

Robust k -subspace discriminant clustering

Chun-Na Li^a, Yuan-Hai Shao^{a,*}, Yan-Ru Guo^{b,c}, Zhen Wang^d, Zhi-Min Yang^b

^a Management School, Hainan University, Haikou, 570228, PR China

^b Zhijiang College, Zhejiang University of Technology, Hangzhou, 310024, PR China

^c Department of Mathematics, Shanghai University, Shanghai 200444, PR China

^d School of Mathematical Sciences, Inner Mongolia University, Hohhot, 010021, PR China

ARTICLE INFO

Article history:

Received 27 May 2019

Received in revised form 21 August 2019

Accepted 8 October 2019

Available online 22 October 2019

Keywords:

Clustering

k -subspace clustering

Robust clustering

ADMM algorithm

ABSTRACT

k -means, k -plane clustering (kPC) and q -flat are partition-based clustering methods that cluster samples around k centers, hyperplanes and affine subspaces, respectively. However, no discriminant or local information is considered. This paper proposes a robust k -subspace discriminant clustering ($kSDC$) method by introducing the discriminant and local information. $kSDC$ constructs k subspaces with centers to proximate the samples in each subspace, and at the same time, maximizes the difference of the samples in between-cluster. By introducing the L1-norm discriminant and local information to each subspace, $kSDC$ realizes robust dimensionality reduction and clustering, and its optimization problems can be solved through an effective alternating direction method of multipliers. Experimental results on benchmark data and image segmentation data show the effectiveness of our $kSDC$.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Clustering is one of the fundamental machine learning methods that groups data into different clusters on the basis of their similarity, and has been applied widely in many areas, such as text mining [1], gene expression [2], image recognition [3], signal processing [4] and incremental structure from motion [5]. In specific, it has been successfully applied to image segmentation [6–9], where the segmentation of images into homogeneous regions has been an important goal of image-understanding researches [6].

Among all clustering techniques, partition-based clustering methods are fundamental and the most applied ones due to their nice adaptivity and extensibility [10]. The most representative partition-based clustering techniques include k -means [11], k -plane clustering (kPC) [12] and q -flat [13]. Among them, k -means is a classical point-based clustering methods, which clustered samples around k sample centers by minimizing the sum of the squared L2-norm distances from samples to their nearest centers. However, the data may naturally fall into clusters grouped around flat surfaces such as planes. Therefore, kPC used planes rather than points as the entity of the center. Similarly, kPC minimized the sum of the squared distances of samples to their nearest planes. Compared to k -means, the advantages of kPC were that it well separated survival curves and had the ability to cluster data that naturally fall into a subspace. Rather than using sample

centers or hyperplane as cluster centers, k -means and kPC were further extended to q -flat ($0 \leq q \leq n - 1$) [13] which grouped samples around $(n - q)$ -dimensional affine subspaces for arbitrary q , where n was the original feature dimension. Accordingly, k -means and kPC were special cases of q -flat by setting $q = 0$ and $q = n - 1$. As in k -means and kPC , q -flat also minimized the sum of the squared distances of samples to their nearest q -flats. By observing the above descriptions, we see k -means, kPC and q -flat all captured the within-cluster structure well by minimizing some within-cluster distances. However, just minimizing the within-cluster distances has some restriction.

Firstly, they all ignored the influence between different clusters, and hence did not consider the discriminant information. In fact, incorporating discriminant information is a main characteristic in supervised dimensionality reduction. Dimensionality reduction methods consider both the within-class and between-class information. For example, classical linear discriminant analysis (LDA) [14] minimized the squared L2-norm within-class distance while maximized the squared L2-norm between-class distance in the projected space. In the clustering case, k -proximal plane clustering ($kPPC$) [15] improved kPC by taking into account the between-cluster information, and showed a big improvement on kPC . Recently, a novel twin support vector clustering (TWSVC) [16] and its variants [17–19] were studied as an extension of twin support vector machine (TWSVM) to clustering, which also considered discriminant information. Secondly, the planes or flats constructed by kPC , $kPPC$, TWSVC or q -flat may extend infinitely and hence will affect clustering performance. To address this problem, local information is needed for centers. For $kPPC$, local k -proximal plane clustering ($LkPPC$) [20]

* Corresponding author.

