IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 6, 2020, accepted February 5, 2020, date of publication February 11, 2020, date of current version February 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973270

Robust Plane Clustering Based on L1-Norm Minimization

HONGXIN YANG, XUBING YANG, FUQUAN ZHANG, AND QIAOLIN YE, (Member, IEEE)
College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China

Corresponding author: Xubing Yang (xbyang@njfu.edu.cn)

This work was supported in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant SJKY19_0907, in part by the National Natural Science Foundation of China under Grant 31670554 and Grant 61871444, and in part by the Natural Science Foundation of Jiangsu under Grant BK20161527

ABSTRACT Plane clustering methods, typically, k plane clustering (kPC), play conclusive roles in the family of data clustering. Instead of point-prototype, they aim to seek multiple plane-prototype fitting planes as centers to group the given data into their corresponding clusters based on L2 norm metric. However, they are usually sensitive to outliers because of square operation on the L2 norm. In this paper, we focus on robust plane clustering and propose a L1 norm plane clustering method, termed as L1kPC. The leading problem is optimized on the L1 ball hull, a non-convex feasible domain. To handle the problem, we provide a new strategy and its related mathematical proofs for L1 norm optimization. Compared to state-of-the-art methods, the advantages of our proposed lie in 4 folds: 1) similar to kPC, it has clear geometrical interpretation; 2) it is more capable of resisting to outlier; 3) theoretically, it is proved that the leading non-convex problem is equivalent to several convex sub-problems. To our best knowledge, this opens up a new way for L1 norm optimization; 4) the k fitting planes are solved by k individual linear programming problems, rather than higher time-consuming eigenvalue equations or quadratic programming problems used in the conventional plane clustering methods. Experiments on some artificial, benchmark UCI and human face datasets show its superiorities in robustness, training time, and clustering accuracy.

INDEX TERMS L1 norm, plane clustering, eigenvalue problem, linear programming.

I. INTRODUCTION

As an important data analysis tool, clustering analysis is usually employed in understanding raw data, especially for unknown distribution. For a variety of purposes, people have proposed many methods in literatures, which were mainly divided into four categories [1]: hierarchical methods [2], [3], partitioning algorithms [4], [5], overlapping clustering procedures [6] and ordination techniques [7]. For instances, partition clustering methods, including k-means [8], k-median [9], fuzzy c-means [10] and some clustering ensemble methods (variants of those point-prototype clustering [40]–[42]), are widely studied with the fixed number of clusters [11]–[13]. They all take so-called point-prototype as cluster centers, and group the data into clusters by the similarities between data and their centers. For example, for a fixed number of clusters, k, k-means partitions n points into k clusters by the L2-norm point-to-point distance (typically, Euclidian distance) between points and k point-prototype centers. To distinguish them with the following plane clustering ones, hereafter we call them point-prototype clustering.

Instead of point-prototype, plane clustering methods take plane-prototype as cluster centers, which go back to k-Plane Clustering [14] (kPC). There have been increasing interests in plane clustering [15] in the last decade. Compared to k-means, kPC aims to seek k planes by minimizing the sum of the L2-norm distance between planes and their corresponding points. The leading problem is solved by k eigenvalue equations. In the line of kPC, Proximal Plane Clustering (PPC) [16] fuses inter-cluster information into optimization, and assigns a point to the cluster corresponding nearest plane and far away from the other planes. With heuristic selection for initial cluster centers, kernel PPC [17] (kPPC) discusses the problem in feature space by so-called kernel tricks. To improve the performance for clustering the points located at the plane-overlapped area, unsupervised transfer learning (but not clustering) [43] and Local kPPC [18] (LkPPC) introduce cost functions and add localized terms to their objectives, respectively. In doing so they expect to relieve overlapped-cluster errors caused by plane

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

infinite extensibility. Obviously, LkPPC has to cope with heavier training burden occurred in cluster centers initialization from constructing Laplacian graph and selecting kernel functions and kernel parameters. Furthermore, it also confronts the matrix singularity problems in the objective functions because of the difference between inter-cluster and intra-cluster Gram matrixes.

Unlike the aforementioned, another branch relaxes the constraint of kPC, and borrows ideas from supervised learning methods, including Support Vector Machine (SVM) [19], Proximal Support Vector Machine via Generalized Eigenvalue (GEPSVM) [20] and TWin SVM (TWSVM) [21]. For binary classification, for example, TWSVM aims at seeking two fitting planes such that each plane is closer to one of the two classes and is as far as possible from the other. The constraints in TWSVM require the plane to be at a distance of at least 1 from the other class (similar constraints in SVM). Inspiring by TWSVM, Twin Support Vector Clustering (TWSVC) [22] absorbs the foresaid constraints into plane clustering, and aims to make inter-cluster separable by setting the planes far away at least at a distance of 1 from the points of other clusters. Hence, its k fitting/clustering planes are solved by k quadratic programming (QP) problems. To speed training TWSVC, Fuzzy Least Squares TWSVC (F-LS-TWSVC) [23] relaxes the inequality constrains with equalities, and introduces fuzzy membership into the objective, thus it is analytically solved by linear equations.

Note that the foresaid methods are all based on L2 norm. It is well known that, to make the problems easier to be solved, people usually adopt square loss function or square operator on L2 norm to avoid square root problem. However, such square operation also exponentially amplifies the adverse effect on the data, especially for outliers. Inspiring by the successes of L1 norm based learning machines, besides feature extractors (typically, principal component analysis), they have been achieved more robustness than L2 ones in literatures [24], [25], [46], [47]. However, different from L2 norm optimization, it is a bigger challenge for solving L1 norm problem because of the its non-differentiability, especially in data clustering. Relaxing the fitting planes, Hyperplane clustering via dual principal component pursuit (HC-DPCP) [48], [49] aims to seek multiple projection planes, in view of (orthogonal) subspaces learning, by minimizing the projection distance with L2 norm constraint. Following the line of data fitting, another L1 norm plane clustering is our Fast Robust TWSVC (FRTWSVC) [26], which adopts TWSVC-like constraints, and incorporates inter-cluster information into the objective. The leading problems and its approximate version with equality constraints are solved by QP and linear equations, respectively. As for inter-cluster information used in PPC and TWSVC-like data clustering methods, is it indeed helpful for data clustering? In the view of data clustering, we know that, in the processing of data clustering, the relationship between a point and its cluster to which it temporarily belongs may be changed in the next updating steps. That is, such relationship is not fixed until data clustering terminated, whereas it is quite different from supervised learning, where the relationship (label) between for a given point and its class is fixed before training classifier. Intuitionally, once incorrect inter-cluster information, generated from wrong relationships between points and their clusters, is absorbed into clustering in the training phase, it may result in more time-consuming, even failure for data clustering. Furthermore, the cost in doing so is to lose original geometry of plane clustering methods: clustering the data to its cluster by minimizing distance of the data to and its nearest cluster plane.

In this paper, inheriting the geometrical interpretation of kPC, we propose a novel L1 norm k Plane Clustering (L1kPC). The leading problem is also a non-convex optimization. We provide an equivalent strategy that non-convex problem is decomposed into a series of convex sub-problems, it is solved by k linear programming (LP). In summary, the main contributions of our work are as follows.

L1kPC is a robust learning machine, owing to adopting L1 norm metric to characterize the plane clustering method. In addition, it also has clear geometrical interpretation.

The leading problem is solved by LP which leads to low-complexity.

It opens up a new way for solving L1 norm non-convex optimization problem. Different from the aforesaid plane clustering methods, the feasible region of L1kPC is non-convex, which makes the leading optimization non-convex. To handle the problem, an equivalent strategy is proposed. That is, it can be decomposed into several convex sub-problems. Besides, such equivalence proofs are provided in Section III.

Compared to state-of-the-art plane clustering methods, experiments on artificial and benchmark datasets demonstrate that L1kPC achieves better performance in robustness, training time and clustering accuracies.

The remainder of this paper is organized as follows. Section II briefly reviews related work. The L1kPC is described in Section III, including geometrical interpretation, model optimization, solutions and theoretical proofs. Experimental simulation and comparison are reported in Section IV. The conclusion is arranged in Section V.

II. RELATED WORK

In this section, we first review some related work about plane clustering methods.

A. NOTATIONS

For convenience, the symbols used in this paper are reported in table 1.

