct the matrices $\mathbf{G}_i = [\mathbf{X}_i \quad \mathbf{e}]$ and $\mathbf{H}_i = [\overline{\mathbf{X}}_i \quad \mathbf{e}]$.
While not converge **do**

1. Compute the diagonal matrix $\mathbf{D}_i^{(p)}$ with its j-th diagonal entry $d_{jj}^{(p)} = 1/|\mathbf{g}_{i,j}\mathbf{z}_i^{(p)}|$, where $\mathbf{g}_{i,j} \in \mathbf{R}^{1\times n}$ denotes the j-th row of $\mathbf{G}_i$.
2. Compute the diagonal matrix $\mathbf{F}_i^{(p)}$=diag(sign($\mathbf{H}_i\mathbf{z}_i^{(p)}$)).
3. Compute the diagonal matrix $\mathbf{A}_i^{(p)}$ with its j-th diagonal entry as $a_{jj}^{(p)} = 1/|\boldsymbol{\xi}_{i,j}|$, where $\boldsymbol{\xi}_{i,j}$ is the j-th element of $\boldsymbol{\xi}_i$.
4. Compute $\mathbf{z}_i^{(p+1)}$ by solving

$$\mathbf{z}_i^{(p+1)} = \arg\min_{\mathbf{z}_i} \ \mathbf{z}_i^T\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i\mathbf{z}_i + c\boldsymbol{\xi}_i^T\mathbf{A}_i^{(p)}\boldsymbol{\xi}_i,$$
$$\text{s.t. } \mathbf{F}_i^{(p)}(\mathbf{H}_i\mathbf{z}_i) + \boldsymbol{\xi}_i = \mathbf{e}..$$

5. $p = p + 1$.

End while
Output: The learned $\mathbf{w}_i$ and b_i from $\mathbf{z}_i = [\mathbf{w}_i^T \quad b_i]^T$.

In each iteration of Algorithm 2, one needs to solve the least-squares problem in step 4. Substituting the equality constraints into the objective function, the problem becomes

$$\mathbf{z}_i^{(p+1)} = \min_{\mathbf{z}_i} \ \mathbf{z}_i^T\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i\mathbf{z}_i + c(\mathbf{e} - \mathbf{F}_i^{(p)}\mathbf{H}_i\mathbf{z}_i)^T\mathbf{A}_i^{(p)}(\mathbf{e} - \mathbf{F}_i^{(p)}\mathbf{H}_i\mathbf{z}_i). \tag{17}$$

Taking the derivative of (17) with respect to $\mathbf{z}_i$ and setting it as zero, we obtain $\mathbf{z}_i^{(p+1)} = (1/c\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i + \mathbf{H}_i^T\mathbf{F}_i^{(p)}\mathbf{A}_i^{(p)}\mathbf{F}_i^{(p)}\mathbf{H}_i)^{-1}\mathbf{H}_i^T\mathbf{F}_i^{(p)}\mathbf{A}_i^{(p)}\mathbf{e}$. Since $\mathbf{F}_i^{(p)}$ and $\mathbf{A}_i^{(p)}$ are diagonal matrices, and each of the diagonal elements of $\mathbf{F}_i^{(p)}$ are either 1 or -1, $\mathbf{F}_i^{(p)}\mathbf{A}_i^{(p)}\mathbf{F}_i^{(p)} = \mathbf{A}_i^{(p)}$. Therefore, we have

$$\mathbf{z}_i^{(p+1)} = \left(1/c\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i + \mathbf{H}_i^T\mathbf{A}_i^{(p)}\mathbf{H}_i\right)^{-1}\mathbf{H}_i^T\mathbf{F}_i^{(p)}\mathbf{A}_i^{(p)}\mathbf{e}. \tag{18}$$

Hence, we can compute a system of linear equations in (18).

The following theorems guarantee the convergence of Algorithms 1 and 2. Recall that TWSVC can be guaranteed to yield a local minimal solution under the CCCP. Similarly, we show that the solutions of our RTWSVC and FRTWSVC are also local minimal by the following theorems.

Theorem 1: Algorithm 1 monotonically decreases the objective of problem (9) in each iteration.

Proof: First, we rewrite the problem in step 3 of Algorithm 1 with the following equivalent formulation:

$$\mathbf{z}_i^{(p+1)} = \arg\min_{\mathbf{z}_i} \ 0.5\mathbf{z}_i^T \mathbf{G}_i^T \mathbf{D}_i^{(p)} \mathbf{G}_i \mathbf{z}_i + c\mathbf{e}^T \max\big(0,\ \mathbf{e} - \mathbf{F}_i^{(p)} \mathbf{H}_i \mathbf{z}_i\big). \tag{19}$$

According to step 3 in Algorithm 1, in the p iteration, we have

$$\begin{aligned} 0.5\mathbf{z}_i^{(p+1)^T} \mathbf{G}_i^T \mathbf{D}_i^{(p)} \mathbf{G}_i \mathbf{z}_i^{(p+1)} + c\mathbf{e}^T \max\big(0, \mathbf{e} - \mathbf{F}_i^{(p)} \mathbf{H}_i \mathbf{z}_i^{(p+1)}\big) \\ \le 0.5||\mathbf{G}_i \mathbf{z}_i^{(p)}||_1 \ + c\mathbf{e}^T \max\big(0, \mathbf{e} - \mathbf{F}_i^{(p)} \mathbf{H}_i \ \mathbf{z}_i^{(p)}\big). \end{aligned} \tag{20}$$

For each j, we have

$$\begin{aligned} &\big(|\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}| - |\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}|\big)^2 \\ &\quad = \big(\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}\big)^2 + \big(\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}\big)^2 - 2|\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}||\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}| \ge 0 \end{aligned}$$

which leads to

$$\begin{aligned} &\frac{\big(\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}\big)^2}{2|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}|} + \frac{|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}|}{2} - |\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}| \ge 0 \\ &\Rightarrow |\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}| - \frac{\big(\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}\big)^2}{2|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}|} \le \frac{|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}|}{2} \\ &= |\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}| - \frac{\big(\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}\big)^2}{2|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}|} \\ &\Rightarrow 2|\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}| - \frac{\big(\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}\big)^2}{|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}|} \\ &\le 2|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}| - \frac{\big(\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}\big)^2}{|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}|}. \end{aligned} \tag{21}$$

Thus, the following inequality holds:

$$\begin{aligned} &\sum_j \left(|\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}| - 0.5 \frac{\big(\mathbf{g}_{i,j} \mathbf{z}_i^{(p+1)}\big)^2}{|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}|} \right) \\ &\quad \le \sum_j \left(|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}| - 0.5 \frac{\big(\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}\big)^2}{|\mathbf{g}_{i,j} \mathbf{z}_i^{(p)}|} \right) \\ &\Rightarrow ||\mathbf{G}_i \mathbf{z}_i^{(p+1)}||_1 - 0.5\mathbf{z}_i^{(p+1)^T} \mathbf{G}_i^T \mathbf{D}_i^{(p)} \mathbf{G}_i \mathbf{z}_i^{(p+1)} \\ &\le ||\mathbf{G}_i \mathbf{z}_i^{(p)}||_1 - 0.5\mathbf{z}_i^{(p)^T} \mathbf{G}_i^T \mathbf{D}_i^{(p)} \mathbf{G}_i \mathbf{z}_i^{(p)}. \end{aligned} \tag{22}$$

Using the equality $\mathbf{F}_i^{(p)} \mathbf{H}_i \mathbf{z}_i^{(p)} = |\mathbf{H}_i \mathbf{z}_i^{(p)}|$ and combining (20) and (22), we arrive at

$$\begin{aligned} ||\mathbf{G}_i \mathbf{z}_i^{(p+1)}||_1 + c\mathbf{e}^T \max\big(0,\ \mathbf{e} - \mathbf{F}_i^{(p)} \mathbf{H}_i \mathbf{z}_i^{(p+1)}\big) \\ \le ||\mathbf{G}_i \mathbf{z}_i^{(p)}||_1 + c\mathbf{e}^T \max\big(0,\ \mathbf{e} - |\mathbf{H}_i \mathbf{z}_i^{(p)}|\big). \end{aligned} \tag{23}$$