E-mail address: shaoyuanhai21@163.com (Y.-H. Shao).

brought the idea of k -means into k PPC, and the local information used in Lk PPC enhanced the performance of k PPC. Lastly, all the above methods were constructed under the squared L2-norm. As we know, compared to the squared L2-norm, the L1-norm is more robust to samples at the category boundary or outliers. In fact, the robustness of L1-norm has been stressed on many machine learning problems [21–24]. In specific, an L1-norm distance minimization-based robust TWSVC (RTWSVC) [25] recently proposed as a robust improvement of TWSVC.

Therefore, it is necessary to design a robust partition-based clustering method that considers both discriminant and local information. Recently, [26] proposed an L1-norm based robust linear discriminant analysis criterion via the Bhattacharyya error bound estimation (L1BLDA). L1BLDA maximized the between-class scatters which were measured by the weighted pairwise distances of class means and meanwhile minimized the within-class scatters under the L1-norm. The weighting constant of L1BLDA between the between-class and within-class terms was determined by the involved data, which made L1BLDA adaptive and computationally efficient. The experimental results demonstrated its effectiveness [26]. Inspired by the spirit of L1BLDA, we manage to achieve within-cluster similarity minimization and between-cluster dissimilarity maximization, and further cluster data into k subspaces by using the advantages of L1BLDA, we propose a robust L1-norm based k -subspace clustering to improve the q -flat type clustering methods, where both discriminant and local information are considered. In summary, a robust k -subspace based clustering method (k SDC) is proposed in this paper, and k SDC has the following characteristics:

- (i) k SDC constructs k subspaces by minimizing the distances from the samples within a subspace to their clustering center, at the same time maximizing between-cluster distances as much as possible, and clusters samples into these k subspaces directly.
- (ii) Different from q -flat, discriminant information and the local centers are embedded in k SDC.
- (iii) The application of the L1-norm criterion in the proposed optimization problem brings robustness of k SDC to outliers.
- (iv) k SDC realizes dimensionality and clustering simultaneously.
- (v) Experimental results on benchmark data sets and image segmentation data sets show the effectiveness of k SDC.

The paper is organized as follows. Section 2 gives a brief review of k PC, k PPC and q -flat. Section 3 presents our k SDC. The experimental results and the conclusion are arranged in Sections 4 and 5, respectively.

2. Backgrounds

Given a set of data samples in multidimensional space, the goal of clustering is to compute a partition of these samples into sets named clusters, such that the samples in the same cluster are more similar than samples across different clusters. Clustering can be divided into two fundamental steps. The first one is the choice of a proximity measure, which is used to measure the similarity of two samples. The second one is the choice of cluster-building algorithm, which is used to assign samples to clusters on the basis of the above chosen proximity measure. Formally, assume the data set $T = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ contains N unlabeled samples, and the i th sample $\mathbf{x}_i$ is an n -dimensional column vector in $\mathbb{R}^n$. Then clustering aims to partition the data set T into k disjoint clusters $\{C_i | i = 1, 2, \dots, k\}$ satisfying $C_i \cap C_{i' \neq i} = \emptyset$ and $T = \bigcup_{i=1}^k C_i$. Correspondingly, $\lambda_l \in \{1, 2, \dots, k\}$ can be used to indicate the cluster label of the sample $\mathbf{x}_l$, $l = 1, 2, \dots, N$.

In this paper, we consider the above clustering problem for a given k . In the whole paper, vectors are referred as column vectors, and all the vectors and matrices are written in upright

bold figures. $\mathbf{I}$ is the identity matrix of an appropriate dimension. The superscript $(\cdot)^T$ denotes the transposition, $\|\cdot\|$ and $\|\cdot\|_1$ are the L2-norm and L1-norm of a vector, and $\|\cdot\|_F$ means the F -norm of a matrix. Assume after a round of labels assignment, these N samples belong to k clusters with their corresponding labels in $\{1, 2, \dots, k\}$, and the i th cluster contains N_i samples. We then organize the data samples with label i into a matrix $\mathbf{X}_i \in \mathbb{R}^{n \times N_i}$, and the samples with the other labels into a matrix $\hat{\mathbf{X}}_i \in \mathbb{R}^{n \times (N - N_i)}$. Let $\bar{\mathbf{x}} = \frac{1}{N} \sum_{l=1}^N \mathbf{x}_l$ be the mean of all samples and $\bar{\mathbf{x}}_i$ be the mean of samples in the i th cluster.