B. KPC: K-PLANE CLUSTERING

kPC aims to seek k planes as centers to fit k cluster samples. The k planes are defined as below:

$$P_i = \{\boldsymbol{x}|\boldsymbol{w}_i^T\boldsymbol{x} + \gamma_i = 0\}, \quad i = 1, 2, \cdots, k \tag{1}$$

The foresaid kPC leads to the following non-convex optimization:

$$\sum_{i=1}^{k} \min_{w_i,\gamma_i} \frac{1}{2}||A_i w_i + \gamma_i e||_2^2$$
$$s.t.\ ||w_i||_2^2 = 1, \quad i = 1, 2, \cdots, k \tag{2}$$

With random initialization for the plane parameter pairs (w_i, γ_i), $i = 1, 2, \cdots, k$, kPC alternatively run two procedures: cluster assignment and plane update. For the i-th cluster, A_i, under the constraint $||w_i|| = 1$, the expression $||A_i w_i + \gamma_i e||^2$ in formula (2) is just a sum of square distance between the points of A_i and its corresponding plane $w_i^T x + \gamma_i = 0$.

C. PPC: PROXIMAL PLANE CLUSTERING

PPC addresses the fitting planes by fusing inter-cluster and intra-cluster information, which leads to the following optimization:

$$\min_{(w_i,\gamma_i)} ||A_i w_i + \gamma_i e_i||_2^2 - C||\overline{A}_i w_i + \gamma_i \overline{e_i}||_2^2$$
$$s.t.\ ||w_i||_2^2 = 1, \quad i = 1, 2, \cdots, k \tag{3}$$

where $\overline{A_i}$ denotes the difference set $X - A_i$, and C is a regularization parameter. Since the quadratic matrix in the objective is a difference of two positive semi-definite matrixes respectively derived from intra-cluster and inter-cluster samples, it is not always to be positive. That is, when facing indefinite quadratic matrix problem, the objective function of (3) is non-convex, and thus its solution would miss theoretical support.

D. TWSVC: TWIN SUPPORT VECTOR CLUSTERING

TWSVC fuses inter-cluster information by inequality constraints generated from TWSVM [21]. For the i-th plane, it leads to the following optimization:

$$\min_{(w_i,\gamma_i)} ||A_i w_i + \gamma_i e_i||_2^2 + Ce^T \xi_i$$
$$s.t.\ |\overline{A}_i w_i + \gamma_i e_i| + \boldsymbol{\xi}_i \geq e_i, \quad \boldsymbol{\xi}_i \geq 0 \tag{4}$$

where $\boldsymbol{\xi}_i$ is a non-negative slack vector. The i-th constraint, $|\overline{A}_i w_i + \gamma_i e_i| + \xi_i \geq e_i$, means that the i-th plane is away at least at a distance of 1 from the points of other clusters, $\overline{A}_i$, when $\xi_i = 0$. Different from kPC and PPC, TWSVC describes intra-cluster compactness by minimizing $||A_i w_i + \gamma_i e_i||_2^2$. Without the constraint $||w_i||_2 = 1$, $||A_i w_i + \gamma_i e_i||_2^2$ cannot be interpreted by point-to-plane distance.

To attain the fast robust version TWSVC, the FRTWSVC replaces the L2 norm term in the objective of formula (4) and inequality constraint with L1 norm and equality constraint. Our discussed is obviously different from robust principal components analysis (PCA) [44], [45], which aims to seek multiple principle components by maximizing the projections, instead of the foresaid objectives, i.e. minimization of the sum of point-to-plane distance. They are suitable for coping with point cloud data [50], [51] rather than multiple plane-shaped data. The foresaid plane clustering methods, kPC, PPC and TWSVC, are all based on L2 norm. Among them, both kPC and PPC can be interpreted by point-to-plane distance, while TWSVC and FRTWSVC relax

TABLE 1. List of symbols used in the manuscript.

Symbol	Meaning
X	Original space data
R^d	Linear space of real number
n	Number of samples
d	Dimension of linear space
$\lVert\cdot\rVert_p$	p norm of a vector, usually set p = 1, 2 or ∞
e	Column vector with all entries ones at an appropriate size
w_i , γ_i	Normal vector and threshold of the i-th hyperplane
A_i	i-th cluster samples
m_i	Number of samples in A_i
k	Number of clusters
$\overline{A}_i$	Difference set $X - A_i$
$A_j^{(i)}$	j-th row of A_i
C	Regularization parameter
δ_i	Non-negative slack vector
H	Hypothesis space
$H(n)$	Growth function set
$B_H(n)$	Growth function quantity
y	Linear combination
θ_i	Coefficient vector of linear combination y
V	Affine set
L	Pseudo subspace
v , x_0	Fixed point belongs to R^d and L, respectively
T	Transpose of a matrix
I	Index set
$p(I)$	Power set of index set
$g(\cdot)$	Objective function of the optimization problem
z	Normal orthogonal basis corresponds to a pseudo subspace
z_i , z_j	Corresponding component of z
z_0	Center of the pseudo subspace
M	Initial value of the objective function
sgn($\cdot$)	Sign function
Q_i	Sub-region of the optimization problem
p	Group of normal orthogonal basis on the Q_i
α	Coefficient group of the normal orthogonal basis
m	i-th cluster center
h , η , D , B	Matrix corresponds to the linear programming model, respectively
I	Identity matrix
$[\cdot]^+$	Generalized inverse of a matrix
t	Number of iteration
ε	Tolerance factor
G	Predict label set
Q	Ground-true label set
$\overline{G}$, $\overline{Q}$	Complementary set of G and Q
f_{11} , f_{10} , f_{01} , f_{00}	Cardinality of two different set which selects from G , Q , $\overline{G}$ and $\overline{Q}$
q	Number of non-zero elements

such geometrical interpretation. In consideration of geometrical meaning and training time, in this paper, we propose a novel L1kPC method.

III. L1KPC

Based on our previous work [29], the infinite norm point-to-plane distance derived from its dual L1 norm, it motivates us to design L1 norm plane clustering algorithm. Likewise, it is also helpful for maintaining the geometric interpretation, as described in the kPC. That is, L1kPC also aims to seek k planes by minimizing the sum of the infinite norm distance, rather than L2 norm between planes and points of their corresponding clusters. Obviously, it is a genuine point-to-plane distance derived from the infinite norm, as described in Eq. (5). For the i-th cluster, suppose the corresponding fitting plane is derived from the following problem:

$$\min_{\mathbf{w}_i, \gamma_i} ||\boldsymbol{A}_i \boldsymbol{w}_i + \gamma_i \boldsymbol{e}||_1 \quad s.t.\ ||\boldsymbol{w}_i||_1 = 1 \tag{5}$$

Obviously, the feasible region of (5), called a L1 ball hull, is non-convex. So the optimization problem is also non-convex.

A. GEOMETRICAL INTERPRETATION OF L1KPC

According to the definition of L1 norm, L1 norm of a given vector is equals to the sum of its absolute components. Thus, by substituting $||\boldsymbol{w}_i||_1 = 1$ into the objective, we rewrite (5) as below.

$$\sum_{j=1}^{m_i} \frac{|\boldsymbol{A}_j^{(i)} \boldsymbol{w}_i + \gamma_i|}{||\boldsymbol{w}_i||_1} \tag{6}$$

where $\boldsymbol{A}_j^{(i)}$ denotes the j-th row of $\boldsymbol{A}_i$, corresponding to the j-th sample of the i-th cluster. The j-th term $|\boldsymbol{w}_i^T \boldsymbol{A}_j^{(i)} + \gamma_i|/||\boldsymbol{w}_i||_1$, is just equal to the infinite norm distance between the point $\boldsymbol{A}_j^{(i)}$ and the plane $\boldsymbol{w}_i^T \boldsymbol{x} + \gamma_i = 0$. That is, the objective of (5) is to minimize the sum of the infinite-norm distance, which is described in Theorem 1.