Cleary, the function $f(\mathbf{z}_i) = |\mathbf{H}_i \mathbf{z}_i|$ is convex with respect to $\mathbf{z}_i$. According to the research [25], for any convex function $f(\mathbf{x})$ with respect to variable $\mathbf{x}$, the inequality $f(\mathbf{x}) \ge f(\mathbf{x}^{(t)}) + \nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^{(t)}} (\mathbf{x} - \mathbf{x}^{(t)})$ is satisfied, where $\nabla_{\mathbf{x}} f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^{(t)}}$ denotes the gradient of $f(\mathbf{x})$ at point $\mathbf{x}^{(t)}$. Using this fact and $\nabla f(\mathbf{z}_i)|_{\mathbf{z}_i=\mathbf{z}_i^{(p)}} = \text{diag}(\text{sign}(\mathbf{H}_i \mathbf{z}_i^{(p)}))\mathbf{H}_i$, we have $|\mathbf{H}_i \mathbf{z}_i^{(p+1)}| \ge |\mathbf{H}_i \mathbf{z}_i^{(p)}| + \text{diag}(\text{sign}(\mathbf{H}_i \mathbf{z}_i^{(p)}))\mathbf{H}_i (\mathbf{z}_i^{(p+1)} - \mathbf{z}_i^{(p)})$, leading to

$$\begin{aligned} \big|\mathbf{H}_i \mathbf{z}_i^{(p+1)}\big| \ &\ge \ \text{diag}\big(\text{sign}\big(\mathbf{H}_i \mathbf{z}_i^{(p)}\big)\big)\mathbf{H}_i \mathbf{z}_i^{(p+1)} = \mathbf{F}_i^{(p)} \mathbf{H}_i \mathbf{z}_i^{(p+1)} \\ &\Rightarrow \mathbf{e} - \big|\mathbf{H}_i \mathbf{z}_i^{(p+1)}\big| \le \mathbf{e} - \mathbf{F}_i^{(p)} \mathbf{H}_i \mathbf{z}_i^{(p+1)}. \end{aligned} \tag{24}$$

We further have $\mathbf{e}^T \max(0, \ \mathbf{e} - |\mathbf{H}_i \mathbf{z}_i^{(p+1)}|) \le \mathbf{e}^T \max(0, \mathbf{e} - \mathbf{F}_i^{(p)} \mathbf{H}_i \mathbf{z}_i^{(p+1)})$, which leads to

$$\begin{aligned} &\big|\big|\mathbf{G}_i \mathbf{z}_i^{(p+1)}\big|\big|_1 + \mathbf{e}^T \max\big(0, \mathbf{e} - \big|\mathbf{H}_i \mathbf{z}_i^{(p+1)}\big|\big) \\ &\qquad \le \big|\big|\mathbf{G}_i \mathbf{z}_i^{(p+1)}\big|\big|_1 + \mathbf{e}^T \max\big(0, \mathbf{e} - \mathbf{F}_i^{(p)} \mathbf{H}_i \mathbf{z}_i^{(p+1)}\big). \end{aligned}$$

Using the inequality and (23), one gets

$$\begin{aligned} \big|\big|\mathbf{G}_i \mathbf{z}_i^{(p+1)}\big|\big|_1 + c\mathbf{e}^T \max\big(0, \mathbf{e} - \big|\mathbf{H}_i \mathbf{z}_i^{(p+1)}\big|\big) \le \big|\big|\mathbf{G}_i \mathbf{z}_i^{(p)}\big|\big|_1 \\ + c\mathbf{e}^T \max\big(0, \mathbf{e} - \big|\mathbf{H}_i \mathbf{z}_i^{(p)}\big|\big). \end{aligned} \tag{25}$$

Since problem (9) is lower bounded by 0, Algorithm 1 converges. The equality in (25) holds when the algorithm converges. Therefore, the objective value of problem (9) decreases in each iteration till the algorithm converges. □

Theorem 2: Algorithm 1 converges to a local minimal solution to problem (9).

Proof: Define the Lagrangian function of problem (9) as

$$\mathcal{L}(\mathbf{z}_i, \boldsymbol{\xi}_i) \ = \ ||\mathbf{G}_i \mathbf{z}_i||_1 + c\mathbf{e}^T \boldsymbol{\xi}_i - \boldsymbol{\alpha}^T (|\mathbf{H}_i \mathbf{z}_i| + \boldsymbol{\xi}_i - \mathbf{e}) \ - \boldsymbol{\beta}^T \ \boldsymbol{\xi}_i \tag{26}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the vectors of Lagrange multipliers. Taking the derivative of $\mathcal{L}(\mathbf{z}_i, \boldsymbol{\xi}_i)$ with respect to $\mathbf{z}_i$ and $\boldsymbol{\xi}_i$, respectively, and setting them as 0, we get the Karush-Kuhn-Tucker (KKT) condition of problem (9) as follows:

$$\mathbf{G}_i^T \text{sign}(\mathbf{G}_i \mathbf{z}_i) + \mathbf{H}_i^T \text{diag}\big(\text{sign}(\mathbf{z}_i^T \mathbf{H}_i)\big)\boldsymbol{\alpha} = 0, \quad c\mathbf{e} - \boldsymbol{\alpha} - \boldsymbol{\beta} = 0. \tag{27}$$

In each iteration of Algorithm 1, we find the optimal $\mathbf{z}_i^{(p+1)}$ to the problem in step 3. Therefore, the converged solution of Algorithm 1 satisfies the KKT condition of the problem. Define the Lagrangian function of the problem in step 3 of Algorithm 1 as

$$\begin{aligned} \mathcal{L}_2(\mathbf{z}_i, \boldsymbol{\xi}_i) \ = \ 0.5\mathbf{z}_i^T \mathbf{G}_i^T \mathbf{D}_i^{(p)} \mathbf{G}_i \mathbf{z}_i + c\mathbf{e}^T \boldsymbol{\xi}_i \\ - \boldsymbol{\alpha}^T (\mathbf{F}_i^{(p)} (\mathbf{H}_i \mathbf{z}_i) + \boldsymbol{\xi}_i - \mathbf{e}) - \boldsymbol{\beta}^T \boldsymbol{\xi}_i. \end{aligned} \tag{28}$$

Taking the derivative of $\mathcal{L}_2(\mathbf{z}_i, \ \boldsymbol{\xi}_i)$ with respect to $\mathbf{z}_i$ and $\boldsymbol{\xi}_i$, respectively, and setting them as zero gives

$$\mathbf{G}_i^T \mathbf{D}_i^{(p)} \mathbf{G}_i \mathbf{z}_i - \mathbf{H}_i^T \mathbf{F}_i^{(p)} \boldsymbol{\alpha} = 0, \quad c\mathbf{e} - \boldsymbol{\alpha} - \boldsymbol{\beta} = 0. \tag{29}$$

According to the definitions of $\mathbf{D}_i^{(p)}$ and $\mathbf{F}_i^{(p)}$ in Algorithm 1, the equivalence between (27) and (29) holds when Algorithm 1 converges. Thus, the converged solution of Algorithm 1 satisfies (27) [the KKT condition of the problem in (9)] and thus is a local minimum solution to problem (9). In this way, the proof of Theorem 2 is completed. □

Theorem 3: Algorithm 2 monotonically decreases the objective of problem (14) in each iteration.

Proof: Rewriting the problem in step 4 of Algorithm 2 by substituting the equality constraint into the objective gives

$$\begin{aligned} \mathbf{z}_i^{(p+1)} \ = \ \min_{\mathbf{z}_i} \ 0.5\mathbf{z}_i^T \mathbf{G}_i^T \mathbf{D}_i^{(p)} \mathbf{G}_i \mathbf{z}_i \\ + 0.5c\big(\mathbf{e} - \mathbf{F}_i^{(p)} \mathbf{H}_i \mathbf{z}_i\big)^T \mathbf{A}_i^{(p)} \big(\mathbf{e} - \mathbf{F}_i^{(p)} \mathbf{H}_i \mathbf{z}_i\big) \end{aligned} \tag{30}$$

which indicates

$$\begin{aligned}0.5&\mathbf{z}_i^{(p+1)^T}\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i\mathbf{z}_i^{(p+1)}\\&+0.5c\left(\mathbf{e}-\mathbf{F}_i^{(p)}\mathbf{H}_i\mathbf{z}_i^{(p+1)}\right)^T\mathbf{A}_i^{(p)}\left(\mathbf{e}-\mathbf{F}_i^{(p)}\mathbf{H}_i\mathbf{z}_i^{(p+1)}\right)\\\le\ &0.5\mathbf{z}_i^{(p)^T}\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i\mathbf{z}_i^{(p)}+\ 0.5c\left(\mathbf{e}-\mathbf{F}_i^{(p)}\mathbf{H}_i\mathbf{z}_i^{(p)}\right)^T\mathbf{A}_i^{(p)}\\&\times\left(\mathbf{e}-\mathbf{F}_i^{(p)}\mathbf{H}_i\mathbf{z}_i^{(p)}\right).\end{aligned}\tag{31}$$

