IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 16, 2021, accepted January 31, 2021, date of publication February 5, 2021, date of current version February 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3057367

MPMSVC: Multiple Parametric-Margin Support Vector Clustering

YI-BO JIANG[1,2], WEI-JIE CHEN[1], YU-QING WANG[1,2], MING-CHUAN ZHANG[1,2], AND YUAN-HAI SHAO[3]
[1]Zhijiang College, Zhejiang University of Technology, Hangzhou 310024, China
[2]College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310024, China
[3]School of Economics and Management, Hainan University, Haikou 570228, China

Corresponding author: Wei-Jie Chen (wjcper2008@126.com)

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 LY21F030013 and Grant LY17F030015; and in part by the Natural Science Foundation of Hainan Province under Grant 118QN181.

ABSTRACT In this work, we propose an unsupervised multiple parametric-margin support vector clustering (MPMSVC) for noisy clustering tasks. The main idea of MPMSVC is to find a parametric-margin center hyperplane for each cluster in a manner that gathers the within-cluster instances around the corresponding center hyperplane, and keeps the between-cluster instances far away. Specifically, our MPMSVC owns the following attractive merits: i) The primal of MPMSVC is enhanced in the least squares sense, which enjoys an effective learning procedure. ii) The utilization of the linear L_1-norm loss makes MPMSVC be more robust to noisy clustering tasks. iii) An efficient iterative algorithm is presented to optimize the non-smooth problem in MPMSVC, which only involves a set of linear equations. Also, the convergence of the proposed algorithm is guaranteed theoretically. iv) The nonlinear extension is further derived via kernel technique to deal with more complex clustering tasks. Finally, the feasibility and effectiveness of MPMSVC is validated by extensive experiments on both synthetic and real-world datasets.

INDEX TERMS Plane-based clustering, nonparallel support vector clustering, L_1-norm, robustness.

I. INTRODUCTION

Clustering is one of mainstream topics in machine learning community [1]–[4]. Compared with the supervised classification, labels or outputs in clustering tasks are unknown. Thus, the main goal of clustering is to divide similar instances into the same cluster while dissimilar instances into the different ones, such that the meaningful underlying structures in data can be well exploited. During the last decade, the unsupervised clustering has been applied widely to various practical application domains, such as computer vision [5], [6], text mining [7], [8], smart grid [9], bioinformatics [10] and so on.

There are many branches [4], [11] in the field of clustering. Among them, the certain-based clustering is the most popular, which separates data into clusters according to certain cluster prototypes. Generally speaking, there are two kinds of certain-based clustering: the point-based and the plane-based. The point-based one assumes each cluster prototype is a point. That is, it partitions instances into K clusters, in which each instance belongs to the cluster with the nearest centroid or median. The representative methods are $k$means [12], $k$median [13], fuzzy c-means [14]. On the other hand, the plane-based clustering [11], [15] extends the cluster center from point to hyperplane. It is well known that the plane-based clustering is derived from the nonparallel hyperplane learning paradigm [16]–[26], which targets for seeking K optimal hyperplane prototypes to represent the cluster centers via some certain criterion.

The first plane-based clustering, k-plane clustering (kPC) [15], is originally proposed by Mangasarian. It aims to find K hyperplanes via considering the discriminative information from within-cluster. However, kPC only utilizes one side information, similarity for within-cluster. Motivated by GEPSVM [16], Shao *et al.* [27] introduce the dissimilarity for between-cluster, and propose proximal plane clustering (PPC). Both kPC and PPC optimize via solving eigenvalue problems. Subsequently, in light of TWSVM [17], Wang *et al.* [28] propose a novel unsupervised nonparallel hyperplane twin support vector clustering (TWSVC) by considering both similarity and dissimilarity. The optimal cluster plane-centers of TWSVC are obtained via solving SVM-type

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

quadratic programming problems (QPPs). Compared with kPC and PPC, TWSVC gains more solid theoretical results and achieves better performance on benchmark datasets.

However, TWSVC still has some limitations. To improve the model generalization, Bai *et al.* [29] put forward twin bounded support vector clustering (TBSVC) by introducing an additional regularization term. Meanwhile, similar to fuzzy c-means, Khemchandani *et al.* [30] present a fuzzy version of TWSVC (FLSTWSVC) with the soft assignments of clusters. Subsequently, to deal with noisy clustering cases, Ye *et al.* [31] consider a robust loss for TWSVC, and propose robust twin support vector clustering (RTWSVC) and its fast version (FRTWSVC). Beyond that, there are many other nonparallel hyperplane clustering extensions, including ramp-based TWSVC [32], least-square PTSVC [33], RFDPC [34], and so on [35]–[37].

Whereas in fact, the recently proposed twin parametric-margin support vector machine (TPMSVM) [23] is an excellent nonparallel classifier. Compared with TWSVM [17], TPMSVM can capture more complex heteroscedastic error structures via parametric-margin hyperplanes. Thus, motivated by the work on TPMSVM [23] and nonparallel plane-based clustering paradigm [28], [34], in this paper, we propose a novel unsupervised multiple parametric-margin support vector clustering for noisy clustering tasks, termed as MPMSVC. The main idea of MPMSVC is to generate K nonparallel parametric-margin cluster center hyperplanes, such that each parametric-margin hyperplane should be not only as close as possible to its current cluster instances but also far away from the other clusters. Specifically, our MPMSVC has the following attractive merits:

- By replacing the constraints in equalities with inequalities in TPMSVM, the primal of MPMSVC is enhanced in the least squares sense, which enjoys an effective learning procedure. (Section III)
- To improve the robustness to noisy clustering tasks, the L_1-norm based loss is further considered for MPMSVC. Moreover, an efficient iterative algorithm is designed to optimize the non-smooth problem of MPMSVC, whose convergence is guaranteed theoretically. (Section III)
- Formulation of MPMSVC is extended to nonlinear cases via kernel technique to deal with the more complex clustering tasks. (Section IV)
- Extensive experimental results on both noisy synthetic datasets and benchmark datasets confirm the effectiveness of MPMSVC in terms of RI (Rand Index) and learning time. (Section V)

The remainder of this paper is organized as follows. Section II briefly introduces notations and related works. Section III proposes the formulation of MPMSVC with the geometrical interpretation, and the feasibility of learning algorithm is also theoretically analyzed. The nonlinear extension is derived in Section IV. Experimental results are described in Section V, and Section VI gives concluding remarks and future works.

II. PRELIMINARIES

In this section, we briefly introduce the formulation of supervised TPMSVM [23], and clustering methods i.e., kPC [15], PPC [27] and TWSVC [28].

A. NOTATIONS

In this paper, scalars are denoted by lower case italic letters, vectors by lower case bold face letters, and matrices by capital face letters. All vectors will be column unless transformed to row vectors by a prime superscript $(\cdot)'$. Vectors of zeros and ones of arbitrary dimensions are represented by $\mathbf{0}$ and $\boldsymbol{e}$, respectively. Denote $\boldsymbol{I}$ as an identity matrix of arbitrary dimensions.

Clustering is an unsupervised learning task. The learning dataset is represented by $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m)'$ with m instances, where $\boldsymbol{x}_i \in \mathbb{R}^n$. Use $\mathcal{I}_k$ to express the set of indices for instances belonging to the kth cluster, where $k \in \{1, \cdots, K\}$. Denote $\boldsymbol{x}_{i\in\mathcal{I}_k}$ as the kth cluster's instance with its index $i \in \mathcal{I}_k$, while $\boldsymbol{x}_{j\in\mathcal{I}_{\bar{k}}}$ as the rest cluster's instance.

B. SUPERVISED TPMSVM

The TPMSVM [23] is originally proposed for binary classification tasks. It aims to find two nonparallel parametric-margin hyperplanes via the following optimization problems

$$\begin{aligned}\min_{\boldsymbol{w}_1,b_1}\ & \frac{1}{2}\|\boldsymbol{w}_1\|^2 + c_1\sum_{i\in\mathcal{I}_1}\xi_{1i} + c_2\sum_{j\in\mathcal{I}_2}\eta_{1j},\\ \text{s.t. }\ & \boldsymbol{w}_1'\boldsymbol{x}_i + b_1 \geq 0 - \xi_{1i},\quad \xi_{1i}\geq 0,\\ & \boldsymbol{w}_1'\boldsymbol{x}_j + b_1 = \eta_{1j},\end{aligned} \tag{1}$$

and

$$\begin{aligned}\min_{\boldsymbol{w}_2,b_2}\ & \frac{1}{2}\|\boldsymbol{w}_2\|^2 + c_1\sum_{i\in\mathcal{I}_2}\xi_{2i} - c_2\sum_{j\in\mathcal{I}_1}\eta_{2j},\\ \text{s.t. }\ & \boldsymbol{w}_2'\boldsymbol{x}_i + b_2 \leq 0 + \xi_{1i},\quad \xi_{2i}\geq 0,\\ & \boldsymbol{w}_2'\boldsymbol{x}_j + b_2 = \eta_{2j}.\end{aligned} \tag{2}$$

If set $\hat{\boldsymbol{w}}_2 = -\boldsymbol{w}_2$ and $\hat{b}_2 = -b_2$ in (2), we can unify the above formulation of TPMSVM as

$$\begin{aligned}\min_{\boldsymbol{w}_k,b_k}\ & \frac{1}{2}\|\boldsymbol{w}_k\|^2 + c_1\sum_{i\in\mathcal{I}_k}\xi_{ki} + c_2\sum_{j\in\mathcal{I}_{\bar{k}}}\eta_{kj},\\ \text{s.t. }\ & \boldsymbol{w}_k'\boldsymbol{x}_i + b_k \geq 0 - \xi_{ki},\quad \xi_{ki}\geq 0,\\ & \boldsymbol{w}_k'\boldsymbol{x}_j + b_k = \eta_{kj},\end{aligned} \tag{3}$$

where $k = 1, 2$ are represented positive and negative class respectively, and $\mathcal{I}_{\bar{k}}$ is the indices set of the rest class instances. The solution of problem (3) can be obtained via solving its dual problem

$$\begin{aligned}\min_{\boldsymbol{\alpha}_k}\ & \frac{1}{2}\sum_{i\in\mathcal{I}_k}\sum_{j\in\mathcal{I}_k}\alpha_{ik}^2\boldsymbol{x}_i'\boldsymbol{x}_j - c_1\sum_{i\in\mathcal{I}_k}\sum_{j\in\mathcal{I}_{\bar{k}}}\alpha_{ik}\boldsymbol{x}_i'\boldsymbol{x}_j,\\ \text{s.t. }\ & \sum_{i\in\mathcal{I}_k}\alpha_{ik} = c_1,\\ & 0\leq\alpha_{ik}\leq c_2.\end{aligned} \tag{4}$$

Then, $\boldsymbol{w}_k$ and b_k in (3) can be calculated according to the KKT conditions,

$$\boldsymbol{w}_k = \sum_{i\in\mathcal{I}_k} \alpha_{ik}\boldsymbol{x}_i - c_2 \sum_{j\in\mathcal{I}_{\bar{k}}} \boldsymbol{x}_j, \tag{5}$$

$$b_k = -\frac{1}{|\mathcal{I}_{SV}|} \sum_{i\in\mathcal{I}_{SV}} \boldsymbol{w}_k'\boldsymbol{x}_i, \tag{6}$$

where $\mathcal{I}_{SV}$ is indices of the support vector set.