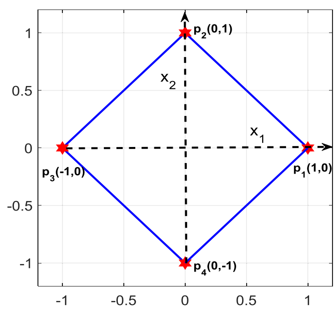
***Theorem 1**: For a given point $\boldsymbol{v} \in R^d$ and a hyperplane $\boldsymbol{w}^T \boldsymbol{x} + \gamma = 0$, the point-to-plane distance based on infinite norm is*

$$\frac{|\boldsymbol{w}^T \boldsymbol{v} + \gamma|}{||\boldsymbol{w}||_1} \tag{7}$$

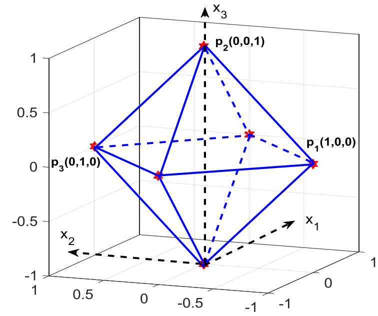
The proof for Theorem 1 refers to our previous work [29]. As foresaid, due to non-convex feasible region of the problem (5), it is difficult to directly solve the optimization in the cluster updating procedure. Next, we will introduce a strategy to handle it. That is, this non-convexity is transformed into a series of convex sub-problems.

B. TRANSFORMATION STRATEGY

Geometrically, the constraint of formula (5), $||\boldsymbol{w}_i||_1 = 1$, is a L1 ball hull. Fig.1 illustrates a toy for the L1 hulls in 2-dimensional (Fig.1a) and 3-dimensional (Fig.1b) cases. The points, marked red stars, stand for convex vertexes, and the borders surrounded by blue solid line segments are called L1 ball hull. Here the hull is just a surface of L1 ball, not convex L1 ball. Obviously, the 2-dimensional L1 hull is just a square composed of 4 line segments, while for the 3-dimensional hull, it consists of 8 equilateral triangles. On each sub-region (line segment in Fig. 1a or triangle in Fig. 1b) of the L1 hull, it is convex. Thus, non-convex L1 hull in formula (5) is divided into multiple convex sub-region.



(a)



(b)

FIGURE 1. L1 ball hull. (a) and (b) stand for 2 and 3 dimensional hull, respectively.

Without loss of generalization, let us discuss the hull in d-dimensional linear space. Suppose a L1 hull in linear space R^d, its vertex set consists of d pairs of vertexes, whose coordinates are noted by $\{(\pm 1, 0, \cdots, 0), (0, \pm 1, \cdots, 0), \cdots\cdots, (0, 0, \cdots, \pm 1)\}$. A subset is composed by d vertexes sampling from d pair of vertexes. The vector group corresponding to such subset is linearly independent, and the space spanned by the subset is a subspace. In the spanned subspace, as illustrated in Fig.1, the bounded sub-region generated by vertex subsets of L1 hull is just convex. The points in sub-region are linearly represented by convex combination of the vertexes of the L1 hull.

To reach the goal of foresaid transformation, there exist two problems to be solved: how many subspaces are there in the d-dimensional L1 hull and how to rule a search order for these subspaces? The main processes for above problems are divided into three parts: 1) there exists a one-to-one map between convex vertex set and so-called growth function set (see Definition 1); 2) the number of subspaces equals to growth function quantity (Definition 2); 3) a search order for subspaces is equivalent to solve a power set for growth function set.

Definition1: For a binary sample set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$, the set $H(n) = \{(h(\boldsymbol{x}_1), h(\boldsymbol{x}_2), \cdots, h(\boldsymbol{x}_n)) | h \in \boldsymbol{H}\}$ on $\boldsymbol{X}$ is called *Growth function set*, where H denotes hypothesis space.

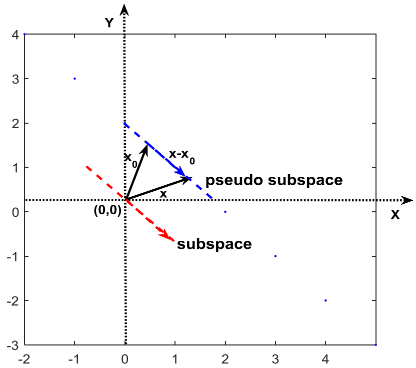


FIGURE 2. Illustration for pseudo subspace and subspace.

Definition 2: A quantity for growth function set defined as $B_H(n) = \max_X |H(n)|$ is called *growth function quantity*.

For instance, for any given two-class data set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ drawn from a distribution $\boldsymbol{D}$, if any subsets of $\boldsymbol{X}$ is scattered by H, then growth function quantity $B_H(n) = 2^n$.

The first question, i.e., the number of subspaces, is answered by the above definitions. Before replying to the second question, a search order for subspaces, we review the conceptions about affine set and pseudo subspace in Definition 3 and 4.

Definition 3: A set V is called *affine set*, if a group of vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_d \in V$ and satisfy $\sum_{i=1}^{d} \theta_i = 1$, then the linear combination $\boldsymbol{y} = \sum_i \theta_i \boldsymbol{x}_i$ also belongs to V.

Definition 4: An affine set L is called *pseudo subspace* [30], for any fixed point $\boldsymbol{x}_0$ belongs to $L(\forall \boldsymbol{x}_0 \in L)$, where $L - \boldsymbol{x}_0$ is a subspace.

For any point in the pseudo subspace, spanned by linearly independent hull vertexes, is linearly represented by those vertexes. Thus, in the strict sense, the foresaid convex sub-region is not a subspace but a pseudo one, as illustrated in Fig.1, because it has no zero elements. If it is traversing through the origin of the coordinates, it is a subspace, Here $L - \boldsymbol{x}_0$ is to make such translation operation, as illustrated in Fig.2, where the blue dash line above coordinate origin (0,0) is a so-called affine set L. For a given fixed point $\boldsymbol{x}_0$ ($\in L$) and the pseudo subspace L, the set $\{\boldsymbol{x} - \boldsymbol{x}_0 | \boldsymbol{x} \in L\}$ is a subspace (shortly $L - \boldsymbol{x}_0$), located at the line marked red dash through the origin. Geometrically, it is just a translation between L and $L - \boldsymbol{x}_0$, where $\boldsymbol{x} - \boldsymbol{x}_0$ acts as zero point when $\boldsymbol{x} = \boldsymbol{x}_0$.

To answer the second question, we conclude the above in the following theorems, including how to rule an order for searching pseudo subspaces.

Theorem 2: *The vector group composed of those d vertexes is a* ***normal orthogonal basis*** *for the d dimensional linear space.*

Theorem 3: *A convex combination L spanned by d linearly independent L1 hull vertexes is a pseudo subspace, and the total of pseudo subspaces spanned by a normal orthogonal basis of d-dimensional linear space R^d is at most 2^d.*

Theorem 4: *An order for searching 2^d pseudo subspaces is equivalent to solve a power set of a set composed of d elements.*

Algorithm 1 Convex Set Search Algorithm

Input: Index set $I = \{1, 2, \cdots, d\}$, set default z with $z_i = -1$, and initial objective value $M = g(z/d)$
Output: The center of pseudo subspace z_0

Step 1. Compute the power set $P(I)$ and set $z_0 = z/d$
Step 2. For $i = 1$ to $|p(I)|$
2.1 Reset all components of z with -1
2.2 If the indexes of I belong to the i-th element of $P(I)$, then set the corresponding components of z to 1 and recalculate $g(z/d)$
2.3 If $M > g(z/d)$, then compute $M = g(z/d)$ and let $z_0 = z/d$
// repeat 2.1∼2.3, until the all components in z turn into element 1
Step 3. Return z_0

Theorem 4 says that, when a search order is fixed, the optimization for sub-problems is finished in a linear time, at most 2^d searches, where 2^d is the total number of subspaces. The algorithm is described as below in d-dimensional linear space R^d. Define an index set $I = \{1, 2, \cdots, d\}$ and note its power set as $p(I)$. The function $g(\cdot)$ denotes the objective of (5). A vector $z = (z_1, z_2, \cdots, z_d)^T$ ($z_i \in \{-1, 1\}$) denotes a normal orthogonal basis corresponding to a pseudo subspace. Firstly set the default value to each component of z with $z_i = -1, i = 1, 2, \cdots, d$, then change some components to 1 by the search order of the power set $p(I)$. That is, if the index j belongs to the element of $p(I)$, we set corresponding component z_j to 1. Simultaneously, to avoid solving 2^d convex sub-problems in corresponding pseudo subspaces, we need to estimate objective values by the centers of pseudo subspaces z/d. Algorithm 1 is described as below.