The goal of partition-based clustering methods is to find k cluster centers that are of the forms of points, hyperplanes or affine subspaces, etc., such that the samples with the same label are around their nearest cluster center. In the following, we briefly review three representative partition-based clustering methods, including k PC, k PPC and q -flat.

2.1. k PC

k PC [12] clusters the data into k clusters such that the samples in the i th cluster are around the following center clustering hyperplane

$$f_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + \mathbf{b}_i, \quad (1)$$

which is obtained from

$$\begin{aligned} \min_{\mathbf{w}_i, \mathbf{b}_i} & \|\mathbf{w}_i^T \mathbf{X}_i + \mathbf{b}_i \mathbf{e}_i\|^2 \\ \text{s.t.} & \|\mathbf{w}_i\|^2 = 1, \end{aligned} \quad (2)$$

where $\mathbf{w}_i \in \mathbb{R}^n$, $\mathbf{b}_i \in \mathbb{R}$, $\mathbf{e}_i$ is the vector of ones of an appropriate dimension, $i = 1, 2, \dots, k$, and $\|\cdot\|$ is the L2-norm.

Obviously, k hyperplanes of k PC are determined by minimizing the sum of squares of distances of each sample to its nearest hyperplane $f_i(\mathbf{x})$, which makes all samples distributed round some hyperplane. Model (2) can be solved through an eigenvalue problem. k PC starts from a random initial assignment of samples, and relabels a sample $\mathbf{x}$ by

$$\text{Cluster}(\mathbf{x}) = \arg \min_{i=1,2,\dots,k} \|\mathbf{w}_i^T \mathbf{x} + \mathbf{b}_i\|, \quad (3)$$

after obtaining all $f_i(\mathbf{x})$.

2.2. k PPC

Compared to k PC, k PPC [15] not only requests the samples in each cluster as close as possible to their own center hyperplane, but also pushes the samples in the other clusters far away from this center hyperplane. k PPC formulates as

$$\begin{aligned} \min_{\mathbf{w}_i, \mathbf{b}_i} & \|\mathbf{w}_i^T \mathbf{X}_i + \mathbf{b}_i \mathbf{e}_i\|^2 - C \|\mathbf{w}_i^T \hat{\mathbf{X}}_i + \mathbf{b}_i \hat{\mathbf{e}}_i\|^2 \\ \text{s.t.} & \|\mathbf{w}_i\|^2 = 1, \end{aligned} \quad (4)$$

where C is a positive parameter, $\hat{\mathbf{e}}_i$ the vector of ones of an appropriate dimension as $\mathbf{e}_i$.

Similar to k PC, model (4) is solved through an eigenvalue problem. A Laplacian graph-based initialization is constructed in k PPC instead of random initialization in k PC, which makes k PPC more stable than k PC.

2.3. q -flat

q -flat [13] aims to partition the data set T into k sets, each of which is well approximated by its best fit $(n - q)$ -dimensional subspace. Formally, given k and $q \leq n$, q -flat minimizes the following problem

$$\begin{aligned} \min_{\mathbf{w}_i, \gamma_i} & \|\mathbf{W}_i^T \mathbf{X}_i - \gamma_i \mathbf{e}_i^T\|^2 \\ \text{s.t.} & \mathbf{W}_i^T \mathbf{W}_i = \mathbf{I}, \end{aligned} \quad (5)$$

Algorithm 1 *k*SDC Algorithm.**Input:** Data set $T = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$; cluster number k .**Output:** The final cluster labels of samples in T .**Process:**1. Set the iteration number $r = 0$ and initialize cluster labels of samples in T .2. **Repeat:**(a) **Subspace update:** Solve model (7) with current label assignment;(b) **Assignment update:** Assign each sample according to Eq. (8);**until** $\mathbf{W}_i^{r+1} = \mathbf{W}_i^r$ for all $i = 1, 2, \dots, k$, or reaching maximum iteration number.

where $\mathbf{W}_i \in \mathbb{R}^{n \times q}$, $d \leq n$, and $\mathbf{y}_i \in \mathbb{R}^q$, $i = 1, 2, \dots, k$, $\mathbf{e}_i \in \mathbb{R}^N$ is the vector of ones. q -flat also starts from a random initial assignment of samples, and relabels samples by

$$\text{Cluster}(\mathbf{x}) = \arg \min_{i=1,2,\dots,k} \|\mathbf{W}_i^T \mathbf{x} - \mathbf{y}_i\|, \quad (6)$$

after obtaining all $\mathbf{W}_i$ and $\mathbf{y}_i$.