According to step 3 of Algorithm 2, we can define $\boldsymbol{\xi}_i^{(p+1)}=\mathbf{e}-\mathbf{F}_i^{(p)}\mathbf{H}_i\mathbf{z}_i^{(p+1)}$ and $\boldsymbol{\xi}_i^{(p)}=\mathbf{e}-\mathbf{F}_i^{(p)}\mathbf{H}_i\mathbf{z}_i^{(p)}$. The former is an update of the latter. Applying the definitions of $\mathbf{D}_i^{(p)}$ and $\mathbf{A}_i^{(p)}$ in step 1 and step 3, problem (31) can be rewritten as the following one by decoupling the computation for each row for $\mathbf{D}_i^{(p)}$ and $\mathbf{A}_i^{(p)}$:

$$\begin{aligned}0.5&\left(\sum_j\frac{\left(\mathbf{g}_{i,j}\mathbf{z}_i^{(p+1)}\right)^2}{\left|\mathbf{g}_{i,j}\mathbf{z}_i^{(p)}\right|}+c\sum_j\frac{\left(\xi_{i,j}^{(p+1)}\right)^2}{\left|\xi_{i,j}^{(p)}\right|}\right)\\&\le 0.5\left(\sum_j\frac{\left(\mathbf{g}_{i,j}\mathbf{z}_i^{(p)}\right)^2}{\left|\mathbf{g}_{i,j}\mathbf{z}_i^{(p)}\right|}+c\sum_j\frac{\left(\xi_{i,j}^{(p)}\right)^2}{\left|\xi_{i,j}^{(p)}\right|}\right).\end{aligned}\tag{32}$$

Using inequality (21) in the proof of Theorem 1 and adding it to (32) gives

$$\left\|\mathbf{G}_i\mathbf{z}_i^{(p+1)}\right\|_1+\ 0.5c\sum_j\frac{\left(\xi_{i,j}^{(p+1)}\right)^2}{\left|\xi_{i,j}^{(p)}\right|}\le\left\|\mathbf{G}_i\mathbf{z}_i^{(p)}\right\|_1+0.5c\sum_j\frac{\left(\xi_{i,j}^{(p)}\right)^2}{\left|\xi_{i,j}^{(p)}\right|}.\tag{33}$$

With the similar proof of (22), it is easy to conclude that

$$c\sum_j\left(|\xi_{i,j}^{(p+1)}|-0.5\frac{\left(\xi_{i,j}^{(p+1)}\right)^2}{\left|\xi_{i,j}^{(p)}\right|}\right)\le c\sum_j\left(|\xi_{i,j}^{(p)}|-0.5\frac{\left(\xi_{i,j}^{(p)}\right)^2}{\left|\xi_{i,j}^{(p)}\right|}\right).\tag{34}$$

Combining (33) and (34) leads to

$$\left|\left|\mathbf{G}_i\mathbf{z}_i^{(p+1)}\right|\right|_1+c\left|\left|\boldsymbol{\xi}_i^{(p+1)}\right|\right|_1\le\left|\left|\mathbf{G}_i\mathbf{z}_i^{(p)}\right|\right|_1+c\left|\left|\boldsymbol{\xi}_i^{(p)}\right|\right|_1.\tag{35}$$

Algorithm 2 converges, since the problem in (14) has a lower bound 0. In such a case, the equality in (35) holds. Therefore, the objective value of problem (14) decreases in each iteration till the algorithm converges. □

Theorem 4: Algorithm 2 will converge to a local minimal solution to problem (14).

Proof: Substituting the equality constraints of (14) into the objective function gives $\min_{\mathbf{z}_i}\ \left|\left|\mathbf{G}_i\mathbf{z}_i\right|\right|_1+c\left\|\mathbf{e}-\left|\mathbf{H}_i\mathbf{z}_i\right|\right\|_1$, which is rewritten as

$$\min_{\mathbf{z}_i}\ \sum_j|\mathbf{g}_{i,j}\mathbf{z}_i|+c\sum_j|1-|\mathbf{h}_{i,j}\mathbf{z}_i||\tag{36}$$

where $\mathbf{h}_{i,j}$ denotes the j-th row of $\mathbf{H}_i$. Define the Lagrangian function of (36) as $\mathcal{L}(\mathbf{z}_i)$. Taking the derivative of $\mathcal{L}(\mathbf{z}_i)$ with respect to $\mathbf{z}_i$ and setting it as 0, we get the KKT condition of problem (36) as follows:

$$\begin{aligned}\sum_j\text{sign}(\mathbf{g}_{i,j}\mathbf{z}_i)\mathbf{g}_{i,j}^T+c\sum_j&\text{sign}(1-|\mathbf{h}_{i,j}\mathbf{z}_i|)\\&\times(-\text{sign}(\mathbf{h}_{i,j}\mathbf{z}_i))\mathbf{h}_{i,j}^T=0.\end{aligned}\tag{37}$$

In each iteration of Algorithm 2, the optimal solution to the problem in step 4 is found. Therefore, the converged solution satisfies the KKT condition of the problem. Rewrite the problem in step 4 of Algorithm 2 as

$$\begin{aligned}\mathbf{z}_i^{(p+1)}=\min_{\mathbf{z}_i}\ \mathbf{z}_i^T&\left(\sum_j d_{j,j}^{(p)}\mathbf{g}_{i,j}^T\mathbf{g}_{i,j}\right)\mathbf{z}_i\\&+c\sum_j a_{j,j}^{(p)}(1-\mathbf{f}_{jj}^{(p)}\mathbf{h}_{i,j}\mathbf{z}_i)^2\end{aligned}\tag{38}$$

where $\mathbf{f}_{j,j}^{(p)}=\text{sign}(\mathbf{h}_{i,j}\mathbf{z}_i)$ is the j-th diagonal element of $\mathbf{F}_i^{(p)}$. Forming the Lagrangian function of the objective of (38) and setting its derivative with respect to $\mathbf{z}_i$ as zero, one gets

$$\sum_j d_{jj}^{(p)}\mathbf{g}_{i,j}^T\mathbf{g}_{i,j}\mathbf{z}_i+c\sum_j a_{jj}^{(p)}\mathbf{h}_{ij}^T\mathbf{f}_{jj}^{(p)^T}(\mathbf{f}_{jj}^{(p)}\mathbf{h}_{ij}\mathbf{z}_i-1)=0.\tag{39}$$

Since $\mathbf{g}_{i,j}\mathbf{z}_i$ satisfies $\mathbf{g}_{i,j}\mathbf{z}_i=\text{sign}(\mathbf{g}_{i,j}\mathbf{z}_i)|\mathbf{g}_{i,j}\mathbf{z}_i|$ and $d_{jj}=1/|\mathbf{g}_{i,j}\mathbf{z}_i|$, we have

$$\sum_j d_{j_j}\mathbf{g}_{i,j}^T\mathbf{g}_{i,j}\mathbf{z}_i=\sum_j\text{sign}(\mathbf{g}_{i,j}\mathbf{z}_i)\mathbf{g}_{i,j}^T.\tag{40}$$

Similarly, we have $\mathbf{f}_{j,j}\mathbf{h}_{i,j}\mathbf{z}_i-1=\text{sign}(\mathbf{f}_{j,j}\mathbf{h}_{i,j}\mathbf{z}_i-1)|\mathbf{f}_{j,j}\mathbf{h}_{i,j}\mathbf{z}_i-1|$. Since $\mathbf{f}_{j,j}=\text{sign}(\mathbf{h}_{i,j}\mathbf{z}_i)$ and $a_{j,j}=1/|\mathbf{f}_{j,j}\mathbf{h}_{i,j}\mathbf{z}_i-1|$ according to step 3 of Algorithm 2, we achieve that

$$\begin{aligned}c&\sum_j a_{j,j}\mathbf{h}_{i,j}^T\mathbf{f}_{j,j}^T(\mathbf{f}_{j,j}\mathbf{h}_{i,j}\mathbf{z}_i-1)\\&=c\sum_j\mathbf{h}_{i,j}^T\text{sign}(\mathbf{h}_{i,j}\mathbf{z}_i)\text{sign}(\text{sign}(\mathbf{h}_{i,j}\mathbf{z}_i)\mathbf{h}_{i,j}\mathbf{z}_i-1)\\&=c\sum_j\text{sign}(1-|\mathbf{h}_{i,j}\mathbf{z}_i|)\text{sign}(-\mathbf{h}_{i,j}\mathbf{z}_i)\mathbf{h}_{i,j}^T.\end{aligned}\tag{41}$$