After once traversal, algorithm 1 returns a pseudo subspace center z_0, corresponding to the minimum value estimation of objective function. Taking the signs of z_0 by sgn(z_0), it is easy to know which pseudo subspace is used for further optimization, where sgn($\cdot$) denotes the sign function.

C. SOLUTION FOR L1kPC

From Algorithm 1, we obtain the corresponding normal orthogonal basis, and note them as a group of vectors $(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_d)$. Assume the following optimization is on the i-th subspace, Q_i, a feasible sub-region of the problem (5), spanned by basis vectors. Q_i is defined as below,

$$Q_i = \{\boldsymbol{w} = \sum_{j=1}^{d} \alpha_j \boldsymbol{p}_j | \alpha_j \geq 0, \quad \sum \alpha_j = 1\} \tag{8}$$

Recalling the feasible region Ω in formula (5), obviously, the expression $\Omega = \cup Q_i$ holds. Thus, the formula (5) are rewritten as

$$\begin{aligned} &\min_{\boldsymbol{w}_i, \gamma_i} ||\boldsymbol{A}_i \boldsymbol{w}_i + \gamma_i \boldsymbol{e}||_1 \\ &s.t.\ \boldsymbol{w}_i \in \bigcup_j Q_j \end{aligned} \tag{9}$$

According to the order provided by the theorem 4, the optimization (5) is solved by a series of convex sub-problems

Algorithm 2 Training L1 k PC

Input: Data points $\boldsymbol{X}$ and initial k random planes $(\boldsymbol{w}_i, \gamma_i)$, $i = 1, 2, \ldots k$.
Output: $(\boldsymbol{w}_i, \gamma_i)$

Step1. Assign $\boldsymbol{X}$ into k clusters by L1-norm point-to-plane distance.
//Cluster assignment
Step2. For $i = 1$ to k
// Plane update
2.1 Call Algorithm 1 to get subspace center z_0, and its corresponding normal orthogonal basis $\boldsymbol{p}$ according to the formula (8)
2.2 Calculate $\boldsymbol{B}$ and $\boldsymbol{D}$ by formula (11) to obtain LP solution $\boldsymbol{\eta}$.
2.3 According to the formula (8), updating the i-th plane $(\boldsymbol{w}_i, \gamma_i)$.
//repeat Step1$\sim$ Step2
Until terminal condition is satisfied.

corresponding to the sub-region Q_i. To simplify the problem, let the fitting plane passing through the i-th cluster center $\boldsymbol{m}$, i.e., $\gamma_i = -\boldsymbol{m}\boldsymbol{w}_i$. For each sub-problem, substituting it and formula (8) into the optimization problem (5), we ignore some subscripts for describing problem under without causing ambiguity, and have the following optimization:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} ||(\boldsymbol{A} - \boldsymbol{e}\boldsymbol{m})\boldsymbol{P}\boldsymbol{\alpha}||_1 \\ &s.t.\ \boldsymbol{\alpha} \geq \boldsymbol{0} \\ &\quad \boldsymbol{e}^T\boldsymbol{\alpha} = 1 \end{aligned} \tag{10}$$

Theorem 5: The solution of problem (10) is equivalent to that of the following LP

$$\begin{aligned} &\min_{\boldsymbol{\eta}} \boldsymbol{h}^T\boldsymbol{\eta} \\ &s.t.\ \boldsymbol{D}\boldsymbol{\eta} \geq \boldsymbol{0} \\ &\quad \boldsymbol{e}^T\boldsymbol{B}\boldsymbol{\eta} = 1 \end{aligned} \tag{11}$$

where $\boldsymbol{B} = [(\boldsymbol{A} - \boldsymbol{e}\boldsymbol{m})\boldsymbol{P}]^+ [\boldsymbol{I} - \boldsymbol{I}]$, $\boldsymbol{D} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{I} \end{bmatrix}$, and $\boldsymbol{I}$ denotes identity matrix at appropriate size.

The above procedure is concluded in Algorithm 2.

In usual, the terminal conditions lie in 3-fold: 1) the fitting planes tend to be stable, popularly measured by L1 norm like $||\boldsymbol{w}_i^{t+1} - \boldsymbol{w}_i^t||_1 < \varepsilon$, where t means the t-th iteration and ε is a tolerance factor. 2) the membership between points and their clusters is unchanged any more. 3) maximum iteration is set to avoid lower convergence rate, especially for "bad" random initialization. In the next experiment section, we adopt both 1) and 3) as terminal conditions.

IV. EXPERIMENT

In order to evaluate the performance of L1kPC, in this section, we conduct some experiments to validate our method on the artificial and benchmark datasets [31]. To report comparison, we take foresaid data clustering methods kPC, PPC, TWSVC and FRTWSVC as base line. All methods were implemented on the MATLAB 2015b platform running on the PC with Intel 2.60 GHz CPU and 4GB RAM. The clustering accuracy is defined in (12) as described in Refs [17], [32]. The symbols, G and Q, denote predict label set and ground-true label set, respectively.

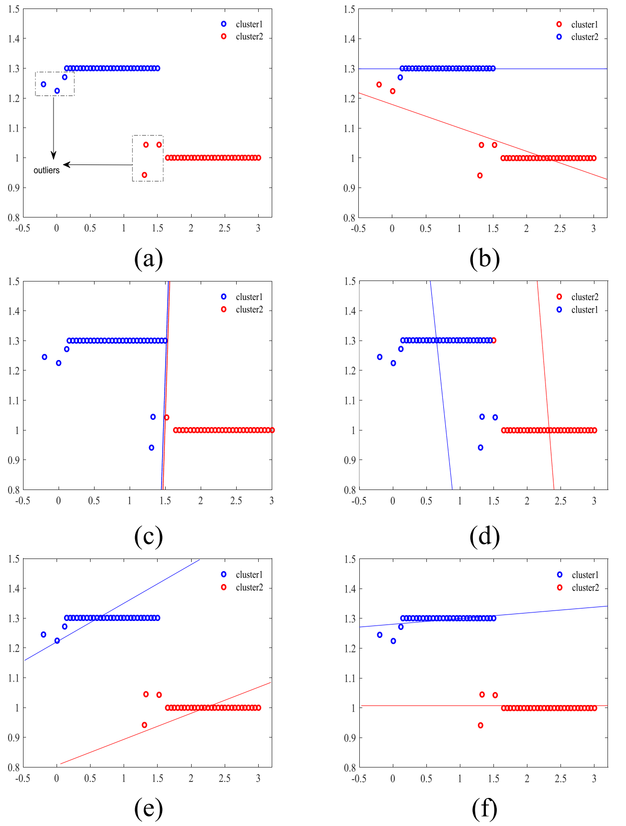


FIGURE 3. Illustration for clustering results of two clusters, (a) original data distribution, (b) kPC, (c) PPC, (d) TWSVC, (e) FRTWSVC, and (f) L1kPC.

$$\text{accuracy} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}} \tag{12}$$

where f_{11} is the cardinality of the set of $G \cap Q$, i.e., $f_{11} = |G \cap Q|$, and $\cap$ means set intersection. Similarly, we set $f_{10} = |G \cap \bar{Q}|$, $f_{01} = |\bar{G} \cap Q|$, and $f_{00} = |\bar{G} \cap \bar{Q}|$, where $\bar{Q}$ stands for complementary set of Q.

A. ARTIFICIAL DATA

To validate the robustness of our L1kPC, we compared our proposed with related plane clustering methods (kPC, PPC, TWSVC and FRTWSVC) on artificial datasets named NoiseData and cross3D. Fig.3 illustrates a toy on the NoiseData, drawn from two-class linear-shaped distribution plus several outliers, and marked red "o" and blue "o", respectively. Each class consists of 69 points including 3 outliers, marked "outliers" in Fig. 3a.