3. Robust *k*-subspace discriminant clustering**3.1. *k*-subspace discriminant clustering**

The proposed *k*SDC constructs k subspaces generated by $\mathbf{W}_i$, $i = 1, \dots, k$, such that the samples in each cluster are closer to its center in the corresponding subspace, and at the same time, far from the other samples. In specific, our *k*SDC first initializes the cluster assignment, and computes the k subspaces. For the i th cluster, $i = 1, \dots, k$, we solve the following optimization problem

$$\min_{\mathbf{W}_i} \Omega \sum_{s=1}^{N_i} \|\mathbf{W}_i^T (\mathbf{x}_s^i - \bar{\mathbf{x}}_i)\|_1 - \frac{1}{N} \sum_{i \neq j} \sqrt{N_i N_j} \|\mathbf{W}_i^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)\|_1 \quad (7)$$

s.t. $\mathbf{W}_i^T \mathbf{W}_i = \mathbf{I}$,

where $\mathbf{W}_i \in \mathbb{R}^{n \times d}$ is the projection matrix for the i th subspace, $d \leq n$, $\Omega = \frac{\sqrt{n}}{4N} \sum_{i \neq j}^c \sqrt{N_i N_j} \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_1$ is a weighting constant.

After getting the solution of model (7) for each i , each sample $\mathbf{x}$ is relabeled by

$$\text{Cluster}(\mathbf{x}) = \arg \min_{i=1,2,\dots,k} \|\mathbf{W}_i^T (\mathbf{x} - \bar{\mathbf{x}}_i)\|, \quad (8)$$

and k clusters are updated correspondingly. These updated clusters are used to identify new projection directions by model (7). The whole process continues until there is no new update.

We now give the geometric meaning of model (7). By minimizing the first term in model (7), *k*SDC makes the samples in the i th cluster as close as possible to its own cluster center in its subspace. By minimizing the second term in model (7), the weighted sum of the distances between the cluster center of the i th cluster and the cluster centers not in the i th cluster are requested to as large as possible, and $\frac{1}{N} \sqrt{N_i N_j}$ is the weight between the i th and the j th clusters. The constraint $\mathbf{W}_i^T \mathbf{W}_i = \mathbf{I}$ makes the data samples in the i th cluster be projected to a space that is generated by orthonormal vectors, ensuring the minimum redundancy in the projected space. This idea is similar to that of *k*PPC, but with local center information. This criterion is similar to that of LDA [26,27] for classification, while we introduce it to clustering.

3.2. The solving algorithm of *k*SDC

With the initial cluster labels of all samples in T , *k*SDC updates k clustering subspaces and the sample labels alternatively. Precisely, the clustering procedure of *k*SDC can be realized through Algorithm 1.

Clearly, given a cluster assignment, the essential work of Algorithm 1 is to solve model (7). In the following we apply an alternating direction method of multipliers (ADMM) [28] algorithm to solve it. We first transfer model (7) to its corresponding ADMM form as

$$\begin{aligned} \min_{\mathbf{W}_i, \mathbf{B}_{ij}, \mathbf{Z}_{is}, \mathbf{D}} \quad & \Omega \sum_{s=1}^{N_i} |\mathbf{Z}_{is}| - \sum_{j \neq i} |\mathbf{B}_{ij}| + \ell(\mathbf{W}_i) \\ \text{s.t.} \quad & \frac{1}{N} \sqrt{N_i N_j} \mathbf{W}_i^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) - \mathbf{B}_{ij} = 0, \quad j \neq i, \\ & \mathbf{W}_i^T (\mathbf{x}_s^i - \bar{\mathbf{x}}_i) - \mathbf{Z}_{is} = 0, \\ & \mathbf{D} - \mathbf{W}_i = 0, \\ & i = 1, \dots, k, \quad j = 1, \dots, N_i, \end{aligned} \quad (9)$$

where $\mathbf{B}_{ij}, \mathbf{Z}_{is} \in \mathbb{R}^d$, $\mathbf{D}, \mathbf{W}_i \in \mathbb{R}^{n \times d}$ are variables, and $\ell(\cdot)$ is a function defined by $\ell(\mathbf{W}_i) = \begin{cases} 0, & \mathbf{W}_i^T \mathbf{W}_i = \mathbf{I} \\ +\infty, & \text{otherwise.} \end{cases}$