Note that the TPMSVM can capture more complex heteroscedastic error structures via parametric-margin hyperplanes compared with TWSVM [17]. However, similar to TWSVM, the solution $\boldsymbol{w}_k$ and b_k of problem (3) needs solve the QPP (4), which is intractable or even impossible for large-scale learning tasks.

C. kPC

The kPC [15] is the first plane-based clustering. It aims to seek K optimal cluster plane-centers

$$f_k(\boldsymbol{x}) := \boldsymbol{w}_k'\boldsymbol{x} + b_k = 1, \quad k = 1, \cdots, K. \tag{7}$$

Specifically, kPC randomly initializes the cluster assignment for instances, and then updates the K cluster plane-centers via optimizing the following problems with $k = 1, \cdots, K$,

$$\begin{aligned} \min_{\boldsymbol{w}_k, b_k} \ & \frac{1}{2}\sum_{i\in\mathcal{I}_k} (\boldsymbol{w}_k'\boldsymbol{x}_i + b_k)^2, \\ \text{s.t. } & \|\boldsymbol{w}_k\|^2 = 1. \end{aligned} \tag{8}$$

Once obtained the plane-center (7), the cluster of each instance is reassigned by

$$\text{Cluster}(\boldsymbol{x}) = \arg\min_k\{|\boldsymbol{w}_k'\boldsymbol{x} + b_k|, k = 1, \cdots, K\} \tag{9}$$

In brief, the cluster of instances are updated according to (9) and the new cluster plane-centers are obtained by solving (8), until some terminate conditions are satisfied.

D. TWSVC

The recently proposed TWSVC [28] is a powerful plane-based clustering, which is an unsupervised extension to TWSVM [17]. Specifically, it aims to seek K cluster plane-centers (7) via optimizing the following problems with $k = 1, \cdots, K$,

$$\begin{aligned} \min_{\boldsymbol{w}_k, b_k} \ & \frac{1}{2}\sum_{i\in\mathcal{I}_k} (\boldsymbol{w}_k'\boldsymbol{x}_i + b_k)^2 + c\sum_{j\in\mathcal{I}_{\bar{k}}} \xi_{jk}, \\ \text{s.t. } & |\boldsymbol{w}_k'\boldsymbol{x}_j + b_k| \geq 1 - \xi_{jk}, \quad \xi_{jk} \geq 0, \end{aligned} \tag{10}$$

where $c > 0$ is a penalty parameter, and ξ_k is a slack vector.

However, problem (10) is difficult to be optimized due to the non-smooth absolute terms. Thus, by using the sub-gradient of $|\boldsymbol{w}_k'\boldsymbol{x}_j + b_k|$ and the Taylor series expansion, problem (10) can be successively optimized by concave–convex procedure (CCCP) [38] with initials $\boldsymbol{w}_k^0$ and b_k^0 as

$$\begin{aligned} \min_{\boldsymbol{w}_k^t, b_k^t} \ & \frac{1}{2}\sum_{i\in\mathcal{I}_k} ((\boldsymbol{w}_k^t)'\boldsymbol{x}_i + b_k^t)^2 + c\sum_{j\in\mathcal{I}_{\bar{k}}} \xi_{jk}, \\ \text{s.t. } & \boldsymbol{D}^t((\boldsymbol{w}_k^t)'\boldsymbol{x}_j + b_k^t) \geq 1 - \xi_{jk}, \quad \xi_{jk} \geq 0, \end{aligned} \tag{11}$$

where $\boldsymbol{D}^t = \text{diag}(\text{sign}((\boldsymbol{w}_k^{t-1})'\boldsymbol{x}_j + b_k^{t-1}))$.

Note that the solution of subproblem (11) is obtained via solving its dual QPP, which is computational costly for large-scale learning tasks. Moreover, kPC, PPC and TWSVC all adopt the L_2-norm least-square loss to measure the similarity of within cluster, which is sensitive to outliers.

III. THE PROPOSED METHOD

In the following, we introduce our MPMSVC, a novel unsupervised multiple parametric-margin support vector clustering for noisy learning tasks. Firstly, formulate the optimization problem of MPMSVC and give its geometrical interpretation in subsection III-A. Afterwards, elaborate the model optimization, and design an iteration algorithm for the model solution in subsection III-B. The convergence of algorithm is analyzed theoretically in subsection III-C.

A. MODEL FORMULATION

It has been seen that the quadratic L_2-norm measurement in kPC, PPC and TWSVC is sensitive to outliers. That is, their performance will be degenerated in noisy situation. On the other hand, the recently proposed twin parametric-margin support vector machine (TPMSVM) [23] is an excellent supervised nonparallel classifier. Compared with GEPSVM [16] and TWSVM [17], TPMSVM can capture more complex heteroscedastic error structures via parametric-margin hyperplanes. Thus, motivated by the work on TPMSVM and nonparallel plane-based clustering paradigm, we propose a novel unsupervised multiple parametric-margin support vector clustering (MPMSVC) for noisy clustering tasks.

In what follows, we derive the optimization problems of MPMSVC. With the initial cluster assignment of $\boldsymbol{X}$, our MPMSVC iteratively updates the following K cluster plane-centers and the corresponding clusters of instances,

$$f_k(\boldsymbol{x}) := \boldsymbol{w}_k'\boldsymbol{x} + b_k = 1, \quad k = 1, \cdots, K. \tag{12}$$

Such that, for the k-th cluster, gather the within-cluster instances $\boldsymbol{x}_{i\in\mathcal{I}_k}$ as close as possible to its corresponding cluster plane-center $f_k(\boldsymbol{x}) = 1$, meanwhile keeping the between-cluster instances $\boldsymbol{x}_{j\in\mathcal{I}_{\bar{k}}}$ far away from the k-th cluster plane-center.

To derive the model of MPMSVC, firstly, we enhance MPMSVC in least squares sense via replacing the equality constraints with inequality in TPMSVM, which enjoys an effective learning procedure. Besides, to reduce the impact of outliers, L_1-norm metric [31], [39]–[41] is further considered for MPMSVC, which can also improve the flexibility of model. For this purpose, our MPMSVC considers the following linear L_1-norm loss function for each cluster plane-center

$$\mathcal{R}_k = c_1 \sum_{i\in\mathcal{I}_k} |f(\boldsymbol{x}_i) - 1| + c_2 \sum_{j\in\mathcal{I}_{\bar{k}}} f(\boldsymbol{x}_j), \tag{13}$$

where $k = 1, \cdots, K$, and $c_1, c_2 > 0$ are penalty parameters for balancing the loss of within-cluster instances $\boldsymbol{x}_{i\in\mathcal{I}_k}$ and between-cluster instances $\boldsymbol{x}_{j\in\mathcal{I}_{\bar{k}}}$. Furthermore, by introducing the regularization term $\frac{1}{2}(\|\boldsymbol{w}_k\|^2+b_k^2)$, yield the primal problem of MPMSVC

$$\begin{aligned}\min_{\Xi_k}\ & \frac{1}{2}(\|\boldsymbol{w}_k\|^2+b_k^2)+c_1\sum_{i\in\mathcal{I}_k}|\xi_{ki}|+c_2\sum_{j\in\mathcal{I}_{\bar{k}}}\eta_{kj},\\ \text{s.t. } & \boldsymbol{w}_k'\boldsymbol{x}_i+b_k-1=\xi_{ki},\\ & \boldsymbol{w}_k'\boldsymbol{x}_j+b_k=\eta_{kj},\end{aligned} \tag{14}$$

where $\Xi_k = (\boldsymbol{w}_k, b_k, \boldsymbol{\xi}_k, \boldsymbol{\eta}_k)$. To deliver the mechanism of MPMSVC, we carry out the geometrical explanation for problem (14):

- The first term of objective is used to control the model complexity of the parametric-margin plane-center $f_k(\boldsymbol{x})$. Optimizing it aims to avoid model overfitting.
- For the first constraint with the second term of objective, the linear L_1-norm loss is adopted to implement the empirical risk of within cluster instances $\boldsymbol{x}_{i\in\mathcal{I}_k}$. Minimizing it encourages each within-cluster instance $\boldsymbol{x}_i$ to gather within the parametric-margin plane-center $f_k(\boldsymbol{x}) = 1$ as much as possible. Otherwise, a slack variable ξ_{ki} is introduced to measure this error.
- For the second constraint with the third term of objective, the linear loss is utilized to measure the empirical risk of between-cluster instances $\boldsymbol{x}_{j\in\mathcal{I}_{\bar{k}}}$. Optimizing this term pushes the between cluster instances $\boldsymbol{x}_j$ far away from hyperplanes $f_k(\boldsymbol{x}) = 1$. Otherwise, a slack variable η_{kj} is utilized to measure this error.

Once the solution of problem (14) for each cluster in MPMSVC is obtained, the corresponding k-th cluster plane-center $f_k(\boldsymbol{x}) = \boldsymbol{w}_k'\boldsymbol{x} + b_k$ is updated, $k = 1, \cdots, K$. Then, reassign the cluster of instances depending on which cluster plane-center it is nearest to, i.e.,

$$\mathbf{cluster}(\boldsymbol{x}) = \arg\min_{k=\{1,\cdots,K\}} |f_k(\boldsymbol{x})-1|. \tag{15}$$

The whole routine is that, MPMSVC updates the cluster plane-center (12) by solving problem (14), and the assignments of instances by operation (15) alternately until some terminate conditions are satisfied. In brief, we give the learning procedure of MPMSVC in Algorithm 1.

Algorithm 1 The Main Procedure of MPMSVC

Input: Data matrix $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^m$, and parameters c_1, c_2.
1: Initialize the labels $\boldsymbol{Y} = \{y_i\}_{i=1}^m$ for $\boldsymbol{X}$..
2: **while** There is no more difference of $\boldsymbol{Y}$ between iterations. **do**
3: Update the k-th cluster plane-center $f(\boldsymbol{x})$ by solving problem (14), where $k = 1, \cdots, K$.
4: Relabel instances to the nearest cluster plane-center via operation (15).
5: **end while**
Output: The labels $\boldsymbol{Y}$ for $\boldsymbol{X}$.