The fitting planes and their corresponding clusters generate from five plane clustering methods, kPC, PPC, TWSVC, FRTWSVC and L1kPC, as illustrated in Fig.3 (b-f), respectively. kPC, PPC and TWSVC obtain 97.22%, 95.83%

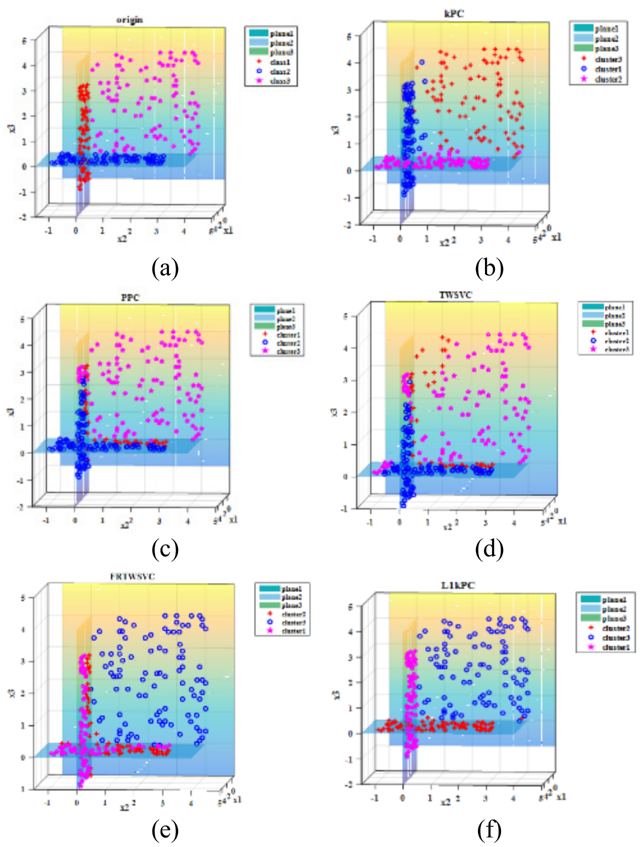


FIGURE 4. **Clustering results of five methods on the cross3D dataset. (a) original distribution, (b) *k*PC, (c) PPC, (d) TWSVC, (e) FRTWSVC and (f) L1*k*PC.**

and 94.44% while FRTWSVC and L1kPC achieve 100% clustering accuracies respectively. For kPC, its fitting plane for cluster2 (blue solid line) almost correctly reflects the linear tendency of class1 data, while the plane for cluster1 has heavily deviated from the data distribution of the class2 data. Benefiting from L1 norm, FRTWSVC assigns points to correct clusters, its corresponding plane does not fit data well while the two fitting planes of L1kPC are capable of reflecting data original distribution, and the points in each class are assigned into the corresponding cluster. The result is in line with our expectations, because squared L2 norm in kPC exaggerates the effect of outliers.

The cross3D is from three-class plane-shaped distributions plus 10 percentage of uniform noise, where one plane is separate from the other two planes orthogonal to each other, as illustrated in Fig.4a. Points of each class consist of 100 samples, marked red "+", blue "o" and magenta "☆". The parameter C in PPC, TWSVC and FRTWSVC is selected from the values $\{2^i | i = -5, -4, \cdots, 4\}$. The clustering results are displayed in Fig.4.

Fig.4 reports that L1kPC, kPC, PPC, TWSVC and FRTWSVC obtain 99.67%, 98.00%, 59.00%, 60.91% and 71.67% clustering accuracies respectively. Compared with PPC, TWSVC and FRTWSVC, both L1kPC and kPC are more capable of closing to the original distribution of the dataset. While for PPC, TWSVC and FRTWSVC, due to adding inter-to cluster information to cluster points which corresponds nearest planes and simultaneously far away from the other planes, it is difficult to assign points located at overlapped area to their correct clusters.

TABLE 2. **UCI data information.**

Dataset	#Samples	#Dimensions	#Classes
Tae	151	5	3
Liver	345	6	2
Wine	178	13	3
Hab	306	3	2
Iris	150	4	3
PID	768	8	2
Vow	528	11	11
Mok	432	6	2
Veh	846	19	4
Bals	625	4	3
User	403	5	4
Zoo	101	17	7
Derm	366	33	6
Aus	690	14	2
Burst	1075	21	3
Waveform	5000	21	3
Pushing	11055	31	2
Letter	20000	17	26

B. UCI DATASETS

In this subsection, we further compared kPC, PPC, TWSVC, FRTWSVC and L1kPC on eighteen UCI datasets. The details of UCI datasets briefly described in Table 2. The average results of clustering performance (Test Acc), training time (Train Time), standard deviation (std) and p-value are reported in Table 3, where the highest accuracy is bold. The symbol "-" in the cell of table 3 means unavailable results, where training time is at least beyond 24 hours. To further validate them, the paired t-test between our L1kPC with other four methods are listed in Table 3. The threshold for t-test is set to 0.05. The p-value for each test is the probability of the observed or a greater difference occurring between the Train Time of the two methods, under the assumption of the null hypothesis that there is no difference between the Train Time distributions. Hence, the smaller the p-value, the less likely it is that the observed different resulted from datasets. The Train Time means CPU time (in seconds) for training five plane clustering methods, and clustering accuracies are reported by percentage (%) according to the formula (12).

The two groups of clustering methods are averaged in Table 3, without inter-cluster and with inter-cluster information, marked Without-Avg and With-Avg. Divide five clustering methods into two groups: one is to seek k fitting planes only by intra-cluster information, such as kPC and L1kPC. The other is fusing inter-cluster and intra-cluster information, including PPC, TWSVC and FRTWSVC to seek fitting planes. Table 3 reports L1kPC achieves higher clustering accuracies on 10 out of 16 datasets. For instance, on the dataset Hab, L1kPC achieves 75.00%, while for kPC, PPC, TWSVC and FRTWSVC, achieve 54.57%, 60.95%,

TABLE 3. Comparison among five plane clustering methods on UCI datasets.

Dataset	*k*PC	PPC	TWSVC	FRTWSVC	L1*k*PC	Without-Avg	With-Avg
	Train Time Test Acc ± std *p*-value	Train Time Test Acc ± std *p*-value	Train Time Test Acc ± std *p*-value	Train Time Test Acc ± std *p*-value	Train Time Test Acc ± std		
Tae	0.0071 55.46 ± 1.78 0.0074	0.0090 54.84 ± 2.95 0.014	1.5300 55.96 ± 2.03 0.046	0.3800 56.83 ± 3.68 0.200	**0.0016** **57.65 ± 1.65**	**56.56***	55.88
Liver	0.0069 50.31 ± 5.14 0.013	0.0076 51.22 ± 2.58 0.0012	7.7100 51.32 ± 3.79 0.012	1.4100 51.54 ± 3.70 0.02	**0.0035** **56.41 ± 2.06**	**53.36***	51.36
Wine	0.0053 56.39 ± 3.02 0.00032	0.0072 73.17 ± 1.90 6.04E-10	6.2600 73.65 ± 3.15 2.89E-08	0.5000 **82.14 ± 1.52** 4.77E-10	**0.0033** 50.00 ± 3.64	53.20	**76.32***
Hab	**0.0032** 54.57 ± 2.61 1.35E-09	0.0065 60.95 ± 2.75 5.93E-08	1.5700 61.26 ± 3.07 1.86E-07	0.2100 62.54 ± 2.57 9.45E-08	0.0064 **75.00 ± 0.00**	**64.79***	61.58
Iris	**0.0033** 57.36 ± 2.75 0.093	0.0068 83.68 ± 2.98 2.59E-09	1.7000 **91.24 ± 3.53** 1.93E-11	0.2200 **91.24 ± 1.64** 1.86E-10	**0.0033** 60.00 ± 2.67	58.68	**88.72***
PID	**0.0048** 58.80 ± 3.14 0.001	0.0200 54.74 ± 3.04 3.64E-05	0.0170 54.43 ± 3.37 8.12E-05	1.7800 55.94 ± 2.86 0.0002	0.0100 **63.10 ± 1.26**	**60.95***	55.04
Vow	**0.0230** 83.13 ± 2.73 0.0046	0.0660 80.83 ± 3.15 0.0011	1182.7 83.09 ± 1.95 0.00014	5.5600 83.25 ± 4.87 0.049	0.0960 **87.50 ± 1.91**	**85.32***	82.39
Mok1	0.0048 49.88 ± 3.71 0.024	0.0036 52.90 ± 5.00 0.70	2.2200 50.50 ± 3.67 0.034	0.1300 52.36 ± 3.84 0.34	**0.0025** **53.70 ± 2.44**	51.79	**51.92***
Veh	0.1600 51.58 ± 2.79 4.23E-05	**0.0430** 60.29 ± 3.07 0.50	311.56 59.85 ± 3.46 0.39	1.2400 60.65 ± 4.57 0.79	0.1200 **61.13 ± 1.77**	56.36	**60.26***
Bals	0.0488 55.81 ± 6.47 0.084	0.0057 52.73 ± 3.53 0.00035	0.3180 55.82 ± 2.41 0.0081	2.8641 57.28 ± 10.84 0.40	**0.0045** **60.15 ± 3.06**	**57.98***	55.28
User	0.0423 62.09 ± 2.98 0.00038	**0.0057** 65.35 ± 2.32 0.0004	323.72 61.17 ± 3.13 0.00030	14.845 **68.36 ± 1.10** 0.41	0.1489 67.69 ± 1.71	64.89	**64.96***
Zoo	0.0199 55.39 ± 3.08 0.059	0.0527 80.75 ± 3.23 9.27E-08	1.3128 88.91 ± 4.25 4.46E-09	0.8427 **88.99 ± 2.53** 7.72E-11	**0.0147** 58.93 ± 3.45	57.16	**86.22***
Derm	**0.0280** 49.48 ± 3.51 0.0013	0.1534 62.43 ± 2.78 0.021	451.674 70.13 ± 4.22 0.00067	39.5285 **76.24** ± 1.36 5.52E-08	0.5245 58.45 ± 3.64	53.97	**69.60***
Aus	0.0439 50.06 ± 3.56 0.00014	0.1953 51.45 ± 5.56 0.005	234.674 50.13 ± 3.79 0.00066	24.6234 54.13 ± 3.66 0.012	**0.0156** **57.75 ± 2.32**	**53.91***	51.90
Burst	0.0100 32.83 ± 2.24 5.18E-09	**0.0034** 32.68 ± 4.08 5.28E-01	176.630 51.31 ± 3.37 0.22	30.3223 32.68 ± 3.01 1.41E-07	74.070 **52.30 ± 2.08**	**42.57**	38.89
Waveform	2.8329 58.74 ± 3.39 0.04	**0.0627** 59.35 ± 4.07 0.05	87.9180 60.13 ± 4.54 0.14	90.4391 **65.26** ± 2.31 0.92	35.700 65.04 ± 6.13	**61.89**	61.58
Pushing	897.27 55.70 ± 3.98	- -	- -	- -	853.91 60.27 ± 4.37	57.99	-
Letter	2098.53 75.32 ± 3.31	- -	- -	- -	1653.24 86.61 ± 2.74	80.97	-