Using the three equalities in (39)–(41), it is easy to observe that problems (37) and (39) are the same when Algorithm 2 converges. This implies that the converged solution satisfies (37) and thus is a local minimal solution to problem (14). □

Suppose that $\mathbf{X}_i$ and $\overline{\mathbf{X}}_i$ are given. Problem (7) shares the same formulation of TWSVM; thus, its time complexity is the same as that of TWSVM, which is mainly dominated by two parts: solving a CQPP and matrix inverse. The time complexity of solving the CQPP in (7) is no more than $m^3/4$ when $\mathbf{X}_i$ is equivalent to $\overline{\mathbf{X}}_i$ in the number of samples [12]. The time complexity of computing the inverse of $\mathbf{G}_i^T\mathbf{G}_i$ is around n^3. Thus, the total time complexity of solving problem (7) is around $m^3/4+n^3$. TWSVC costs no more than $t(m^3/4+n^3)$ time complexity to solve problem (6), where t is the iterative number of the CCCP. Likewise, Algorithm 1 is used to solve CQPP (11) of RTWSVC iteratively that is similar to (7); thus, its time complexity is $s(m^3/4+n^3)$, where s is the iterative number. In contrast, FRTWSVC solves a system of linear equations in each iteration, whose time complexity is around $\ln^3$, which is mainly dominated by the inverse computation of matrix $1/c\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i+\mathbf{H}_i^T\mathbf{A}_i^{(p)}\mathbf{H}_i$, where l is the iterative number. Without loss of generality, the iterative number of each algorithm is by far less than the number of samples. Since TWSVC and RTWSVC have cubic time complexity in the number of samples, FRTWSVC performs faster in the case of $m\gg n$. This is also supported by latter experimental results. It should be pointed out that in clustering problems, when one is confronted with high-dimensional data (or curse of dimensionality), a common practice is to use a dimension-reduction technique to avoid this problem [29]. In this way, the clustering techniques finally handle the data sets with $m\gg n$.

The matrix $1/c\mathbf{G}_i^T\mathbf{D}_i^{(p)}\mathbf{G}_i+\mathbf{H}_i^T\mathbf{F}_i^{(p)}\mathbf{A}_i^{(p)}\mathbf{F}_i^{(p)}\mathbf{H}_i$ could be rank deficient in real problems. Note that the rank-deficiency problem also appears in TWSVC. Following [16] and [17], we introduce a regularization term $\varepsilon\mathbf{I}$ to address this problem, where ε is a small perturbation and $\mathbf{I}$ an identity matrix.

TABLE I

CLUSTERING PERFORMANCE AND COMPUTING TIME OF LINEAR K MEANS ON THE BENCHMARK DATA SETS

Leaf	Userknow	Tae	Cleveland	Dermatoloy	Ecoli	Wine	Haberman	Iris	Glass	Vowel	Zoo	Mush	Ahythmia	Yale	Mist	Isolet1	Mean Acc.
93.53	70.81	55.53	73.27	70.58	87.37	71.87	63.21	87.97	59.21	86.05	89.94	50.51	63.27	85.77	87.43	94.58	75.92
0.047	0.0139	0.0068	0.0147	0.0102	0.012	0.016	0.4991	0.018	0.011	0.020	0.009	0.0726	0.3325	0.3367	8.5794	5.2811	

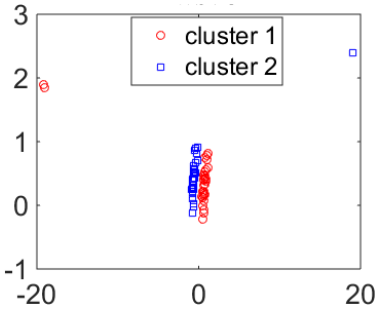

Fig. 1. True distribution of the synthetic data set.

B. Nonlinear RTWSVC and FRTWSVC

We extend RTWSVC and FRTWSVC to the nonlinear cases via the application of kernel trick. Similar to [14], [16] and [17], nonlinear RTWSVC and FRTWSVC find the following k-kernel-generated cluster surfaces instead of planes:

$$\mathcal{K}_{P_i} = \mathcal{K}(\mathbf{x}^T,\ \mathbf{X}^T)\mathbf{u}_i^T + b_i = 0, \quad i = 1, \ldots, k \tag{42}$$

where $\mathcal{K}$ is an appropriately selected kernel, and $\mathbf{u}_i \in \mathbf{R}^m$. By using the linear kernel $\mathcal{K}(\mathbf{x}^T,\ \mathbf{X}^T) = \mathbf{x}^T\mathbf{X}^T$ and defining $\mathbf{w}_i = \mathbf{X}\mathbf{u}_i$, (42) becomes (1). This means that the cluster planes can be obtained as a special case of (42).

In line with the arguments in Section III-A, in each cluster update, our nonlinear RTWSVC and FRTWSVC problems generalize (8) and (13), respectively, to

(RTWSVC)

$$\min_{\mathbf{u}_i,\ \mathbf{X}_i, \overline{\mathbf{X}}_i, b_i} \ ||\mathcal{K}(\mathbf{X}_i,\ \mathbf{X}^T)\mathbf{u}_i + \mathbf{e}b_i||_1 + c\mathbf{e}^T\boldsymbol{\xi}_i$$
$$\text{s.t. } |\mathcal{K}(\overline{\mathbf{X}}_i,\ \mathbf{X}^T)\mathbf{u}_i + \mathbf{e}b_i| + \boldsymbol{\xi}_i \geq \mathbf{e}, \boldsymbol{\xi}_i \geq 0, \quad i = 1, \ldots, k \tag{43}$$

(FRTWSVC)

$$\min_{\mathbf{u}_i,\ \mathbf{X}_i, \overline{\mathbf{X}}_i, b_i} \ ||\mathcal{K}(\mathbf{X}_i,\ \mathbf{X}^T)\mathbf{u}_i + \mathbf{e}b_i||_1 + c||\boldsymbol{\xi}_i||_1$$
$$\text{s.t. } |\mathcal{K}(\overline{\mathbf{X}}_i,\ \mathbf{X}^T)\mathbf{u}_i + \mathbf{e}b_i| + \boldsymbol{\xi}_i = \mathbf{e}, \quad i = 1, \ldots, k. \tag{44}$$

The two problems can be solved using the same iterative procedures of Algorithms 1 and 2, respectively. Since $\mathcal{K}(\mathbf{X}_i,\ \mathbf{X}^T)$ and $\mathcal{K}(\overline{\mathbf{X}}_i,\ \mathbf{X}^T)$ are in $\mathbf{R}^{m_i \times m}$ and $\mathbf{R}^{(m-m_i)\times m}$, the two methods, like TWSVC, require inverse of matrix with size $m \times m$. In practice, the rectangular kernel technique [26] can be applicable to reduce the dimensions of the matrices, as done in [14], [16] and [17].

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed RTWSVC and FRTWSVC, we conduct experiments on a synthetic data set and several benchmark data sets [14], [27]. Fig. 1 plots the true distribution of the synthetic data set. The synthetic data set is constructed as follows. We generated 2-D datapoints under the Gaussian distribution, which belong to two clusters and are distinguished by "o" and "□." We inserted two and one outliers into the clusters "o" and "□," respectively. The same experiments are also conducted by K means, PPC, TWSVC, RTWSVC, and FRTWSVC.