B. MODEL OPTIMIZATION

It is known that the most computational cost in Algorithm 1 is solving problem (14) for updating each cluster plane-center. However, the objective function in problem (14) involves the non-smooth L_1-norm term. As a result, it is a challenge to obtain its optimal solution directly by traditional gradient-based optimization techniques. In what follows, we will concern on how to optimize problem (14).

For simplicity, denote $\boldsymbol{A}_k = \{\boldsymbol{x}_i\}_{i\in\mathcal{I}_k} \in \mathbb{R}^{m_k\times n}$ as instances of the k-th cluster with m_k size in j-th iteration, and $\boldsymbol{B}_k = \{\boldsymbol{x}_j\}_{j\in\mathcal{I}_{\bar{k}}} \in \mathbb{R}^{m_{\bar{k}}\times n}$ as instances of the rest clusters with $m_{\bar{k}}$ size, $k = 1, \cdots, K$. Then, we can simply express problem (14) in the following matrix form as

$$\begin{aligned}\min_{\Xi_k}\ & \frac{1}{2}(\|\boldsymbol{w}_k\|^2+b_k^2)+c_1\|\boldsymbol{\xi}_k\|_1+c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{\eta}_k,\\ \text{s.t. } & (\boldsymbol{A}_k\boldsymbol{w}_k+b_k\boldsymbol{e}_k)-\boldsymbol{e}_k=\boldsymbol{\xi}_k,\\ & (\boldsymbol{B}_k\boldsymbol{w}_k+b_k\boldsymbol{e}_k)=\boldsymbol{\eta}_k,\end{aligned} \tag{16}$$

where $\|\cdot\|_1$ is the L_1-norm, $\boldsymbol{e}_k \in \mathbb{R}^{m_k}$ and $\boldsymbol{e}_{\bar{k}} \in \mathbb{R}^{m_{\bar{k}}}$ are vectors with all one elements.

Furthermore, denote $\boldsymbol{u}_k^t = [\boldsymbol{w}_k; b_k]$, $\boldsymbol{H}_k = [\boldsymbol{A}_k, \boldsymbol{e}_k]$, $\boldsymbol{G}_k = [\boldsymbol{B}_k, \boldsymbol{e}_{\bar{k}}]$. Then, substitute the constraint $\boldsymbol{G}_k\boldsymbol{u}_k = \boldsymbol{\eta}_k$ into the objective of problem (16) and obtain

$$\begin{aligned}\min_{\Xi_k}\ & \frac{1}{2}\|\boldsymbol{u}_k\|^2+c_1\|\boldsymbol{\xi}_k\|_1+c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{G}_k\boldsymbol{u}_k,\\ \text{s.t. } & \boldsymbol{H}_k\boldsymbol{u}_k-\boldsymbol{e}_k=\boldsymbol{\xi}_k.\end{aligned} \tag{17}$$

According to mathematical knowledge, we can reformulate the nonsmooth term $\|\boldsymbol{\xi}_k\|_1$ as

$$\begin{aligned}\|\boldsymbol{\xi}_k\|_1 &= \frac{1}{2}\boldsymbol{\xi}_k'\begin{pmatrix}\frac{1}{|\xi_{k,1}|} & & \\ & \vdots & \\ & & \frac{1}{|\xi_{k,m_k}|}\end{pmatrix}\boldsymbol{\xi}_k\\ &= \frac{1}{2}\boldsymbol{\xi}_k'\operatorname{diag}\left(\frac{1}{|\boldsymbol{\xi}_k|}\right)\boldsymbol{\xi}_k,\end{aligned} \tag{18}$$

where $\xi_{k,i}$ is the i-th element of vector $\boldsymbol{\xi}_k$, the operation $\operatorname{diag}(\boldsymbol{x})$ denotes the conversion of a vector $\boldsymbol{x}$ into a diagonal matrix. Afterwards, problem (17) can be rewritten as

$$\begin{aligned}\min_{\Xi_k}\ & \frac{1}{2}\|\boldsymbol{u}_k\|^2+\frac{c_1}{2}\boldsymbol{\xi}_k'\operatorname{diag}\left(\frac{1}{|\boldsymbol{\xi}_k|}\right)\boldsymbol{\xi}_k+c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{G}_k\boldsymbol{u}_k,\\ \text{s.t. } & \boldsymbol{H}_k\boldsymbol{u}_k-\boldsymbol{e}_k=\boldsymbol{\xi}_k.\end{aligned} \tag{19}$$

However, optimizing problem (19) is difficult for traditional optimization algorithm due to the absolute value operation. In what follows, we will present an efficient iterative algorithm. That is, it updates the solution $\boldsymbol{u}_k$ iteratively via solving a convex subproblem (20) until converges. Suppose $\boldsymbol{u}_k^t$ is the optimal solution achieved in the t-th iteration. Then, the next iteration solution $\boldsymbol{u}_k^{t+1}$ can be solved by

$$\begin{aligned}\min_{\Xi_k}\ & \frac{1}{2}\|\boldsymbol{u}_k\|^2+\frac{c_1}{2}\boldsymbol{\xi}_k'\boldsymbol{D}_\xi^t\boldsymbol{\xi}_k+c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{G}_k\boldsymbol{u}_k,\\ \text{s.t. } & \boldsymbol{H}_k\boldsymbol{u}_k-\boldsymbol{e}_k=\boldsymbol{\xi}_k,\end{aligned} \tag{20}$$

where $\boldsymbol{D}_\xi^t$ is computed according to the t-th solution $\boldsymbol{u}_k^t$ as

$$\boldsymbol{D}_\xi^t = \operatorname{diag}\left(\frac{1}{|\boldsymbol{\xi}_k^t|}\right) = \operatorname{diag}\left(\frac{1}{|\boldsymbol{H}_k\boldsymbol{u}_k^t-\boldsymbol{e}_k|}\right) \tag{21}$$

In what follows, we will derive the close-form solution of problem (20) in each iteration by Proposition 1.

Proposition 1: Suppose that the solution $\boldsymbol{u}_k^t$ of (20) *at the t-th iteration have been obtained. $\boldsymbol{D}^t$ is updated with respect to $\boldsymbol{u}_k^t$ according to* (21). *Then, the solution $\boldsymbol{u}_k^{t+1}$ of the* $(t+1)$-*th iteration optimization problem* (20) *can be obtained in the following close-form*

$$\boldsymbol{u}_k^{t+1} = \frac{c_1 \boldsymbol{M}_k^t \boldsymbol{e}_k - c_2 \boldsymbol{G}_k' \boldsymbol{e}_{\bar{k}}}{\boldsymbol{I} + c_1 \boldsymbol{M}_k^t \boldsymbol{H}_k}, \tag{22}$$

where $\boldsymbol{M}_k^t = \boldsymbol{H}_k' \boldsymbol{D}_\xi^t$.

Proof: Substitute the equality constraint and isolate $\boldsymbol{\xi}_k$ at the objective function of problem (20). Then, obtain the following unconstrained optimization problem

$$\min_{\boldsymbol{u}_k} \frac{1}{2}\|\boldsymbol{u}_k\|^2 + \frac{c_1}{2}(\boldsymbol{H}_k\boldsymbol{u}_k - \boldsymbol{e}_k)'\boldsymbol{D}_\xi^t(\boldsymbol{H}_k\boldsymbol{u}_k - \boldsymbol{e}_k) + c_2 \boldsymbol{e}_{\bar{k}}' \boldsymbol{G}_k \boldsymbol{u}_k. \tag{23}$$

Furthermore, denote $\boldsymbol{J}(\boldsymbol{u}_k)$ as the objective function of problem (23). Setting the gradient of $\boldsymbol{J}(\boldsymbol{u}_k)$ with respect to $\boldsymbol{u}_k$ to zero, give

$$\nabla_{\boldsymbol{u}_k}\boldsymbol{J} = \boldsymbol{u}_k + c_1 \boldsymbol{H}_k' \boldsymbol{D}_\xi^t(\boldsymbol{H}_k\boldsymbol{u}_k - \boldsymbol{e}_k) + c_2 \boldsymbol{G}_k' \boldsymbol{e}_{\bar{k}} = 0. \tag{24}$$

Denote $\boldsymbol{M}_k^t = \boldsymbol{H}_k'\boldsymbol{D}_\xi^t$ and arrange (24) as

$$(\boldsymbol{I} + c_1 \boldsymbol{M}_k^t \boldsymbol{H}_k)\boldsymbol{u}_k - c_1 \boldsymbol{M}_k^t \boldsymbol{e}_k + c_2 \boldsymbol{G}_k' \boldsymbol{e}_{\bar{k}} = 0, \tag{25}$$

$$\Rightarrow (\boldsymbol{I} + c_1 \boldsymbol{M}_k^t \boldsymbol{H}_k)\boldsymbol{u}_k = (c_1 \boldsymbol{M}_k^t \boldsymbol{e}_k - c_2 \boldsymbol{G}_k' \boldsymbol{e}_{\bar{k}}), \tag{26}$$

Then, archive the solution $\boldsymbol{u}_k^{t+1}$ of problem (20) as

$$\boldsymbol{u}_k^{t+1} = \frac{c_1 \boldsymbol{M}_k^t \boldsymbol{e}_k - c_2 \boldsymbol{G}_k' \boldsymbol{e}_{\bar{k}}}{\boldsymbol{I} + c_1 \boldsymbol{M}_k^t \boldsymbol{H}_k}. \tag{27}$$

Note that, according to (27), the solution $\boldsymbol{u}_k^{t+1}$ can be easily obtained by solving an linear equation system. ■

In short, the solution of problem (17) can be solved by the following Algorithm 2.

Algorithm 2 Procedure for Solving (17) in Linear MPMSVC

Input: Data matrices $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$, and parameters c_1, c_2.