"-" means unavailable results, where training time is at least beyond 24 hours.
* Best results are bold.

61.26% and 61.54%, respectively. It is almost as 14 or even bigger percentage points higher as the other four methods. In addition, the *p*-value is 1.35E-09, 5.93E-08, 1.86E-07 and 9.45E-08, respectively. The similar results can be observed on the datasets Derm, Vow, PID, Wine and Liver. This indicates the L1*k*PC has significant differences from the other plane clustering methods. Although FRTWSVC obtains higher accuracies than L1*k*PC on the datasets Tae and Waveform, the differences between them is about 2 percentage points and the *p*-value is 0.2 (>0.05) and 0.92 (>0.05), respectively. That is, there has no significant difference between L1*k*PC and FRTWSVC. However, on the datasets Wine, Iris and Zoo, FRTWSVC is far superior to L1*k*PC. A reasonable explanation is that inter-cluster information is helpful for FRTWSVC

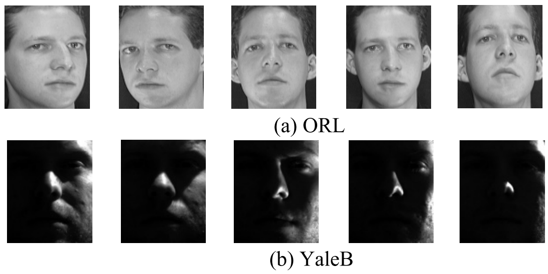


FIGURE 5. Example test images of ORL and YaleB.

TABLE 4. Results of the clustering methods on face datasets.

Dataset	*k*PC	PPC	TWSVC	FRTWSVC	L1*k*PC
ORL	81.33	82.87	86.34	88.94	**90.50**
YaleB	78.60	78.48	78.38	78.92	**79.32**

to improve clustering performance in these cases. Thus, we also record the accuracies of Without-Avg and With-Avg to measure the performance of above two types plane clustering algorithms. It is easily observed that Without-Avg achieve better clustering accuracies than With-Avg on 9 out of 16 datasets without regard to the results located in the last two lines. So is it better for fusing inter-cluster information into clustering objective? It is still an open problem.

C. HUMAN FACE DATA

To further verify the performance of above methods, we also compare these methods on the face dataset "ORL" [35] and "Yale B" [36]. ORL includes total 400 face grey images from 40 classes/persons (10 images for each person) with the size 112×92 image resolution, while YaleB includes 2414 faces from 38 classes and each face is a grey image with the size 32×32. The rest 5 images for each class in ORL and YaleB are used for testing. Some example images are shown in Fig.5.

Table 4 shows the test accuracy of five plane clustering methods on the face datasets. As far as the accuracy is concerned, L1*k*PC is obviously better than other four plane clustering algorithms. And the main reason is, the L1*k*PC has good robustness, which avoids the effect of noise caused by shadow in the image on recognition.

D. TIME COMPLEXITY

As far as Train Time is concerned, L1*k*PC runs fast on 8 out of 16 datasets, and *k*PC and PPC runs fast on the rest 5 and 4 datasets, respectively. In the view of computation, the Train Time of L1*k*PC is composed of two parts: one is for searching appropriate subspaces and the other is for computing fitting planes. The former can be finished in linear time, $O(2^d)$, as proved in *Theorem 4*, while the latter depends on LP. The time complexity of simplex method for LP is at most $O(n^2)$ [33]). Considering the formula (11) is also a sparse LP problem, its time complexity is decreased to the order of $O(nq*\min(n, q))$, where q denotes the number of non-zero elements [34]. In the real world, if satisfying $n \gg d$ and $n > q$, the time complexity of L1*k*PC will be decreased to $O(nq^2)+O(2^d)$. Both *k*PC and PPC need to solve eigenvalue equation, and their time complexities are $O(n^3)$. Owing to

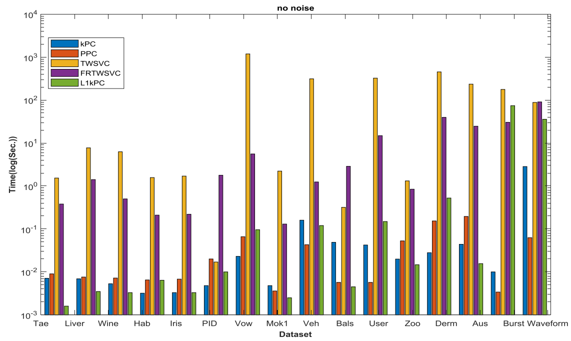


FIGURE 6. Train Time of sixteen UCI dataset.

TABLE 5. The average training time on 16 UCI datasets and *P*-values for paired *t*-test of training time.

Noise Ratio	L1*k*PC Train time	*k*PC Train time *p*-value	PPC Train time *p*-value	TWSVC Train time *p*-value	FRTWSVC Train time *p*-value
0% noise	6.9203#	0.203# 0.19	0.041# 0.19	174.47# **0.046***	13.43# 0.23

\# average training time on 16 datasets
* *p*-value less than 0.05

L2 norm QP optimization by interior point method, the time complexity for TWSVC is of the order $O(n^{3.5})$ [20], while the FRTWSVC, $O(tn^3)$, where t is the total of iteration [26].

In addition, The Train Time show in Fig.6 reflects that the real CPU time does not coincide with the foresaid time complexity. The reason is that *k*PC, PPC and FRTWSVC can be analytically solved by eigenvalue or linear equations, while the other two, L1*k*PC and TWSVC, need to be iteratively computed by LP or QP problems. An asterisk (*) indicates a significant difference from L1*k*PC, which corresponds to the *p*-value is less than 0.05.