To measure the clustering performance, we use the metric accuracy, which is defined in [14]. Specifically, given the cluster labels y_i,

TABLE II

CLUSTERING PERFORMANCE AND COMPUTING TIME OF LINEAR kPC, PPC, TWSVC, RTWSVC, AND FRTWSVC ON THE BENCHMARK DATA SETS ("–" MEANS THAT WE STOP EXPERIMENTS AS COMPUTING TIME IS VERY HIGH)

Dataset m×n	kPC Acc./Time	PPC Acc./Time	TWSVC Acc./Time	RTWSVC Acc./Time	FRTWSVC Acc./Time
Leaf 340×15	89.78/0.0314	67.89/0.2096	**94.56**/183.83	94.25/ 251.57	93.97/84.942
Userknow 403×5	62.09/0.0136	68.33/0.0195	61.17/43.116	**72.26**/199.92	70.69/1.7766
Tae 151×5	44.79/0.0015	55.46/0.0071	55.96/0.8767	**59.51**/3.2963	56.17/0.3387
Cleveland 297×13	51.43/0.0108	51.95/0.0132	60.51/2.0545	**63.14**/ 5.6017	61.46/0.1214
Dermatoloy 366×34	60.50/0.0152	60.50/0.0822	70.13/93.478	**71.40**/ 89.851	70.82/14.576
Ecoli 336×7	42.55/0.0062	77.89/0.0503	77.89/39.192	79.18/ 56.721	**79.91**/2.5905
Wine 178×13	57.93/0.0037	73.17/0.0062	73.65/3.8322	89.08/7.3394	**92.84**/1.4722
Haberman 306×3	55.86/0.0039	60.95/0.0071	61.26/1.2151	**63.28**/ 3.1464	61.26/0.2945
Iris 150×4	67.54/0.0033	83.68/0.0040	91.24/0.9046	91.95/4.6280	**95.75**/0.5503
Glass 214×9	68.70/0.0112	65.71/0.0378	65.56/11.088	**70.98**/ 17.638	63.75/1.6875
Vowel 528×10	83.91/0.0335	83.32/0.0717	83.37/949.15	81.87/1201.9	**84.25**/61.179
Zoo 101×16	55.39/0.0017	81.56/0.0492	**88.91**/2.2886	88.77/ 1.0188	88.59/2.2623
Mush 8124×22	89.40/15.742	**89.49**/2.2023	89.45/7544.1	**89.49**/9279.6	**89.49**/320.12
Ahythmia 452×280	42.78/4.8683	45.00/7.3952	77.31/45.375	88.27/633.34	**91.79**/961.03
Yale 165×1024	59.73/196.62	36.32/119.85	**88.80**/73.288	**88.80**/255.31	**88.80**/895.49
Mist 4000×784	**100.0**/31.914	**100.0**/170.60	–	--	**100.0**/2091.8
Isolet1 1560×617	88.47/2650.7	87.55/3179.5	77.28/1152.3	85.19/4564.7	**93.54**/9024.8
Mean Acc.	65.98	70.04	76.15	79.82	**81.38**

the corresponding similarity matrix $\mathbf{M} \in \mathbf{R}^{m\times m}$ is easy to be computed, where

$$\mathbf{M}(i, j) = \begin{cases} 1, & \text{if } y_i = y_j \\ 0, & \text{otherwise.} \end{cases}$$

Suppose $\mathbf{M}_t$ and $\mathbf{M}_p$ are two similarity matrices, which are, respectively, computed by the truth cluster labels of the data and the prediction of a clustering method. The metric accuracy of the clustering method is defined as the Rand statistic Accuracy $= (n_{00}+n_{11}-m)/(m^2 - m) \times 100\%$, where n_{00} and n_{11} are the respective number of 0s and 1s in $\mathbf{M}_t$ and $\mathbf{M}_p$. All these methods require selecting the initial cluster labels. Following [14], the initial cluster labels of each method are selected using the effective nearest neighbor graph (NNG)-based initialization. For a linear case, PPC, TWSVC, RTWSVC, and FRTWSVC have two common parameters c and p (neighborhood size in the NNG-based initialization). The parameter c is selected from the values $\{2^i | i = -5, -4, \ldots, 4\}$, while p is selected from the values $\{1, 2, \ldots, 5\}$ as in [14]. The stopping tolerance of TWSVC, RTWSVC, and FRTWSVC is set as the difference between two successive iterations less than 0.001. In using TWSVM, RTWSVC, and FRTWSVC, the initial cluster plane is set as the solution of PPC.

Fig. 2 depicts the result of each method on the synthetic data set. p is simply set as 1. We can see from Fig. 2 that RTWSVC and FRTWSVC are far better than other methods, which obtain 93.75% clustering accuracy. Although, TWSVC is better than kPC, it only achieves 52.07% clustering accuracy. This indicates the robustness

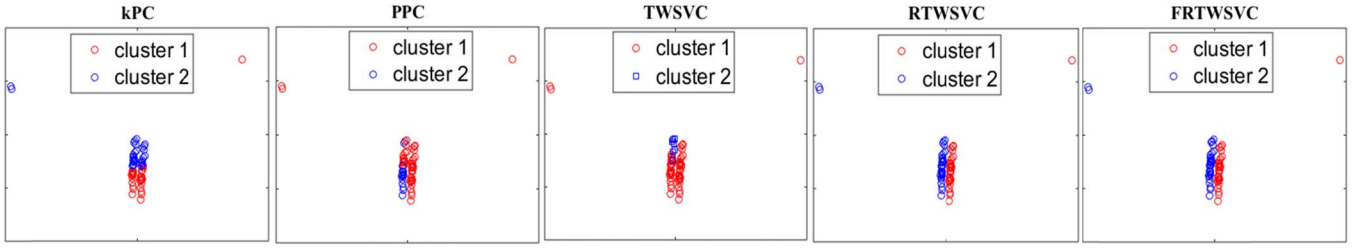

Fig. 2. Clustering result of each method on the synthetic data set.

TABLE III
CLUSTERING PERFORMANCE AND COMPUTING TIME OF NONLINEAR K MEANS ON THE BENCHMARK DATA SETS

Leaf	Userknow	Tae	Cleveland	Dermatoloy	Haberman	Iris	Glass	Vowel	Zoo	Ahythmia	Yale	Mean Acc.
92.50	74.23	56.41	73.27	70.73	61.26	88.59	67.22	87.33	90.26	47.29	82.05	74.26
0.0241	0.414	0.0046	0.0143	0.0172	0.0162	0.0127	0.0128	0.0408	0.0089	0.0089	0.0044	

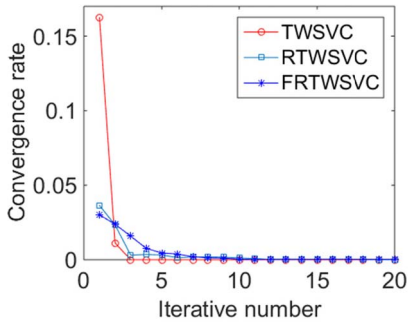

Fig. 3. Convergence rate of TWSVC, RTWSVC, and FRTWSVC.

TABLE IV
CLUSTERING PERFORMANCE AND COMPUTING TIME OF NONLINEAR *k*PC, PPC, TWSVC, RTWSVC, AND FRTWSVC ON THE BENCHMARK DATA SETS

Dataset m×n	*k*PC Acc./Time	PPC Acc./Time	TWSVC Acc./Time	RTWSVC Acc./Time	FRTWSVC Acc./Time
Leaf 340×15	92.32/1.131	91.21/6.595	94.38/196.7	**94.47**/246.8	93.44/2.357
Userknow 403×5	66.97/0.474	65.23/1.001	69.85/177.8	**72.25**/196.5	71.68/12.54
Tae 151×5	55.39/0.075	53.17/0.078	57.92/1.587	57.18/2.219	**58.44**/0.628
Cleveland 297×13	70.58/0.292	71.90/0.507	63.63/3.255	70.58/7.459	**75.63**/2.736
Dermatoloy 366×34	69.44/1.089	71.55/1.541	**73.21**/46.43	72.26/56.74	71.43/14.56
Haberman 306×3	62.21/0.462	61.57/0.841	62.87/1.015	61.26/ 7.595	**63.55**/0.335
Iris 150×4	88.82/0.019	93.41/0.121	89.23/0.873	90.99/4.542	**96.56**/0.603
Glass 214×9	66.09/0.138	64.04/0.268	70.25/15.68	**71.88**/ 31.35	68.49/6.167
Vowel 528×10	**85.72**/2.152	85.41/6.862	84.64/1052.3	84.42/1252.8	83.12/138.6
Zoo 101×16	88.79/0.066	88.95/0.094	89.96/2.357	89.64/3.387	**90.01**/1.886
Ahythmia 452×280	65.46/6.089	62.31/1.9992	69.53/442.16	71.49/521.5	7 **72.03**/241.3
Yale 165×1024	90.01/0.4982	80.21/12.5133	90.98/94.689	**91.92**/299.83	89..36/30.04
Mean Acc.	75.14	74.38	75.60	76.50	**77.23**

of TWSVC and FRTWSVC. Tables I and II show the clustering performance and computing time of linear K means, *k*PC, PPC, TWSVC, RTWSVC, and FRTWSVC. Note that the computing time reported is the average of times to learn all the sets of clustering planes under the combinations of c and p.