1: Initialize $\boldsymbol{u}_k^0$ and set iteration $t = 0$.
2: Augment data matrices $\boldsymbol{H}_k = [\boldsymbol{A}_k, \boldsymbol{e}_k]$ for within-cluster instances, and $\boldsymbol{G}_k = [\boldsymbol{B}_k, \boldsymbol{e}_{\bar{k}}]$ for between-cluster instances.
3: **while** not converge **do**
4: Set $t = t + 1$.
5: Compute $\boldsymbol{\xi}_k^t = \boldsymbol{H}_k \boldsymbol{u}_k^t - \boldsymbol{e}_k$ according to $\boldsymbol{u}_k^t$, and the diagonal matrix $\boldsymbol{D}_\xi^t$ with its i-th diagonal elements as $d_{\xi_i}^t = \frac{1}{|\xi_i^t|}$.
6: Compute $\boldsymbol{M}_k^t = \boldsymbol{H}_k' \boldsymbol{D}_\xi^t$.
7: Update the solution $\boldsymbol{u}_k^{t+1}$ of problem (20) by solving an linear equation system as

$$\boldsymbol{u}_k^{t+1} = \frac{c_1 \boldsymbol{M}_k^t \boldsymbol{e}_k - c_2 \boldsymbol{G}_k' \boldsymbol{e}_{\bar{k}}}{\boldsymbol{I} + c_1 \boldsymbol{M}_k^t \boldsymbol{H}_k}. \tag{28}$$

8: **end while**

Output: Solution $\boldsymbol{u}_k^*$ for problem (17).

C. CONVERGENCE ANALYSIS

The iterative algorithm 2 firstly updates the diagonal matrix $\boldsymbol{D}_\xi^t$ based on $\boldsymbol{u}^t$, and then updates $\boldsymbol{u}^{t+1}$ according to (27) via solving an linear equation problem. That is, $\boldsymbol{u}_k$ and $\boldsymbol{D}_\xi^t$ is updated alternately until converge. In what follows, we will concern on the convergence of Algorithm 2.

Lemma 1: For any nonzero vector $\boldsymbol{\xi}_k^t, \boldsymbol{\xi}_k^{t+1} \in \mathbb{R}^n$, the following inequality is established:

$$\|\boldsymbol{\xi}_k^{t+1}\|_1 - \frac{\|\boldsymbol{\xi}_k^{t+1}\|_1^2}{2\|\boldsymbol{\xi}_k^t\|_1} \leq \|\boldsymbol{\xi}_k^t\|_1 - \frac{\|\boldsymbol{\xi}_k^t\|_1^2}{2\|\boldsymbol{\xi}_k^t\|_1} \tag{29}$$

Proof: For any nonnegative scalar a and b, have inequality $(\sqrt{a} - \sqrt{b})^2 \geq 0$. Then, derive

$$\begin{aligned}
&(\sqrt{a} - \sqrt{b})^2 \geq 0 \\
&\Rightarrow a - 2\sqrt{ab} + b \geq 0 \\
&\Rightarrow \frac{\sqrt{b}}{2} \geq \sqrt{a} - \frac{a}{2\sqrt{b}} \quad (\textit{divide } 2\sqrt{ab}) \\
&\Rightarrow \sqrt{b} - \frac{b}{2\sqrt{b}} \geq \sqrt{a} - \frac{a}{2\sqrt{b}} \quad (\textit{split } \sqrt{b} = \frac{b}{\sqrt{b}})
\end{aligned} \tag{30}$$

Substitute $a = \|\boldsymbol{\xi}_k^{t+1}\|_1^2$ and $b = \|\boldsymbol{\xi}_k^t\|_1^2$ into (30)*, then achieve* (29). ■

Theorem 1: Algorithm 2 monotonically non-increases the objective function of problem (17) *in each iteration.*

Proof: The main routine in Algorithm 2 is to iteratively update the solution of problem (17) via solving problem (20). Given the t-th solution $\boldsymbol{u}_k$, problem (20) can be rewritten as

$$\boldsymbol{u}_k^{t+1} = \arg\min_{\boldsymbol{u}_k} \boldsymbol{J}(\boldsymbol{u}_k) \tag{31}$$

where the objective function $\boldsymbol{J}(\boldsymbol{u}_k)$ is defined as

$$\boldsymbol{J}(\boldsymbol{u}_k) = \frac{1}{2}\|\boldsymbol{u}_k\|^2 + \frac{c_1}{2}(\boldsymbol{H}_k\boldsymbol{u}_k - \boldsymbol{e}_k)'\boldsymbol{D}_\xi^t \times (\boldsymbol{H}_k\boldsymbol{u}_k - \boldsymbol{e}_k) + c_2 \boldsymbol{e}_{\bar{k}}' \boldsymbol{G}_k \boldsymbol{u}_k. \tag{32}$$

According to step 7 in Algorithm 2, $\boldsymbol{u}_k^{t+1}$ is the optimal solution of problem (31) in the $t+1$ iteration. Thus, we have

$$\begin{aligned}
&\frac{1}{2}\|\boldsymbol{u}_k^{t+1}\|^2 + \frac{c_1}{2}(\boldsymbol{H}_k\boldsymbol{u}_k^{t+1} - \boldsymbol{e}_k)'\boldsymbol{D}_\xi^t \\
&\quad \times (\boldsymbol{H}_k\boldsymbol{u}_k^{t+1} - \boldsymbol{e}_k) + c_2 \boldsymbol{e}_{\bar{k}}' \boldsymbol{G}_k \boldsymbol{u}_k^{t+1} \\
&\leq \frac{1}{2}\|\boldsymbol{u}_k^t\|^2 + \frac{c_1}{2}(\boldsymbol{H}_k\boldsymbol{u}_k^t - \boldsymbol{e}_k)'\boldsymbol{D}_\xi^t \\
&\quad \times (\boldsymbol{H}_k\boldsymbol{u}_k^t - \boldsymbol{e}_k) + c_2 \boldsymbol{e}_{\bar{k}}' \boldsymbol{G}_k \boldsymbol{u}_k^t
\end{aligned} \tag{33}$$

Denote $\boldsymbol{\xi}_k^t = \boldsymbol{H}_k\boldsymbol{u}_k^t - \boldsymbol{e}_k$, then update $\boldsymbol{D}_\xi^t = \text{diag}\left(\frac{1}{|\xi_k^t|}\right)$ according to step 5 of Algorithm 2. The above equation (33) can be simplified as

$$\frac{1}{2}\|\boldsymbol{u}_k^{t+1}\|^2 + c_1 \sum_{i \in \mathcal{I}_k} \frac{\left(\xi_{k,i}^{t+1}\right)^2}{2|\xi_{k,i}^t|} + c_2 \boldsymbol{e}_{\bar{k}}' \boldsymbol{G}_k \boldsymbol{u}_k^{t+1}$$

$$\leq \frac{1}{2}\|\boldsymbol{u}_k^t\|^2 + c_1 \sum_{i \in \mathcal{I}_k} \frac{\left(\xi_{k,i}^t\right)^2}{2|\xi_{k,i}^t|} + c_2 \boldsymbol{e}_{\bar{k}}' \boldsymbol{G}_k \boldsymbol{u}_k^t \tag{34}$$

On the other hand, from Lemma 1, we obtain

$$c_1\left(\|\boldsymbol{\xi}_k^{t+1}\|_1 - \frac{\|\boldsymbol{\xi}_k^{t+1}\|_1^2}{2\|\boldsymbol{\xi}_k^t\|_1}\right) \le c_1\left(\|\boldsymbol{\xi}_k^t\|_1 - \frac{\|\boldsymbol{\xi}_k^t\|_1^2}{2\|\boldsymbol{\xi}_k^t\|_1}\right). \quad (35)$$

That is,

$$c_1\left(\|\boldsymbol{\xi}_k^{t+1}\|_1 - \sum_{i\in\mathcal{I}_k}\frac{\left(\xi_{k,i}^{t+1}\right)^2}{2|\xi_{k,i}^t|}\right) \le c_1\left(\|\boldsymbol{\xi}_k^t\|_1 - \sum_{i\in\mathcal{I}_k}\frac{\left(\xi_{k,i}^{t}\right)^2}{2|\xi_{k,i}^t|}\right). \quad (36)$$

Combining (31), (34) and (36), we get

$$\begin{aligned} \boldsymbol{J}(\boldsymbol{u}_k^{t+1}) &= \frac{1}{2}\|\boldsymbol{u}_k^{t+1}\|^2 + \frac{c_1}{2}(\boldsymbol{H}_k\boldsymbol{u}_k^{t+1} - \boldsymbol{e}_k)'\boldsymbol{D}_\xi^{t+1} \\ &\quad \times (\boldsymbol{H}_k\boldsymbol{u}_k^{t+1} - \boldsymbol{e}_k) + c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{G}_k\boldsymbol{u}_k^{t+1} \\ &= \frac{1}{2}\|\boldsymbol{u}_k^{t+1}\|^2 + c_1\|\boldsymbol{\xi}_k^{t+1}\|_1 + c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{G}_k\boldsymbol{u}_k^{t+1} \\ &\le \frac{1}{2}\|\boldsymbol{u}_k^t\|^2 + c_1\|\boldsymbol{\xi}_k^t\|_1 + c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{G}_k\boldsymbol{u}_k^t \\ &= \frac{1}{2}\|\boldsymbol{u}_k^t\|^2 + \frac{c_1}{2}(\boldsymbol{H}_k\boldsymbol{u}_k^t - \boldsymbol{e}_k)'\boldsymbol{D}_\xi^t \\ &\quad \times (\boldsymbol{H}_k\boldsymbol{u}_k^t - \boldsymbol{e}_k) + c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{G}_k\boldsymbol{u}_k^t \\ &= \boldsymbol{J}(\boldsymbol{u}_k^t). \end{aligned} \quad (37)$$

Thus, the objective function $\boldsymbol{J}(\boldsymbol{u}_k)$ of problem (17) non-decreases via each iteration in Algorithm 2, which establishes the proof. ■

Note that, the objective function $\boldsymbol{J}(\boldsymbol{u}_k)$ of problem (17) has a lower bound zero. Hence, Theorem 1 indicates that $\boldsymbol{u}_k$ will converge to a local optimal solution of problem (17) by the proposed Algorithm 2.