Then the augmented Lagrangian of model (9) is given by

$$\begin{aligned} & L_\rho(\mathbf{W}_i, \mathbf{B}_{ij}, \mathbf{Z}_{is}, \mathbf{D}; \alpha_{ij}, \beta_{ij}, \Gamma) \\ &= \Omega \sum_{j=1}^{N_i} |\mathbf{Z}_{is}| - \sum_{j \neq i} |\mathbf{B}_{ij}| + \ell(\mathbf{W}_i) \\ &+ \sum_{j \neq i}^k \alpha_{ij}^T \left(\frac{1}{N} \sqrt{N_i N_j} \mathbf{W}_i^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) - \mathbf{B}_{ij} \right) \\ &+ \sum_{j=1}^{N_i} \beta_{ij}^T (\mathbf{W}_i^T (\mathbf{x}_s^i - \bar{\mathbf{x}}_i) - \mathbf{Z}_{is}) + \Gamma^T (\mathbf{D} - \mathbf{W}_i) \\ &+ \frac{\rho}{2} \sum_{j \neq i}^k \left\| \frac{1}{N} \sqrt{N_i N_j} \mathbf{W}_i^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) - \mathbf{B}_{ij} \right\|^2 \\ &+ \frac{\rho}{2} \sum_{j=1}^{N_i} \left\| \mathbf{W}_i^T (\mathbf{x}_s^i - \bar{\mathbf{x}}_i) - \mathbf{Z}_{is} \right\|^2 + \frac{\rho}{2} \|\mathbf{D} - \mathbf{W}_i\|_F^2, \end{aligned} \quad (10)$$

where $\rho > 0$ is the penalty parameter, and $\alpha_{ij} \in \mathbb{R}^d$, $\beta_{ij} \in \mathbb{R}^d$, $\Gamma \in \mathbb{R}^{n \times d}$ are dual variables of model (9).

We now iteratively solve model (9) or equivalently model (7) through Eq. (10). Firstly, set the iteration number $t = 0$ and initialize $\mathbf{D}^0 \in \mathbb{R}^{n \times d}$, $\mathbf{B}_{ij}^0 \in \mathbb{R}^d$, $\mathbf{Z}_{is}^0 \in \mathbb{R}^d$ and $\alpha_{ij}^0 \in \mathbb{R}^d$, $\beta_{ij}^0 \in \mathbb{R}^d$, $\Gamma^0 \in \mathbb{R}^{n \times d}$ for $i = 1, \dots, k$, $j = 1, \dots, N_i$.

Step (a): By fixing other variables in the t th iteration, we first solve $\mathbf{W}_i^{t+1}$ by

$$\mathbf{W}_i^{t+1} = \arg \min_{\mathbf{W}_i} \ell(\mathbf{W}_i) + \frac{\rho}{2} \text{tr}(\mathbf{W}_i^T \mathbf{G} \mathbf{W}_i) + \rho \cdot \text{tr}((\mathbf{A}^t)^T \mathbf{W}_i), \quad (11)$$

where $\mathbf{G}_i = \sum_{j \neq i}^k \frac{N_i N_j}{N^2} (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)^T + \sum_{s=1}^{N_i} (\mathbf{x}_s^i - \bar{\mathbf{x}}_i)(\mathbf{x}_s^i - \bar{\mathbf{x}}_i)^T + \mathbf{I} \in \mathbb{R}^{n \times n}$, and $\mathbf{A}^t = \sum_{j \neq i}^k \frac{1}{N} \sqrt{N_i N_j} (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)(\mathbf{B}_{ij}^t - \alpha_{ij}^t)^T + \sum_{j=1}^{N_i} (\mathbf{x}_s^i - \bar{\mathbf{x}}_i)(\mathbf{Z}_{is}^t -$