For example, the *p*-value of the *t*-test between L1*k*PC and TWSVC is 0.046 (<0.05), while their Train Time is 6.9203s and 174.47s, respectively in Table 5. This means that there exists a significant difference between them. Although Table 5 that *k*PC and PPC run fast than L1*k*PC, the *p*-value of the *t*-test between them are all higher than 0.05, which means there is no significant difference. However, the *p*-value between L1*k*PC vs. TWSVC is 0.046, which means TWSVC significantly runs slower than L1*k*PC because of *p*-value < 0.05.

In the end, we should point out that there exist two situations to speed training L1*k*PC. One is in the step of searching subspaces, where we ignore the time for repeat searching in the same subspaces. Another is due to sparse matrixes [52], [53] existing in the constraints of the formula (11), the leading LP problem will be speeded if combining sparse optimization methods. There have some methods for L1 norm convex optimization methods such as Gradient Projection (GP) and Proximal Gradient (PG) [37], it is proved that they run faster than LP. Furthermore, since the subspace search in the power set is viewed as feature selection problem, some heuristic strategies may be helpful to speed training L1kPC, including sequential forward or backward selection [38], [39]. These will be our next work.

V. CONCLUSION

Instead of point-centered, plane data clustering groups data into clusters by seeking multiple fitting planes as its centers. In this paper, we follow the geometry of kPC and propose a robust plane clustering method based on L1 norm, termed as L1kPC. To handle the non-convex L1 norm minimization problem, a new optimization method is provided. The leading non-convex L1 minimization problem is decomposed into multiple convex sub-problems. Hence, the k fitting planes are solved by k linear programming problems. In view of computation, we open a new way for L1 norm minimization with mathematical proofs. Experimental comparisons on both artificial and benchmark data indicate that, our proposed L1kPC is more robust, comparable or even better cluster accuracies, less training time than that of state-of-the-art plane clustering methods.

APPENDIX

PROOFS OF DEFINITION 3 AND THEOREM 2 AND 3

Definition 3: A set V is called affine set, if a group of vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_d \in V$ and satisfy $\sum_{i=1}^{d} \theta_i = 1$, then the linear combination $\boldsymbol{y} = \sum_i \theta_i \boldsymbol{x}_i$ also belongs to V.

Proof: Step1. If $d = 2$, $\theta_1 + \theta_2 = 1$, according to the definition 3, then $\boldsymbol{y} = \theta_1\boldsymbol{x}_1 + \theta_2\boldsymbol{x}_2 = \theta_1\boldsymbol{x}_1 + (1-\theta_1)\boldsymbol{x}_2 \in V$ holds. Step2. Assume $d = k-1$ holds. That is, there exists a group of θ_i satisfying $\sum_{i=1}^{d} \theta_i = 1$ and $\theta_i \geq 0$, such that $\boldsymbol{y}_{k-1} = \theta_1\boldsymbol{x}_1 + \theta_2\boldsymbol{x}_2 + \cdots + \theta_{k-1}\boldsymbol{x}_{k-1} \in V$ holds. Step3. If $d = k$, $\sum_{i=1}^{k} \theta_i = 1$ and $\theta_i \geq 0$, then $\theta_1 + \theta_2 + \cdots + \theta_{k-1} = 1 - \theta_k$. When $(1-\theta_k) \neq 0$, $\boldsymbol{y}' = \frac{\theta_1}{1-\theta_k}\boldsymbol{x}_1 + \frac{\theta_2}{1-\theta_k}\boldsymbol{x}_2 + \frac{\theta_2}{1-\theta_k}\boldsymbol{x}_2 + \cdots + \frac{\theta_{k-1}}{1-\theta_k}\boldsymbol{x}_{k-1} \in V$, $\boldsymbol{y}_k = \theta_1\boldsymbol{x}_1 + \theta_2\boldsymbol{x}_2 + \cdots + \theta_{k-1}\boldsymbol{x}_{k-1} + \theta_k\boldsymbol{x}_k = (1-\theta_k)\boldsymbol{y}' + \theta_k\boldsymbol{x}_k$. According to the assumption of *step 2*, then $\boldsymbol{y}_k \in V$. When $1-\theta_k = 0$, such that $\boldsymbol{y}_k = \theta_k\boldsymbol{x}_k \in V$. From the above discussion, the linear combination $\boldsymbol{y} = \theta\boldsymbol{x}_1 + \theta_2\boldsymbol{x}_2 + \cdots + \theta_d\boldsymbol{x}_d \in V$.

Theorem 2: The vector group composed of those d vertexes is a normal orthogonal basis for the d dimensional linear space.

Theorem 3: A convex combination L spanned by d linearly independent L1 hull vertexes is a pseudo subspace, and the total of pseudo subspaces spanned by a normal orthogonal basis of d-dimensional linear space R^d is at most 2^d.

Proof: For convenience of the reader, theorem 2 and 3 are proved together. It is easy for the proof of pseudo subspace, which can be directly derived from Definition 4. Suppose d linearly independent vertexes drawn from the foresaid vertex set $S = \{(\pm 1, 0, \cdots, 0), (0, \pm 1, \cdots, 0), \cdots, (0, 0, \cdots, \pm 1)\}$, $|S| = 2d$, where each vertex corresponds to a unit vector in R^d. Among $|\cdot|$ represents the cardinality of set. Obviously, the group where each vector drawn if and only if from each pair of vertexes is a maximum linearly independent group, and noted as $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_d$. Without loss of generality, let $\boldsymbol{a}_i \in \{(0, \cdots, 1, \cdots, 0)^T, (0, \cdots, -1, \cdots, 0)^T\}$, i.e., the i-th component of the vector $\boldsymbol{a}_i$ does not equal to zero. Thus for any two vectors $\boldsymbol{a}_i, \boldsymbol{a}_j$, $<\boldsymbol{a}_i, \boldsymbol{a}_j>= 1$ when $i = j$; otherwise, $<\boldsymbol{a}_i, \boldsymbol{a}_j>= 0$, where $<\cdot,\cdot>$ denotes the scalar product in the d dimensional linear space R^d. That is, the group $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_d$ is a normal orthogonal basis.

Suppose the linearly independent group composed of k vectors, $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_k$, from corresponding hull vertexes. Denote pseudo subspace $V = \{\lambda_1\boldsymbol{a}_1 + \lambda_2\boldsymbol{a}_2 + \cdots + \lambda_k\boldsymbol{a}_k | \sum\lambda_i = 1, \lambda_i \in [0, 1]\}$. It is obvious that any given point in V is linearly represented by $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_k$. Especially, in d-dimensional linear space, when the pseudo subspace is spanned by d linearly independent hull vertexes, i.e., the foresaid normal orthogonal basis, for any point in this subspace, it is represented by this basis. As foresaid, each basis vector corresponds to a L1 hull vertex. There are d pairs of hull vertexes in L1 convex hull. For the sake of linear independence between basis vectors, each basis vector is drawn from a pair of L1 hull vertexes, while a normal orthogonal basis is composed of d linear independent vectors. Hence, there are 2^d pseudo subspaces respectively determined by 2^d normal orthogonal basis different to each other.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

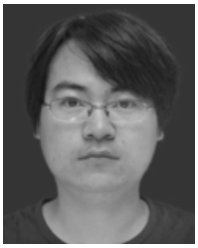
[1] G. W. Milligan and M. C. Cooper, "Methodology review: Clustering methods," *Appl. Psychol. Meas.*, vol. 11, no. 4, pp. 329–354, Dec. 1987.
[2] T. Cutsem and M. Ribbens-Pavella, "Critical survey of hierarchical methods for state estimation of electric power systems," *IEEE Trans. Power App. Syst.*, vols. PAS–102, no. 10, pp. 3415–3424, Oct. 1983.
[3] S. Xu, R. Wang, H. Wang, and H. Zheng, "An optimal hierarchical clustering approach to mobile LiDAR point clouds," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/tits.2019.2912455.
[4] S. Din, A. Ahmad, A. Paul, M. M. Ullah Rathore, and G. Jeon, "A cluster-based data fusion technique to analyze big data in wireless multi-sensor system," *IEEE Access*, vol. 5, pp. 5069–5083, 2017.
[5] J. Amanatides and A. Woo, "A fast voxel traversal algorithm for ray tracing," *Eurographics*, vol. 87, no. 3, pp. 3–10, 1987.
[6] S. Dudoit and J. Fridlyand, "Bagging to improve the accuracy of a clustering procedure," *Bioinformatics*, vol. 19, no. 9, pp. 1090–1099, Jun. 2003.
[7] H. G. Gauch and R. H. Whittaker, "Comparison of ordination techniques," *Ecology*, vol. 53, no. 5, pp. 868–875, Sep. 1972.
[8] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.
[9] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local search heuristics for k-median and facility location problems," *SIAM J. Comput.*, vol. 33, no. 3, pp. 544–562, Jan. 2004.
[10] 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.
[11] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[12] T. J. M. Bench-Capon and P. E. Dunne, "Argumentation in artificial intelligence," *Artif. Intell.*, vol. 171, nos. 10–15, pp. 619–641, Jul. 2007.
[13] C. Town, "Ontological inference for image and video analysis," *Mach. Vis. Appl.*, vol. 17, no. 2, pp. 94–115, May 2006.
[14] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *J. Global Optim.*, vol. 16, no. 1, pp. 23–32, Jan. 2000.
[15] X. Yang, H. Yang, F. Zhang, L. Zhang, X. Fan, Q. Ye, and L. Fu, "Piecewise linear regression based on plane clustering," *IEEE Access*, vol. 7, pp. 29845–29855, 2019.