IV. NONLINEAR EXTENSION

In this section, we extend the above linear MPMSVC to nonlinear case via kernel technique [28], [42]. Specifically, the nonlinear MPMSVC aims to seek K kernel generated cluster surface-centers

$$f_k^\phi(\boldsymbol{x}) = \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{X})\boldsymbol{w}_k^\phi + b_k = 1, \quad k = 1, \cdots, K. \quad (38)$$

where $\boldsymbol{K}(\cdot,\cdot)$ is an appropriate kernel function, $\boldsymbol{w}_k^\phi \in \mathbb{R}^m$ and $b_k \in \mathbb{R}$. Then, in each cluster updating routine, our nonlinear MPMSVC can be expressed in the following kernel formulation as

$$\begin{aligned} \min_{\Xi_k}\ & \frac{1}{2}(\|\boldsymbol{w}_k^\phi\|^2 + b_k^2) + \frac{c_1}{2}\|\boldsymbol{\xi}_k\|_1 + c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{\eta}_k, \\ \text{s.t. } & (\boldsymbol{K}(\boldsymbol{A}_k, \boldsymbol{X})\boldsymbol{w}_k^\phi + b_k\boldsymbol{e}_k) - \boldsymbol{e}_k = \boldsymbol{\xi}_k, \\ & (\boldsymbol{K}(\boldsymbol{B}_k, \boldsymbol{X})\boldsymbol{w}_k^\phi + b_k\boldsymbol{e}_{\bar{k}}) = \boldsymbol{\eta}_k, \end{aligned} \quad (39)$$

where $c_1, c_2 > 0$ are penalty parameters, $\boldsymbol{\xi}_k$ is a slack vector. Denote $\boldsymbol{v}_k = [\boldsymbol{w}_k^\phi; b_k]$, $\boldsymbol{H}_k^\phi = [\boldsymbol{K}(\boldsymbol{A}_k, \boldsymbol{X}), \boldsymbol{e}_k]$ and $\boldsymbol{G}_k^\phi = [\boldsymbol{K}(\boldsymbol{B}_k, \boldsymbol{X}), \boldsymbol{e}_{\bar{k}}]$, then rewrite problem (39) as

$$\begin{aligned} \min_{\Xi_k} =\ & \frac{1}{2}\|\boldsymbol{v}_k\|^2 + \frac{c_1}{2}\|\boldsymbol{\xi}_k\|_1 + c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{\eta}_k, \\ \text{s.t. } & \boldsymbol{H}_k^\phi\boldsymbol{v}_k - \boldsymbol{e}_k = \boldsymbol{\xi}_k, \\ & \boldsymbol{G}_k^\phi\boldsymbol{v}_k = \boldsymbol{\eta}_k, \end{aligned} \quad (40)$$

Similar to linear case, we optimize problem (40) via solving series of convex unconstrained quadratic problem (41) iteratively until converges. That is, the iteration solution $\boldsymbol{u}_k^{t+1}$ is updated by

$$\min_{\boldsymbol{v}_k} \frac{1}{2}\|\boldsymbol{v}_k\|^2 + \frac{c_1}{2}(\boldsymbol{H}_k^\phi\boldsymbol{v}_k - \boldsymbol{e}_k)'\boldsymbol{D}_\xi^t(\boldsymbol{H}_k^\phi\boldsymbol{v}_k - \boldsymbol{e}_k) + c_2\boldsymbol{e}_{\bar{k}}'\boldsymbol{G}_k^\phi\boldsymbol{v}_k, \quad (41)$$

where $\boldsymbol{D}_\xi^t$ is a diagonal matrix with its i-th diagonal elements as $d_{\xi_i}^t = \frac{1}{|\xi_i^t|}$, where $\xi_i^t = \boldsymbol{H}_{k,i}^\phi\boldsymbol{v}_k^t - \boldsymbol{e}_k$ is the i-th element of the slack vector $\boldsymbol{\xi}_k^t$ and $\boldsymbol{H}_{k,i}^\phi$ is the i-th row of $\boldsymbol{H}_k^\phi$. Then, we can derive the solution of problem (41) in each iteration by Proposition 2 without proof.

Proposition 2: Suppose that the solution $\boldsymbol{v}_k^t$ of (41) at the t-th iteration have been obtained. $\boldsymbol{D}^t$ is updated with respect to $\boldsymbol{v}_k^t$ with its i-th diagonal elements as $d_{\xi_i}^t = \frac{1}{|\xi_i^t|}$. Then, the solution $\boldsymbol{v}_k^{t+1}$ of the $(t+1)$-th iteration optimization problem (41) can be obtained in the following close-form

$$\boldsymbol{v}_k^{t+1} = \frac{c_1\boldsymbol{M}_k^t\boldsymbol{e}_k - c_2(\boldsymbol{G}_k^\phi)'\boldsymbol{e}_{\bar{k}}}{\boldsymbol{I} + c_1\boldsymbol{M}_k^t\boldsymbol{H}_k^\phi}, \quad (42)$$

where $\boldsymbol{M}_k^t = (\boldsymbol{H}_k^\phi)'\boldsymbol{D}_\xi^t$.

In short, the problem (40) can be solved efficiently by the following Algorithm 3.

Algorithm 3 Procedure for Solving (40) in Nonlinear MPMSVC

Input: Data matrices $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$, and parameters c_1, c_2.

1: Initialize $\boldsymbol{u}_k^0$ and set iteration $t = 0$.
2: Augment data matrices $\boldsymbol{H}_k^\phi = [\boldsymbol{K}(\boldsymbol{A}_k, \boldsymbol{X}), \boldsymbol{e}_k]$ for within-cluster instances, and $\boldsymbol{G}_k^\phi = [\boldsymbol{K}(\boldsymbol{B}_k, \boldsymbol{X}), \boldsymbol{e}_{\bar{k}}]$ for between-cluster instances.
3: **while** not converge **do**
4: Set $t = t + 1$.
5: Compute $\boldsymbol{\xi}_k^t = \boldsymbol{H}_k^\phi\boldsymbol{v}_k^t - \boldsymbol{e}_k$ according to $\boldsymbol{v}_k^t$, and the diagonal matrix $\boldsymbol{D}_\xi^t$ with its i-th diagonal elements as $d_{\xi_i}^t = \frac{1}{|\xi_i^t|}$.
6: Compute $\boldsymbol{M}_k^t = (\boldsymbol{H}_k^\phi)'\boldsymbol{D}_\xi^t$.
7: Update the solution $\boldsymbol{v}_k^{t+1}$ of problem (20) by solving an linear equation system as

$$\boldsymbol{v}_k^{t+1} = \frac{c_1\boldsymbol{M}_k^t\boldsymbol{e}_k - c_2(\boldsymbol{G}_k^\phi)'\boldsymbol{e}_{\bar{k}}}{\boldsymbol{I} + c_1\boldsymbol{M}_k^t\boldsymbol{H}_k^\phi}. \quad (43)$$

8: **end while**

Output: Solution $\boldsymbol{v}_k^*$ for problem (40).

V. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETTING

To evaluate the performance of MPMSVC, we carry out extensive experiments on synthetic and benchmark datasets.

TABLE 1. Learning results of each method on two synthetic datasets in terms of RI and Time.

Datasets	Performance	Clustering methods					
		kPC	PPC	TBSVC	RTWSVC	FRTWSVC	MPMSVC
Cross3D	RI (%)	56.44	68.12	80.05	83.39	81.41	**96.18**
	Time (s)	0.0134	0.0098	3.8452	27.6294	0.0643	0.0851
TwoSin	RI (%)	49.60	53.06	68.13	84.83	79.51	**89.57**
	Time (s)	0.0207	1.8928	9.7428	16.8219	3.3154	2.4925

TABLE 2. Statistics for benchmark datasets used in experiments.

Datasets	Instances	Features	Clusters
Compound	399	2	6
Aggregation	788	2	7
Pathbased	300	2	3
Dermatology	366	34	6
Iris	150	4	3
Tae	150	3	3
Vowel	528	10	11
Zoo	101	16	7
Glass	214	9	6
Ecoli	336	7	8
Haberman	306	3	2

In experiments, we focus on the comparison between our MPMSVC and five state-of-the-art clustering methods, including kPC [15], PPC [27], TBSVC [32], RTWSVC [31] and FRTWSVC [31]. All methods are implemented by MATLAB on a PC with an i7 Intel Core processor with 32-GB RAM. The details are described as follows:

- The Rand Index (RI) [4] is utilized to measure their performance, which is a measure of the similarity between the ground-truth clusters $\boldsymbol{Y}$ and the predict clusters $\bar{\boldsymbol{Y}}$. The metric RI is defined as

$$RI = \frac{m_1}{m_1 + m_2} \times 100\% = \frac{m_1}{C_m^2} \times 100\%, \quad (44)$$

where m_1 is the number pairs of agreements between $\boldsymbol{Y}$ and $\bar{\boldsymbol{Y}}$, m_2 is the number pairs of disagreements between them, m is the number of instances, and $C_m^2 = m_1 + m_2$. Since the denominator is the total number of pairs, RI represents the frequency of occurrence of agreements over the total pairs. The range of RI is belong to $[0\%, 100\%]$.
- The cluster number K for all methods is set to the real one. The nearest neighbor graph (NNG) [28] is used as the initialization for these plane-clustering methods. The RBF kernel $\boldsymbol{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\{-\gamma\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2\}$ is used for nonlinear case, where $\gamma > 0$ is the kernel parameter.
- Parameters in all methods are optimized via the grid searching routine according to maximizing the performance. The parameters in these methods and γ are selected from $\{2^i | i = -6, -5, \cdots, 6\}$, while neighborhood size in NNG is $\{1, 2, \cdots, 5\}$.
- Similar to [29], [31], the stopping tolerances of TBSVC, RTWSVC, FRTWSVC, and MPMSVC are set as the successive iteration difference less than 0.001, i.e., $\|\boldsymbol{u}^{t+1} - \boldsymbol{u}^t\| < 0.001$. Moreover, the initial cluster plane for these methods is set as the solution of kPC.

B. RESULTS ON SYNTHETIC DATASETS

In this subsection, we validate the performance of these clustering methods on two noisy synthetic datasets: "Cross3D" dataset and "TwoSin" dataset.

The "Cross3D" dataset is consist of three underlying clusters, where a line cross through a plane and an ellipsoid. Then, instances are generated uniformly by these cluster distributions with Gaussian noisy. Moreover, three outliers are added to make the learning task more challenge. The illustration of "Cross3D" dataset is shown in Fig.1(a). We perform these clustering methods on "Cross3D" dataset with linear kernel. The clustering results are shown in Fig.1. It can be seen that kPC and TBSVC break down the line into different clusters, while PPC cannot disclose the instances from plane and ellipsoid clusters. Moreover, RTWSVC and FRTWSVC can discover line and plane clusters, but lose in ellipsoid cluster. Among these methods, our MPMSVC can discover the underlying clusters, and handle such a cross-cluster case well.