From Tables I and II, we first observe that RTWSVC and FRTWSVC yield better accuracy than other methods. Second, the computational advantage of our FRTWSVC over TWSVC and RTWSVC is extremely obvious in the case of $m \gg n$. As seen, on Mush data, TWSVC and RTWSVC take 7554.1 and 9279.6 s, respectively, while FRTWSVC just takes 320.12 s. On the high-dimensional data sets that do not satisfy $m \gg n$, such as Yale, Ahythmia, and Isolet1, FRTWSVC is generally slower than both TWSVC and RTWSVC. As in the previous analysis, on such data sets, FRTWSVC may share similar time cost with that of TWSVC and RTWSVC at each iteration, which, however, is much more expensive. This attributes to the need for a more iterative number. RTWSVC generally runs slower than TWSVC, since it, like RTWSVC, needs a more iterative number. Fig. 3 plots the convergence rates of TWSVC, RTWSVC, and FRTWSVC on Ecoli when $\mathbf{X}_1$ and $\overline{\mathbf{X}}_1$ are given. As shown, TWSVC converges to zero in five iterations, while RTWSVC and FRTWSVC converge to zero in ten iterations. As analyzed previously, RTWSVC, like TWSVC, solves a series of CQPPs. In such cases, naturally RTWSVC performs slower than TWSVC. It should be pointed out that although the dimensionality of the data set "Mist" is very high, it satisfies $m \gg n$. Therefore, on Mist, FRTWSVC still runs far faster than RTWSVC and TWSVC (in the experiment, we stop running RTWSVC and TWSVC, since they cannot complete the entire learning within seven days). In real applications, a common way to manipulate high-dimensional data, such as clustering, is to resort to a dimension-reduction technique to avoid the "curse of dimensionality" problem beforehand [29]. In this way, the clustering techniques finally handle the data sets with $m \gg n$. Anyway, our results likewise show that FRTWSVC and RTWSVC are better than TWSVC when directly coping with the original high-dimensional data. Finally, we can observe that TWSVM outperforms *k*PC and PPC in terms of clustering accuracy, which is consistent with [14].

Owing to page constraints, we only depict the clustering accuracy of FRTWSVC versus the variations of the parameters c and p in Fig. 4. Despite this, we find that RTWSVC has a similar result in our experiments. From Fig. 4, we see that our FRWSVC obtains significantly better clustering accuracy on most data sets when p and c are set to be a smaller number and a larger number, respectively.

Tables III and IV compare nonlinear K means, *k*PC, PPC, TWSVC, RTWSVC, and FRTWSVC with a Gaussian kernel on 12 benchmark data sets. All the k-plane methods have three common parameters:

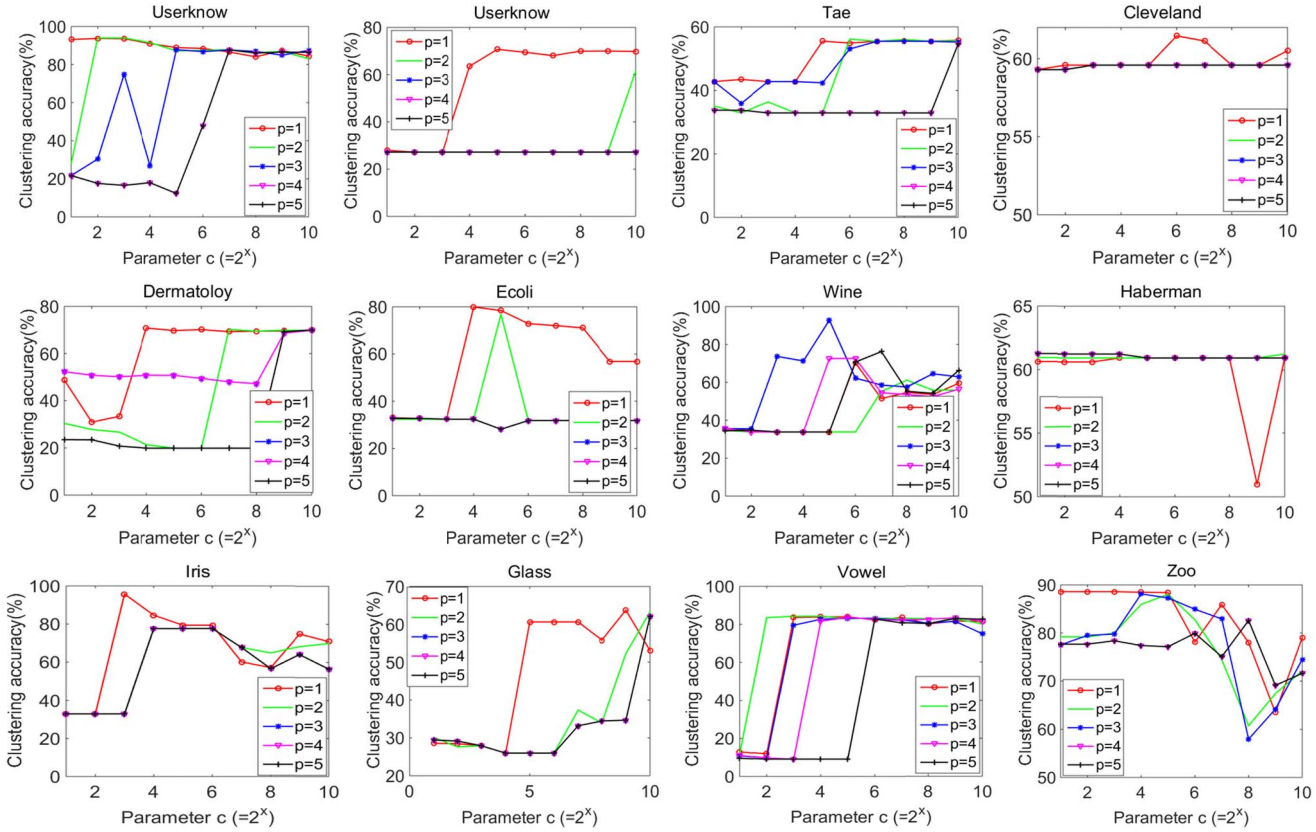

Fig. 4. Clustering accuracy of FRTWSVC versus the variations of the parameters c and p.

TABLE V
CLUSTERING PERFORMANCE AND COMPUTING TIME OF K MEANS ON THE NOISY BENCHMARK DATA SETS

Leaf	Userknow	Tae	Cleveland	Dermatoloy	Ecoli	Haberman	Iris	Glass	Vowel	Zoo	Mush	Ahythmia	Yale	Isolet1	Mean Acc.
89.53	70.19	53.62	59.64	68.75	82.70	50.78	88.59	68.36	86.60	84.68	61.90	65.30	86.27	90.83	73.85
0.0106	0.0186	0.0056	0.0062	0.0277	0.0167	0.0089	0.0085	0.0182	0.0318	0.0054	0.6194	0.0061	0.0036	0.7665	

c, p, and σ. The parameters p and σ exist in K means. In the experiments, we select c and σ from the values $\{2^i | i = -5, -4, \ldots, 4\}$ and p from the values $\{1, 2, \ldots, 5\}$. The stopping tolerance of the nonlinear TWSVC, RTWSVC, and FRTWSVC is 0.01. From the results of Tables III and IV, we see first that TWSVC, RTWSVC, and FRTWSVC outperform other methods in clustering accuracy, second that RTWSVC and FRTWSVC obtain higher accuracy than other methods, and third that TWSVC and RTWSVC run far slower than FRTWSVC. On the whole, these observations are consistent with those drawn from the previous experiments. In addition, we find that most of these nonlinear methods outperform their linear versions on most data sets, demonstrating that the introduction of kernel can promote the performance. Furthermore, FRTWSVC is slightly better than RTWSVC, which is not true in the last experiment. This indicates that there is no consistent winner when comparing RTWSVC and FRTWSVC.