[16] 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, 2013.
[17] L.-M. Liu, Y.-R. Guo, Z. Wang, Z.-M. Yang, and Y.-H. Shao, "K-Proximal plane clustering," *Int. J. Mach. Learn. Cyber.*, vol. 8, no. 5, pp. 1537–1554, Oct. 2017.
[18] Z. M. Y. R. Yang Guo and C. N. Li, "Local k-proximal plane clustering," *Neural Comput. Appl.*, vol. 26, no. 1, pp. 199–211, Sep. 2015.
[19] K. P. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, 1999, pp. 368–374.
[20] 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.
[21] 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.
[22] 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.
[23] R. Khemchandani, A. Pal, and S. Chandra, "Fuzzy least squares twin support vector clustering," *Neural Computing Appl.*, vol. 29, no. 2, pp. 553–563, Jul. 2018.
[24] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jul. 2006, pp. 2161–2168.
[25] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 471–478.
[26] 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, Sep. 2018.
[27] F. Nie, H. Huang, and X. Cai, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2010, pp. 1813–1821.
[28] 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.
[29] H. Yang, X. Yang, F. Zhang, Q. Ye, and X. Fan, "Infinite norm large margin classifier," *Int. J. Mach. Learn. Cyber.*, vol. 10, no. 9, pp. 2449–2457, Sep. 2019.
[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[31] C. Blake and C. Merz. *UCI Repository of Machine Learning Databases*. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[32] G. W. Milligan and M. C. Cooper, "A study of the comparability of external criteria for hierarchical cluster analysis," *Multivariate Behav. Res.*, vol. 21, no. 4, pp. 441–458, Oct. 1986.
[33] K. H. Borgwardt, "Some distribution-independent results about the asymptotic order of the average number of pivot steps of the simplex method," *Math. Oper. Res.*, vol. 7, no. 3, pp. 441–462, Sep. 1982.
[34] I. E. H. Yen, K. Zhong, and C. J. Hsieh, "Sparse linear programming via primal and dual augmented coordinate descent," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 2368–2376.
[35] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Sarasota, FL, USA, Dec. 2002, pp. 138–142.
[36] [Online]. Available: http://vision.ucsd.edu/ leekc/ExtYaleDatabase/Yale%20Face%20Database.htm
[37] A. Y. Yang, S. S. Sastry, and A. Ganesh, "Fast L1-minimization algorithms and an application in robust face recognition: A review," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Hong Kong, Sep. 2010, pp. 1849–1852.
[38] S. F. Cotter, B. D. Rao, K. Kreutz-Delgado, and J. Adler, "Forward sequential algorithms for best basis selection," *IEE Proc., Vis. Image Process.*, vol. 146, no. 5, p. 235, 1999.
[39] H. Y. Peng, C. F. Jiang, and X. Fang, "Variable selection for Fisher linear discriminant analysis using the modified sequential backward selection algorithm for the microarray data," *Appl. Math. Comput.*, vol. 238, pp. 132–140, Jul. 2014.
[40] S.-O. Abbasi, S. Nejatian, H. Parvin, V. Rezaie, and K. Bagherifard, "Clustering ensemble selection considering quality and diversity," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1311–1340, Aug. 2019.
[41] M. Mojarad, S. Nejatian, H. Parvin, and M. Mohammadpoor, "A fuzzy clustering ensemble based on cluster clustering and iterative Fusion of base clusters," *Appl. Intell.*, vol. 49, no. 7, pp. 2567–2581, Jul. 2019.
[42] A. Bagherinia, B. Minaei-Bidgoli, M. Hossinzadeh, and H. Parvin, "Elite fuzzy clustering ensemble based on clustering diversity and quality measures," *Appl. Intell.*, vol. 49, no. 5, pp. 1724–1747, May 2019.
[43] A. Pirbonyeh, V. Rezaie, H. Parvin, S. Nejatian, and M. Mehrabi, "A linear unsupervised transfer learning by preservation of cluster-and-neighborhood data organization," *Pattern Anal. Appl.*, vol. 22, no. 3, pp. 1149–1160, Aug. 2019.
[44] Y. W. Park and D. Klabjan, "Iteratively reweighted least squares algorithms for L1-norm principal component analysis," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 430–438.
[45] P. P. Markopoulos, M. Dhanaraj, and A. Savakis, "Adaptive L1-norm principal-component analysis with online outlier rejection," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1131–1143, Dec. 2018.
[46] M. Mccoy and J. A. Tropp, "Two proposals for robust PCA using semidefinite programming," *Electron. J. Statist.*, vol. 5, pp. 1123–1160, 2011.
[47] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for L1-subspace signal processing," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 5046–5058, Oct. 2014.
[48] M. C. Tsakiris and R. Vidal, "Dual principal component pursuit," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 10–18.
[49] M. C. Tsakiris and R. Vidal, "Hyperplane clustering via dual principal component pursuit," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3472–3481.
[50] S. Xu and R. Wang, "Power line extraction from mobile LiDAR point clouds," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 2, pp. 734–743, Feb. 2019.
[51] T. Yun, L. Cao, F. An, B. Chen, L. Xue, W. Li, S. Pincebourde, M. J. Smith, and M. P. Eichhorn, "Simulation of multi-platform LiDAR for assessing total leaf area in tree crowns," *Agricult. Forest Meteorol.*, vols. 276–277, Oct. 2019, Art. no. 107610.
[52] J. Yang, L. Zhang, Y. Xu, and J.-Y. Yang, "Beyond sparsity: The role of L1-optimizer in pattern classification," *Pattern Recognit.*, vol. 45, no. 3, pp. 1104–1118, Mar. 2012.
[53] J. Yang, D. Chu, L. Zhang, Y. Xu, and J. Yang, "Sparse representation classifier steered discriminative projection with applications to face recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 7, pp. 1023–1035, Jul. 2013.

HONGXIN YANG received the B.S. degree in computer science and technology from Nanjing Forestry University, in 2017, where she is currently pursuing the master's degree. Her research interests include pattern recognition, machine learning, and image processing.

XUBING YANG received the B.S. degree in mathematics from Anhui University, in 1997, and the M.S. and Ph.D. degrees in computer applications from the Nanjing University of Aeronautics and Astronautics, in 2004 and 2008, respectively. In 2008, he joined Nanjing Forestry University, where he is currently an Associate Professor with the Computer Science and Engineering Department. His research interests include pattern recognition, machine learning, and neural computing. In these fields, he has authored or coauthored over 50 scientific journal articles.

FUQUAN ZHANG received the M.S. degree in computer science from Shenyang Ligong University, in 2005, and the Ph.D. degree from Hanyang University, Seoul, South Korea. His research fields include 3G/4G cellular systems and wireless mesh networks.



QIAOLIN YE (Member, IEEE) received the B.S. degree in computer science from the Nanjing Institute of Technology, Nanjing, China, in 2007, the M.S. degree in computer science and technology from Nanjing Forestry University, Nanjing, in 2009, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Jiangsu, China, in 2013. He is currently an Associate Professor with the Computer Science Department, Nanjing Forestry University. He has authored over 50 scientific articles. Some of them are published in the IEEE TNNLS, IEEE TIFS, and IEEE TCSVT. His research interests include machine learning, data mining, and pattern recognition.