To further investigate the nonlinear performance of our MPMSVC, we test these clustering methods on the "TwoSin" dataset with RBF kernel. In particular, the "TwoSin" dataset in Fig.2(a) is consist of two "Sin" shape clusters with Gaussian noisy, where two clusters cross with each other. The learning results of each method in Fig.2 reveal that kPC just divides dataset into two linear sections, while PPC cannot recognize the two clusters at all. TBSVM, RTWSVC and FRTWSVC can discover parts of sine distributions. Our MPMSVC is able to exposing the underlying

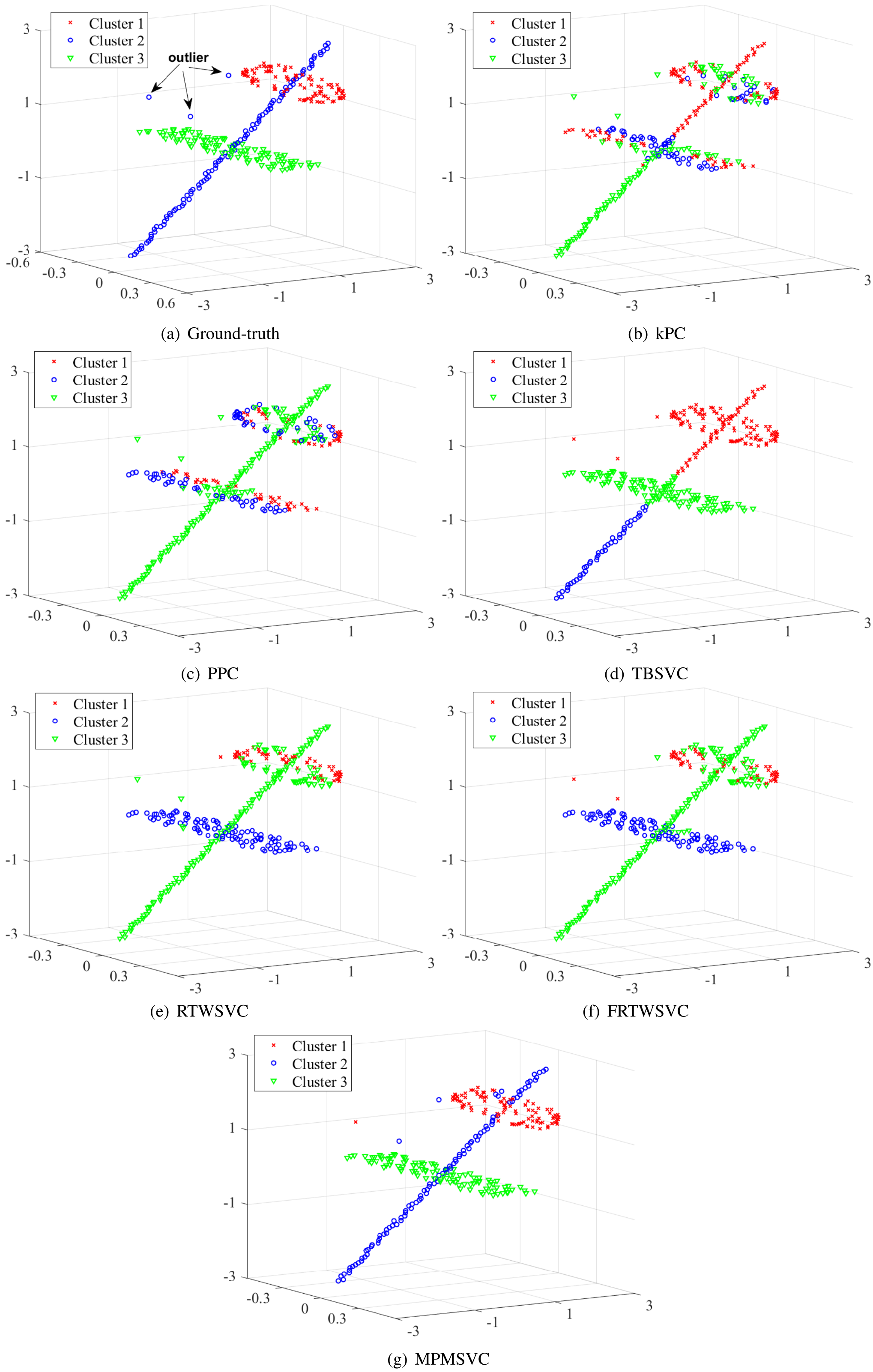


FIGURE 1. **The learning results of each clustering methods on "Cross3D" dataset with linear kernel.**

two "Sin" distributions and distinguishing instances from different clusters exactly except for some instances in overlap intersections.

For better comparisons, we also give the RI and learning time of these methods on the above two synthetic datasets in Table.1. The results show that the performance of

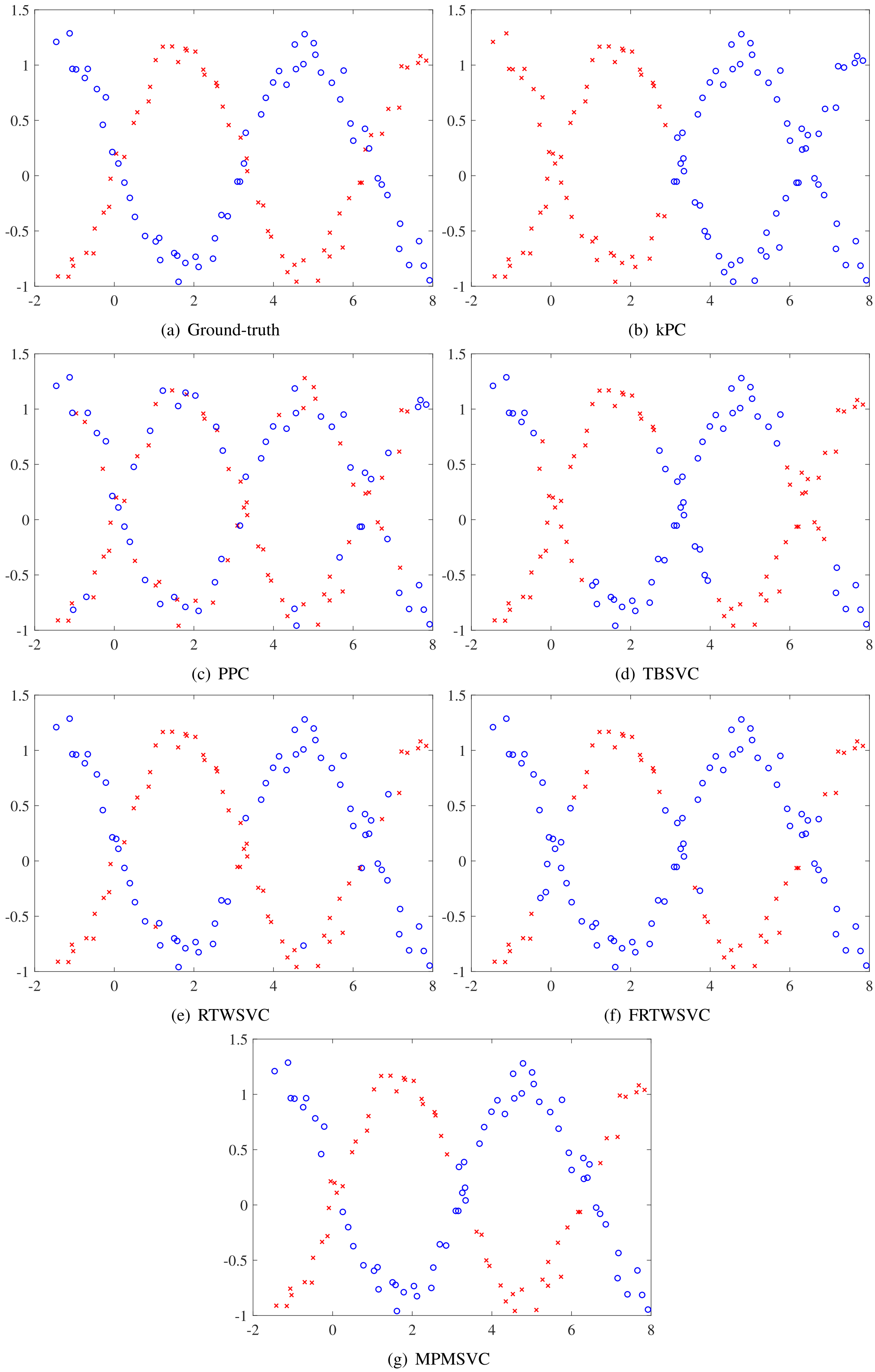


FIGURE 2. **The learning results of each clustering methods on "TwoSin" dataset with nonlinear kernel.**

L_2-norm loss based kPC, PPC and TBSVC deteriorates obviously compared with the L_1-norm loss based RTWSVC, FRTWSVC and MPMSVC. Moreover, our MPMSVC outperforms other methods in terms of RI metric. As for learning time, MPMSVC is a bit slower than FRTWSVC, but faster than TBSVC and RTWSVC. The reason is that,

TABLE 3. Performance of each clustering method with linear formulation on benchmark datasets.

Datasets	Noisy	Clustering methods					
		kPC	PPC	TBSVC	RTWSVC	FRTWSVC	MPMSVC
Compound	5%	70.79±1.53	73.08±2.18	78.81±1.75	80.61±1.96	82.39±1.83	**83.21±1.59**
	15%	67.14±2.35	70.34±3.69	76.90±1.81	79.02±2.37	80.56±2.26	**81.93±1.64**
Aggregation	5%	78.05±3.79	77.57±1.96	78.74±2.28	79.98±1.86	81.24±2.53	**81.57±1.16**
	15%	74.88±4.53	73.34±4.57	75.35±3.16	78.14±2.49	**80.13±3.39**	79.68±2.73
Pathbased	5%	64.66±2.62	60.78±1.92	72.53±1.53	**74.36±2.09**	73.28±2.25	74.17±1.51
	15%	62.75±2.72	59.91±2.45	71.38±2.45	72.12±2.69	71.95±3.50	**72.48±2.85**
Dermatology	5%	61.12±4.27	68.99±3.87	71.56±3.20	**72.76±2.10**	69.25±2.63	72.05±2.26
	15%	58.34±3.30	64.50±3.73	67.18±3.52	68.53±2.25	67.63±1.81	**69.32±2.13**
Iris	5%	67.02±2.18	59.74±2.57	89.82±1.38	92.23±1.93	**93.02±1.47**	92.86±1.72
	15%	63.91±3.75	55.10±4.78	84.22±3.28	86.50±3.13	86.73±2.39	**87.85±2.93**
Tae	5%	43.23±3.29	41.37±3.71	**56.19±3.16**	52.40±2.83	54.12±2.46	55.39±2.48
	15%	41.39±4.24	40.74±5.32	49.36±3.47	51.18±2.34	52.63±3.01	**52.98±2.71**
Vowel	5%	81.68±1.90	82.13±2.06	81.40±1.64	**84.35±1.93**	82.31±1.19	84.23±1.34
	15%	79.66±2.04	80.46±2.71	78.45±1.52	81.32±1.32	80.65±2.43	**82.19±1.61**
Zoo	5%	71.74±5.13	81.06±3.87	86.32±2.26	87.02±3.65	86.44±2.77	**86.70±2.82**
	15%	67.11±4.72	78.49±5.54	80.72±3.39	82.74±2.31	**85.28±3.51**	85.04±2.53
Glass	5%	53.63±4.95	59.83±3.63	**63.52±3.67**	58.38±2.67	59.45±3.54	61.18±2.54
	15%	51.11±3.68	58.12±4.92	58.34±5.69	56.72±2.64	56.53±2.86	**60.64±3.17**
Ecoli	5%	38.62±1.55	59.12±2.77	62.08±2.62	50.64±3.93	52.31±2.71	**63.97±2.83**
	15%	37.48±1.21	56.26±2.06	59.32±2.33	47.50±5.84	45.33±3.92	**61.45±3.05**
Haberman	5%	47.95±2.50	60.53±3.05	60.47±4.02	62.10±3.54	62.47±2.68	**62.74±3.29**
	15%	45.48±3.24	57.03±2.92	57.57±2.59	59.23±2.91	**59.51±3.42**	59.03±2.84
Average Rank		5.54545	4.86363	3.59090	2.77272	2.77272	1.45454
Win-Tie-Loss		22-0-0	22-0-0	20-0-2	18-0-4	18-0-4	

our MPMSVC just needs solve a system of linear equations compared with the QPPs for TBSVC and RTWSVC. The above results confirm the effectiveness of MPMSVC.