The relations between the parameters and the clustering accuracy of our nonlinear FRTWSVC are shown in Fig. 5. Considering the limit of pages, only the results of five data sets are shown. It can be first seen from Fig. 5 that nonlinear FRTWSVC performs well in the case of $p \leq 2$, which is consistent with the linear FRTWSVC. Second, the parameter $c < 1$ is a good option for most data sets. Finally, the kernel parameter σ significantly affects the accuracy of nonlinear FRTWSVC. On Zoo and Userknow, $\sigma > 1$ makes the nonlinear FRTWSVC perform well, and on other three data sets, it generally performs better in the case of $\sigma > 1$.

To illustrate the robustness of each method, we corrupt the training set using a noise matrix $\mathbf{N}_o$ whose element is *i.i.d.* standard Gaussian variables [28]. Given the training set $\mathbf{X}$, each method learns on the corrupted training set $\mathbf{X} + \upsilon \mathbf{N}_o$, where $\upsilon = \kappa ||\mathbf{X}||_F / ||\mathbf{N}_o||_F$ is a given noise factor. In the experiment, κ is set as 0.2. Tables V and VI report the clustering performance and computing time of linear kPC, PPC, TWSVC, RTWSVC, and FRTWSVC. As it can be seen, on most data sets, the performance of each method is more or less impaired by the corruption. Even so, our RTWSVC and FRTWSVC significantly outperform other methods in most cases. In computational cost, TWSVC and RTWSVC are inferior to FRTWSVC. Considering both accuracy and efficiency, FRTWSVC is the best choice among all the compared methods.

In the previous experiments, the cluster number k is set to the true class size (z). This setting follows that of TWSVC in [14]. Therefore, the comparisons are made in a fair way. Despite this, it is interesting to discuss the variation of clustering accuracy versus k. For this purpose, we plot the clustering accuracy of linear kPC, PPC, TWSVC, RTWSVC, and FRTWSVC on ten data sets versus the different cluster number k, as shown in Fig. 6. The maximum k is defined as about $3z$. From Fig. 6, each method can generally obtain very good accuracy when k equals z. We see that the best

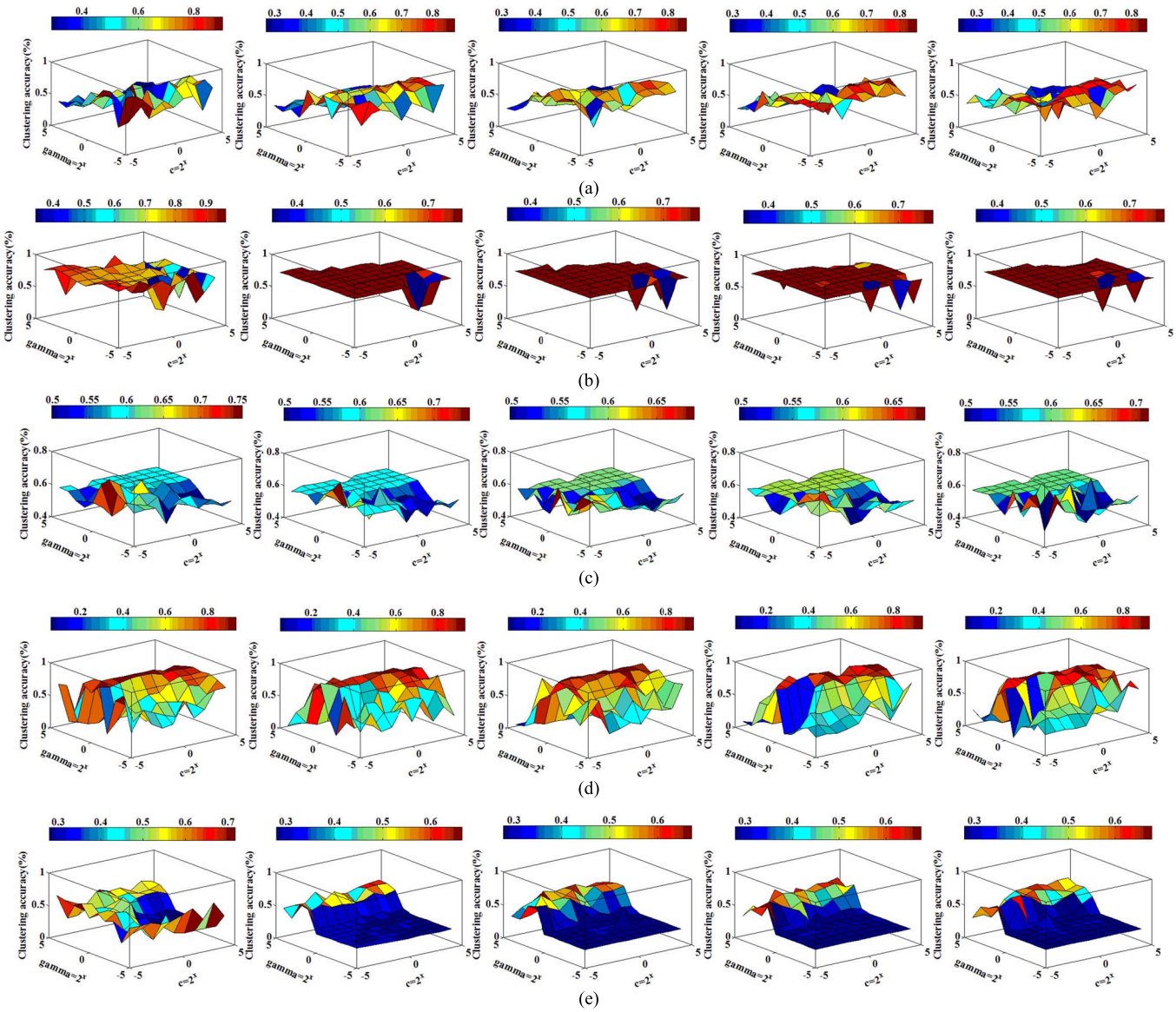

Fig. 5. Clustering accuracy of nonlinear FRTWSVC with different parameters on (a) zoo, (b) iris, (c) cleveland, (d) leaf, and (e) userknow.

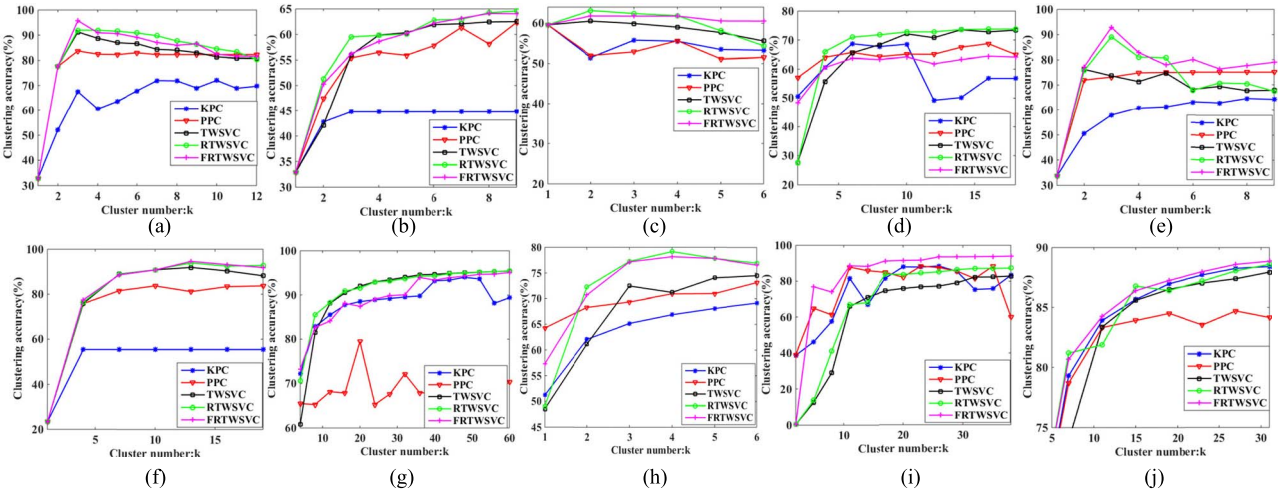

Fig. 6. Clustering accuracy versus different cluster numbers k. (a) Iris. (b) Tae. (c) Cleveland. (d) Glass. (e) Wine. (f) Zoo. (g) Leaf. (h) Userknow. (i) Isolet1. (j) Vowel.

accuracy of TWSVC is higher than PPC and kPC on most cases, and the performance advantage of TWSVC is obvious. From the comparisons, our FRTWSVC and RTWSVC have better results than TWSVC in most cases. Furthermore, on some data sets, such as Leaf, FRTWSVC obtains almost the same result at large value of k, although it is inferior to TWSVC in the case of $k = z$.