C. RESULTS ON BENCHMARK DATASETS

In this subsection, we compare the proposed MPMSVC with the other six state-of-art methods on eleven benchmark

TABLE 4. Performance of each clustering method with nonlinear formulation on benchmark datasets.

Datasets	Noisy	Clustering methods					
		kPC	PPC	TBSVC	RTWSVC	FRTWSVC	MPMSVC
Compound	5%	87.60±2.89	86.23±3.07	88.03±2.05	88.21±2.11	87.84±2.24	**89.27±1.88**
	15%	83.78±3.87	84.44±4.98	83.28±3.33	**86.55±2.48**	85.61±3.39	86.41±1.91
Aggregation	5%	82.85±4.28	82.26±3.38	84.65±3.96	86.40±2.92	85.95±2.83	**86.89±2.90**
	15%	79.17±5.31	78.23±5.08	81.09±3.67	82.59±3.85	83.57±4.07	**84.06±3.16**
Pathbased	5%	73.11±3.93	57.36±2.47	**79.06±3.16**	73.04±3.96	74.16±3.33	78.52±2.61
	15%	70.29±3.06	56.94±3.81	73.65±2.94	71.91±2.95	73.51±5.49	**75.39±3.14**
Dermatology	5%	69.92±5.68	67.31±5.18	68.68±5.06	**73.14±3.24**	72.16±2.79	72.87±2.97
	15%	67.03±3.36	66.24±4.06	66.75±4.22	70.49±3.19	67.76±3.59	**71.08±2.32**
Iris	5%	89.13±2.73	73.28±2.81	86.41±1.77	86.38±1.95	89.61±1.68	**90.07±2.42**
	15%	84.41±3.84	71.53±5.78	85.04±3.78	87.67±3.80	**88.15±4.31**	87.39±3.96
Tae	5%	42.96±3.48	41.77±5.63	46.16±4.39	45.96±3.15	**49.17±2.47**	48.63±3.28
	15%	40.89±5.89	40.53±6.00	42.33±4.42	44.39±3.93	46.04±4.56	**46.78±2.86**
Vowel	5%	82.19±3.29	82.62±3.23	84.13±2.34	**85.67±2.55**	81.75±2.82	84.67±1.82
	15%	78.73±2.67	78.85±3.16	80.45±3.18	82.15±2.38	80.57±4.17	**82.54±1.86**
Zoo	5%	86.29±7.03	77.94±5.37	87.65±3.43	89.13±3.98	87.90±2.94	**89.52±3.19**
	15%	81.90±4.79	76.48±6.05	82.11±4.49	86.09±3.51	86.15±4.31	**87.20±3.42**
Glass	5%	65.83±5.83	66.69±4.64	61.58±5.50	68.26±3.20	67.05±4.06	**69.91±3.37**
	15%	64.41±4.44	62.58±6.32	59.23±6.26	65.19±3.95	**66.43±4.46**	66.07±3.08
Ecoli	5%	81.19±3.08	70.59±4.55	81.72±4.13	82.08±5.31	79.40±3.57	**82.51±4.64**
	15%	78.30±2.80	67.64±3.98	77.81±3.84	79.73±7.34	77.93±5.74	**80.84±4.94**
Haberman	5%	58.26±2.87	59.83±4.14	61.02±4.78	**62.98±4.44**	61.70±3.04	62.51±4.27
	15%	57.94±4.22	56.39±3.20	58.29±3.73	59.70±3.08	**60.96±3.95**	60.42±3.82
Ave. Rank		4.6818	5.5909	3.9999	2.4091	2.8181	1.4999
Win-Tie-Loss		22-0-0	22-0-0	21-0-1	16-0-6	18-0-4	

datasets [33], [34], whose statistics are listed in Table 2. All datasets are normalized before learning such that the features are scaled in the interval $[-1, 1]$. In our experiments, we set up in the following way. Firstly, we randomly choose m ratio of instances, and then pollute their features with Gaussian noise to generate outliers. Here, we consider two noisy clustering situations: the ratio $m = 5\%$ as the light noisy case and $m = 15\%$ as the heavy noisy case. Finally, we transform them into clustering tasks. Each experimental setting is repeated 10 times. The learning results on eleven benchmark datasets with both 5% and 15% noisy levels are reported in Tables 3 for linear case and Table 4 for nonlinear case, respectively. The best result is highlighted in bold style.

The results reveal that, with the noisy level m increasing, the performance for all methods deteriorate generally. Moreover the performance of L_2-based kPC, PPC and TBSVC are dramatically worse than L_1-based RTWSVC, FRTWSVC and MPMSVC in most cases. Take the linear case for example, the L_1-based (L_2-based) methods obtain the best on 9/11 (2/11) datasets for the slight 5% noisy level, and 11/11 (none) for the heavy 15% noisy level. Similar results can be obtained for the nonlinear case. The reason behind is that the quadratic L_2 distance measurement is more sensitive to noisy than the L_1 one. On the other hand, our MPMSVC behaves better performance w.r.t. RI in comparison to existing methods on most datasets, and is comparable with the best one on the rests. This is justified by the fact that MPMSVC obtains the best on 13/22 datasets for linear case and 14/22 datasets for nonlinear case, respectively.

To provide more statistical evidence [43], [44], we employ the Friedman's test to check whether there are significant differences between MPMSVC and other methods according to the RI metric in Tables 3 and Table 4. It can be seen that our MPMSVC is ranked first in both linear and nonlinear cases, followed by FRTWSVC and RTWSVC successively. Now, we calculate the $\mathcal{X}_F^2$ value for Friedman's test as

$$\mathcal{X}_F^2 = \frac{12N}{k(k+1)}\left[\sum_{i=1}^{k} r_i^2 - \frac{k(k+1)^2}{4}\right] \tag{45}$$

where r_i is the average rank on N datasets for i-th method. For the linear case, we compute term $\sum_{i=1}^{k} r_i^2$ in Table 3 as

$$\begin{aligned}\sum_{i=1}^{k} r_i^2 &= 5.54545^2 + 4.86363^2 + 3.59090^2 \\ &\quad + 2.77272^2 + 2.77272^2 + 1.45454^2 \\ &\approx 84.7931\end{aligned} \tag{46}$$

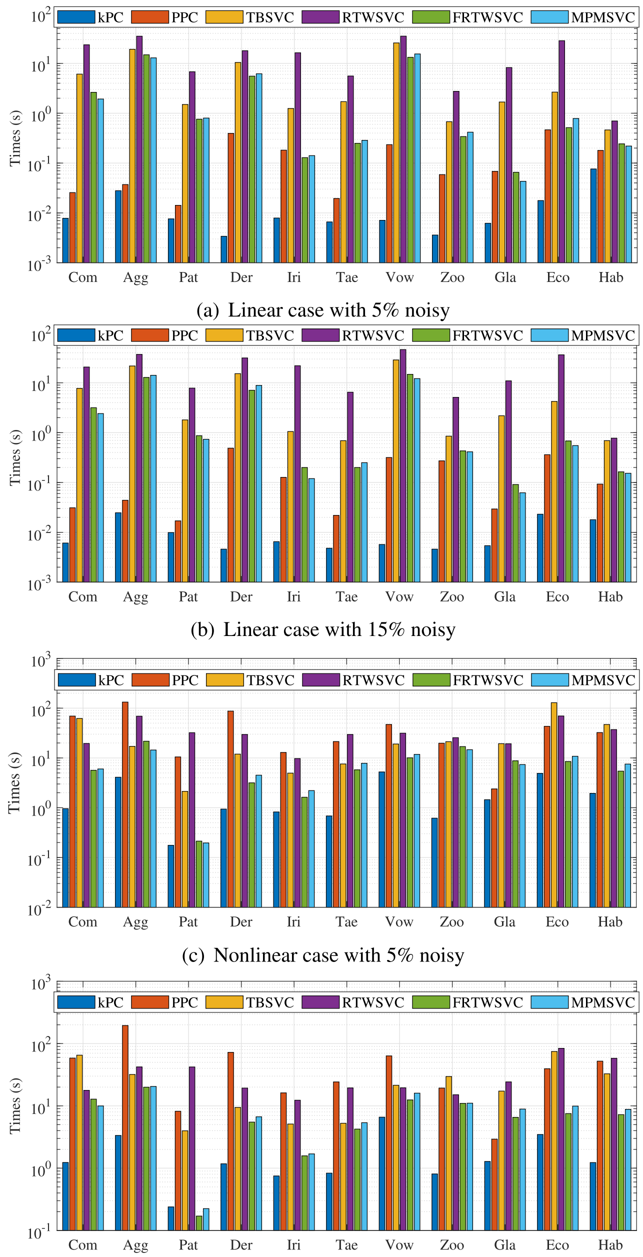
Then, substituting $k = 7, N = 22$ and (46) into (45), we have

$$\mathcal{X}_F^2 = \frac{12}{6(6+1)}\left[84.7931 - \frac{7(7+1)^2}{4}\right] \approx 70.9852 \tag{47}$$

Based on the above Friedman statistic $\mathcal{X}_F^2 = 70.9852$, we calculate the F-distribution statistic $\mathcal{F}_F$ with $(k-1, (k-1)(N-1)) = (5, 105)$ degrees of freedom as

$$\mathcal{F}_F = \frac{(N-1)\times\mathcal{X}_F^2}{N-(k-1)-\mathcal{X}_F^2} = \frac{21\times 70.9852}{22-6-70.9852} \approx 38.2110 \tag{48}$$

In the similar way, we calculate the statistic for the nonlinear case, which summarized in Table 5. The results reject the null hypothesis for both linear and nonlinear cases, and reveal the existence of significant differences among the performance of methods.



(a) Linear case with 5% noisy
(b) Linear case with 15% noisy
(c) Nonlinear case with 5% noisy
(d) Nonlinear case with 15% noisy

FIGURE 3. The learning times of each clustering methods on benchmark datasets.

Furthermore, we carry out the Holm's test [43] as a post-hoc test to validate whether there are the statistical differences between our MPMSVC and the remaining methods. Here, the significant level is set as 0.05. From Table 6, it can be seen that MPMSVC is statistically superior to the compared methods with the RI metric on both linear and nonlinear cases, expect for nonlinear RTWSVC.

Fig.3 recodes the average learning time of these methods on the above benchmark datasets. The results illustrate that

TABLE 5. Results of Friedman's test on benchmark datasets.

	Statistical value	p-value	Hypothesis
Linear	38.2110	3.61719E-22	Rejected
Nonlinear	42.0745	1.11022E-16	Rejected

TABLE 6. Results of the Holm's test on benchmark datasets.

	i	methods	z	p-value	Hypothesis
Linear	5	PPC	7.2524	4.0943E-13	Rejected
	4	kPC	6.0436	1.5064E-9	Rejected
	3	TBSVC	3.7873	1.5225E-4	Rejected
	2	FRTWSVC	2.3368	0.0194	Rejected
	1	RTWSVC	2.3368	0.0194	Rejected
Nonlinear	5	PPC	7.2524	4.0943E-13	Rejected
	4	kPC	5.6407	1.6930E-8	Rejected
	3	TBSVC	4.4320	9.3351E-6	Rejected
	2	FRTWSVC	2.3368	0.0194	Rejected
	1	RTWSVC	1.6116	0.1070	Accpeted

MPMSVC performs much faster than TBSVC and RTWSVC and has comparable efficiency with FRTWSVC at most cases. The reason behind is that MPMSVC is equipped with equality constraints, leading to an efficient close-form solution in each iteration. Overall, the above results confirm the feasibility of MPMSVC.

VI. CONCLUSION

In this paper, we propose an unsupervised MPMSVC method for noisy clustering tasks. Specifically, our MPMSVC utilizes the linear L_1-norm loss to derive the optimization problems to seek the K optimal nonparallel parametric-margin cluster plane-centers in section III-A. The utilization of the L_1-norm metric makes MPMSVC more robust to outliers than the L_2-norm plane-based clustering. Further, in section III-B, an iterative algorithm is presented to solve the non-smooth problem of MPMSVC. Also, the convergence of the proposed algorithm is analyzed theoretically in section III-C. To deal with more complex learning tasks, the nonlinear extension of MPMSVC is carried out via kernel trick in section IV. Extensive experimental results on noisy clustering datasets demonstrate the effectiveness of the proposed MPMSVC. Due to the proposed MPMSVC has one more parameters than PPC and TWSVC, designing more efficient parameter selection techniques will be one of our future works. Another interesting work is to consider some imbalance strategies to further improve the performance.

REFERENCES

[1] R. Xu and D. Wunsch, II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[2] V. Estivill-Castro, "Why so many clustering algorithms: A position paper," *ACM SIGKDD Explor. Newslett.*, vol. 4, no. 1, pp. 65–75, Jun. 2002.

[3] T. Velmurugan and T. Santhanam, "A survey of partition based clustering algorithms in data mining: An experimental approach," *Inf. Technol. J.*, vol. 10, no. 3, pp. 478–484, Feb. 2011.

[4] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015.

[5] E. Gungor and A. Ozmen, "Coarse segmentation with GDD clustering using color and spatial data," *IEEE Access*, vol. 8, pp. 144880–144891, 2020.

[6] F. Han, Z. Tu, and S.-C. Zhu, "Range image segmentation by an effective jump-diffusion method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1138–1153, Sep. 2004.

[7] Y. Li, C. Luo, and S. M. Chung, "Text clustering with feature selection by using statistical data," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 5, pp. 641–652, May 2008.

[8] L. F. D. C. Nassif and E. R. Hruschka, "Document clustering for forensic analysis: An approach for improving computer inspection," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 46–54, Jan. 2013.

[9] Y. Xiang, J. Hong, Z. Yang, Y. Wang, Y. Huang, X. Zhang, Y. Chai, and H. Yao, "Slope-based shape cluster method for smart metering load profiles," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1809–1811, Mar. 2020.

[10] M. C. de Souto, I. G. Costa, D. S. de Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: A comparative study," *BMC Bioinf.*, vol. 9, no. 1, pp. 1–14, Dec. 2008.

[11] T. Sajana, C. M. S. Rani, and K. V. Narayana, "A survey on clustering techniques for big data mining," *Indian J. Sci. Technol.*, vol. 9, no. 3, pp. 1–12, Feb. 2016.

[12] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[13] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3336–3341, Mar. 2009.

[14] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy C-means clustering algorithm," *Comput. Geosci.*, vol. 10, nos. 2–3, pp. 191–203, Jan. 1984.

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

[16] 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.

[17] 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] 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.

[19] 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.

[20] Q. Ye, Z. Li, L. Fu, Z. Zhang, W. Yang, and G. Yang, "Nonpeaked discriminant analysis for data representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3818–3832, Dec. 2019.

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

[22] 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.

[23] X. Peng, "TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2678–2692, Oct. 2011.

[24] 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, Oct. 2014.

[25] H. Yang, X. Yang, F. Zhang, Q. Ye, and X. Fan, "Infinite norm large margin classifier," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 9, pp. 2449–2457, Sep. 2019.

[26] 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.

[27] 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, Jan. 2013.

[28] 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.

[29] L. Bai, Y.-H. Shao, Z. Wang, and C.-N. Li, "Clustering by twin support vector machine and least square twin support vector classifier with uniform output coding," *Knowl.-Based Syst.*, vol. 163, pp. 227–240, Jan. 2019.

[30] R. Khemchandani, A. Pal, and S. Chandra, "Fuzzy least squares twin support vector clustering," *Neural Comput. Appl.*, vol. 29, no. 2, pp. 553–563, Jan. 2018.

[31] Q. Ye, H. Zhao, Z. Li, X. Yang, S. Gao, T. Yin, and N. Ye, "L1-norm distance minimization-based fast robust twin support vector k-plane clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4494–4503, Oct. 2018.

[32] Z. Wang, X. Chen, Y.-H. Shao, and C.-N. Li, "Ramp-based twin support vector clustering," *Neural Comput. Appl.*, vol. 32, no. 14, pp. 9885–9896, Jul. 2020.

[33] B. Richhariya and M. Tanveer, "Least squares projection twin support vector clustering (LSPTSVC)," *Inf. Sci.*, vol. 533, pp. 1–23, Sep. 2020.

[34] Z. Wang, Y.-H. Shao, L. Bai, C.-N. Li, and L.-M. Liu, "General plane-based clustering with distribution loss," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 2, 2020, doi: 10.1109/TNNLS.2020.3016078.

[35] Z.-M. Yang, Y.-R. Guo, C.-N. Li, and Y.-H. Shao, "Local K-proximal plane clustering," *Neural Comput. Appl.*, vol. 26, no. 1, pp. 199–211, Jan. 2015.

[36] H. Yang, X. Yang, F. Zhang, and Q. Ye, "Robust plane clustering based on L1-norm minimization," *IEEE Access*, vol. 8, pp. 29489–29500, 2020.

[37] L. Bai, Y. Shao, Z. Wang, W. Chen, and N. Deng, "Multiple flat projections for cross-manifold clustering," *IEEE Trans. Cybern.*, early access, Feb. 1, 2021, doi: 10.1109/TCYB.2021.3050487.

[38] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (CCCP)," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, 2002, pp. 1033–1040.

[39] W.-J. Chen, C.-N. Li, Y.-H. Shao, J. Zhang, and N.-Y. Deng, "Robust L1-norm multi-weight vector projection support vector machine with efficient algorithm," *Neurocomputing*, vol. 315, pp. 345–361, Nov. 2018.

[40] H. Yan, Q. Ye, T. Zhang, D.-J. Yu, X. Yuan, Y. Xu, and L. Fu, "Least squares twin bounded support vector machines based on L1-norm distance metric for classification," *Pattern Recognit.*, vol. 74, pp. 434–447, Feb. 2018.

[41] W. Chen, Y. Shao, C. Li, Y. Wang, M. Liu, and Z. Wang, "NPrSVM: Nonparallel sparse projection support vector machine with efficient algorithm," *Appl. Soft Comput.*, vol. 90, pp. 1–18, May 2020.

[42] N. Deng, Y. Tian, and C. Zhang, *Support Vector Machines: Theory, Algorithms and Extensions*. Boca Raton, FL, USA: CRC Press, 2013.

[43] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2013.

[44] Z. Yu, Z. Wang, J. You, J. Zhang, J. Liu, H.-S. Wong, and G. Han, "A new kind of nonparametric test for statistical comparison of multiple classifiers over multiple datasets," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4418–4431, Dec. 2017.

YI-BO JIANG received the B.S. and Ph.D. degrees from the College of Information, Zhejiang University of Technology, China, in 2004 and 2008, respectively. He is currently an Associate Professor with the College of Computer Science and Technology, Zhejiang University of Technology. His research interests include machine learning, intelligence computation, and stream-based multimedia networks.

WEI-JIE CHEN received the B.S. and Ph.D. degrees from the College of Information, Zhejiang University of Technology, China, in 2006 and 2011, respectively. He was a Visiting Scholar with the Centre for Artificial Intelligence (CAI), University of Technology Sydney, under the supervisor Prof. Ivor Tsang, from 2017 to 2018. He is currently a Full Professor with the Zhijiang College, Zhejiang University of Technology. He is also a member of the Optimal Group. He has published over 50 refereed articles. His research interests include pattern recognition, intelligence computation, and manifold learning.

YU-QING WANG received the B.S. degree from the College of Information Science and Engineering, Hebei North University, in 2019. She is currently a Graduate Student with the College of Computer Science and Technology, Zhejiang University of Technology. Her research interests include robust learning, support vector machine, and optimization methods.

MING-CHUAN ZHANG received the B.S. degree from the College of Electronic Information Science and Technology, Zhejiang Sci-Tech University, in 2020. He is currently a Graduate Student with the College of Computer Science and Technology, Zhejiang University of Technology. His research interests include optimization methods, feature selection, and machine learning.

YUAN-HAI SHAO received the master'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 the 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 Full 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 articles on these areas.