TABLE VI

CLUSTERING PERFORMANCE AND COMPUTING TIMES OF kPC, PPC, TWSVC, RTWSVC, AND FRTWSVC ON THE NOISY DATA SETS

Dataset m×n	kPC Acc./Time	PPC Acc./Time	TWSVC Acc./Time	RTWSVC Acc./Time	FRTWSVC Acc./Time
Leaf 340×15	93.79/0.0237	79.87/0.2297	94.46/91.098	**94.54**/105.73	91.27/58.493
Userknow 403×5	58.09/0.0161	65.30/0.0161	67.30/37.127	67.51/ 54.818	**68.20**/1.0213
Tae 151×5	50.38/0.0028	54.84/0.0093	52.49/2.3602	**57.65**/ 3.3254	55.77/0.2914
Cleveland 297×13	50.31/0.0183	**63.14**/0.0110	59.89/1.6886	57.09/7.0914	59.89/0.2094
Dermatoloy 366×34	70.21/0.0628	62.58/0.1736	70.14/84.353	**71.41**/90.975	70.25/8.9631
Ecoli 336×7	66.46/0.0315	68.22/0.0538	67.17/46.651	69.74/125.23	**71.19**/6.4913
Haberman 306×3	49.89/0.0043	60.95/0.0053	**61.57**/1.7351	60.64/ 1.4874	60.95/0.1739
Iris 150×4	64.54/0.0032	81.68/0.0073	77.77/0.8918	79.93/2.3106	**84.15**/0.1172
Glass 214×9	66.43/0.0069	68.08/0.0305	64.85/12.617	**69.18**/ 17.663	66.96/1.6836
Vowel 528×10	83.11/0.0377	80.32/0.0377	82.77/817.34	**84.23**/1211.3	83.91/32.247
Zoo 101×16	79.86/0.0067	78.30/0.0544	84.79/0.6326	85.05/2.5682	**89.57**/0.6326
Mush 8124×22	50.27/10.169	88.88/2.026	89.47/7215.8	**89.49**/4878.5	**89.49**/311.89
Arrhythmia 452×280	64.52/40.532	41.78/16.091	58.70/27.989	72.59/280.79	**82.41**/3857.2
Yale 165×1024	74.75/195.779	73.47/132.044	**87.49**/48.299	**87.49**/234.16	**87.49**/1208.9
Isolet1 1560×617	85.14/322.43	49.16/212.21	83.31/742.28	86.31/1360.2	**88.12**/5323.6
Mean Acc.	67.22	67.66	73.64	75.69	**76.91**

V. CONCLUSION

We presented two k-plane clustering methods, called RTWSVC and FRTWSVC. The objective of RTWSVC was formulated using robust L1-norm distance. To achieve this, we derived an effective iterative algorithm where a series of constrained quadratic convex programming problems need to be solved. To reduce the computational cost of RTWSVC, FRTWSVC was further proposed. The objective was achieved by a newly designed iterative algorithm. In each round of optimization of the algorithm, we solved only a system of linear equations. We presented some insightful analysis on the existence of local minimum and the convergence of the algorithms. Furthermore, we also generalized the RTWSVC and FRTWSVC methods to handle the nonlinear k-plane clustering problems. The experimental results on benchmark data sets indicated that: 1) both methods yield better accuracy than the existing k-plane clustering methods and 2) FRTWSVC runs far faster than TWSVC and RTWSVC.

REFERENCES

[1] M. W. Berry, Ed., *Survey of Text Mining: Clustering, Classification, and Retrieval*, vol. 1. Berlin, Germany: Springer, 2004.

[2] A. Hotho, A. Nürnberger, and G. Paaß, "A brief survey of text mining," *J. Comput. Linguistics Lang. Technol.*, vol. 20, no. 1, pp. 19–62, 2005.

[3] S. Bandyopadhyay, A. Mukhopadhyay, and U. Maulik, "An improved algorithm for clustering gene expression data," *Bioinformatics*, vol. 23, no. 21, pp. 2859–2865, 2007.

[4] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

[5] H. Jia, Y.-M. Cheung, and J. Liu, "Cooperative and penalized competitive learning with application to kernel-based clustering," *Pattern Recognit.*, vol. 47, no. 9, pp. 3060–3069, 2014.

[6] Y.-F. Li, I. W. Tsang, J. Kwok, and Z.-H. Zhou, "Tighter and convex maximum margin clustering," in *Proc. Int. Conf. Artif. Intell. Stat.*, Clearwater Beach, FL, USA, 2009, pp. 344–351.

[7] H. Zeng and Y.-M. Cheung, "Semi-supervised maximum margin clustering with pairwise constraints," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 926–939, May 2012.

[8] L. Xu, J. Neufeldy, B. Larsony, and D. Schuurmansy, "Maximum margin clustering," in *Proc. Int. Conf. Adv. Neural Inf. Proces. Syst.*, Vancouver, BC, Canada, 2004, pp. 1537–1544.

[9] M. R. Anderberg, *Cluster Analysis for Applications*. New York, NY, USA: Academic, 1973.

[10] X. Huang, Y. Ye, and H. Zhang, "Extensions of kmeans-type algorithms: A new clustering framework by integrating intracluster compactness and intercluster separation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1433–1446, Aug. 2014.

[11] P. S. Bradley and O. L. Mangasarian, "Clustering via concave minimization," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, 1997, pp. 368–374.

[12] P. S. Bradley and O. L. Mangasarian, "k-plane clustering," *J. Global Optim.*, vol. 16, no. 1, pp. 23–32, 2000.

[13] Y.-H. Shao, L. Bai, Z. Wang, X.-Y. Hua, and N.-Y. Deng, "Proximal plane clustering via eigenvalues," *Procedia Comput. Sci.*, vol. 17, pp. 41–47, May 2013.

[14] Z. Wang, Y.-H. Shao, L. Bai, and N.-Y. Deng, "Twin support vector machine for clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2583–2588, Oct. 2015.

[15] O. L. Mangasarian and E. W. 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.

[16] 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, Jun. 2011.

[17] 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.

[18] N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1672–1680, Sep. 2008.

[19] F. Zhong, J. Zhang, and D. Li, "Discriminant locality preserving projections based on L1-norm maximization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 11, pp. 2065–2074, Nov. 2014.

[20] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann, "Kernel methods for missing variables," in *Proc. Int. Conf. Artif. Intell. Stat.*, Bridgetown, Barbados, 2005, pp. 325–332.

[21] Q. L. Ye, C. X. Zhao, H. F. Zhang, and X. B. Chen, "Recursive 'concave–convex' Fisher Linear Discriminant with applications to face, handwritten digit and terrain recognition," *Pattern Recognit.*, vol. 45, no. 1, pp. 54–65, 2012.

[22] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher, "Block-coordinate Frank–Wolfe optimization for structural SVMs," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 53–61.

[23] Y.-M. Cheung and J. Lou, "Efficient generalized conditional gradient with gradient sliding for composite optimization," in *Proc. Int. Conf. Artif. Intell.*, Austin, TX, USA, 2015, pp. 3409–3415.

[24] S. Ghorai, A. Mukherjee, S. Sengupta, and P. K. Dutta, "Cancer classification from gene expression data by NPPC ensemble," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 8, no. 3, pp. 659–671, May/Jun. 2011.

[25] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Appl. Note EE392o, 2003.

[26] Y.-J. Lee and O. L. Mangasarian, "RSVM: Reduced support vector machines," in *Proc. Int. Conf. Data Mining*, Chicago, IL, USA, 2000, pp. 1–17.

[27] C. Blake and C. Merz. (1998). *UCI Repository for Machine Learning Databases*. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[28] H. Wang, F. Nie, and H. Huang, "Learning robust locality preserving projection via p-order minimization," in *Proc. Int. Conf. Artif. Intell.*, Austin, TX, USA, 2015, pp. 3059–3065.

[29] C. Hou, F. Nie, D. Yi, and D. Tao, "Discriminative embedded clustering: A framework for grouping high-dimensional data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1287–1299, Jun. 2015.