$\beta_{ij}^t)^T + (\mathbf{D}^t + \mathbf{I}^t) \in \mathbb{R}^{n \times d}$. Then model (11) is equivalent to

$$\begin{aligned} \arg \min_{\mathbf{W}_i} & \text{tr}(\mathbf{W}_i^T \mathbf{G}_i \mathbf{W}_i) + 2 \cdot \text{tr}((\mathbf{A}^t)^T \mathbf{W}_i) \\ \text{s.t. } & \mathbf{W}_i^T \mathbf{W}_i = \mathbf{I}. \end{aligned} \quad (12)$$

We solve model (12) by two cases.

Case 1: $d < n$. In this situation, model (12) is an unbalanced Procrustes problem [29], and there is no analytic solution. We here adopt a recently proposed convergent algorithm in Ref. [30] to solve model (12) as shown in Algorithm 2.

Algorithm 2 Algorithm for model (12) when $d < n$.

- (a) Compute the dominant eigenvalue a of $\mathbf{G}_i$.
- (b) Random initialize $\mathbf{W}_i \in \mathbb{R}^{n \times d}$ such that $\mathbf{W}_i^T \mathbf{W}_i = \mathbf{I}$.
- (c) Update $\mathbf{M} \leftarrow 2(a\mathbf{I} - \mathbf{G}_i)\mathbf{W}_i - 2\mathbf{A}^t$.
- (d) Calculate $\mathbf{W}_i$ by solving the following problem

$$\max_{\mathbf{W}_i^T \mathbf{W}_i = \mathbf{I}} \text{tr}(\mathbf{W}_i^T \mathbf{M}).$$
- (e) Iteratively perform the above steps (c) and (d) until convergence.

Case 2: $d = n$. Since $\mathbf{G}_i$ is positive definite, we can further write $\mathbf{G}_i = \mathbf{G}_{i0}(\mathbf{G}_{i0})^T$ by Cholesky decomposition, where $\mathbf{G}_{i0}$ is an invertible lower triangular matrix. Then model (11) is equivalent to

$$\begin{aligned} \arg \min_{\mathbf{W}_i} & \frac{\rho}{2} \|\mathbf{G}_{i0}^T \mathbf{W}_i - (\mathbf{G}_0)^{-1} \mathbf{A}^t\|_F^2 \\ \text{s.t. } & \mathbf{W}_i^T \mathbf{W}_i = \mathbf{I}. \end{aligned} \quad (13)$$

In this situation, model (12) is a balanced Procrustes problem [31], and can be solved as

$$\mathbf{W}^{t+1} = \mathbf{P}_1^t (\mathbf{P}_2^t)^T, \quad (14)$$

where $\mathbf{P}_1^t$ and $\mathbf{P}_2^t$ are orthogonal matrices from the following SVD decomposition

$$\mathbf{A}^t = \mathbf{P}_1^t \Sigma^t \mathbf{P}_2^t.$$

Step (b): In the $(t+1)$ -th iteration, we obtain $\mathbf{B}_{ij}^{t+1}$ by solving

$$\begin{aligned} \mathbf{B}_{ij}^{t+1} = \arg \min_{\mathbf{B}_{ij}} & - \sum_{j \neq i}^k |\mathbf{B}_{ij}| \\ & + \frac{\rho}{2} \sum_{j \neq i}^k \left\| \frac{1}{N} \sqrt{N_i N_j} (\mathbf{W}_i^{t+1})^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) - \mathbf{B}_{ij} + \alpha_{ij}^t \right\|^2. \end{aligned} \quad (15)$$

By direct computation, its solution is given as

$$\mathbf{B}_{ij}^{t+1} = \begin{cases} \frac{1}{N} \sqrt{N_i N_j} (\mathbf{W}_i^{t+1})^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) + \alpha_{ij}^t + 1/\rho, & \text{if } \frac{1}{N} \sqrt{N_i N_j} (\mathbf{W}_i^{t+1})^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) + \alpha_{ij}^t \geq 0, \\ \frac{1}{N} \sqrt{N_i N_j} (\mathbf{W}_i^{t+1})^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) + \alpha_{ij}^t - 1/\rho, & \text{if } \frac{1}{N} \sqrt{N_i N_j} (\mathbf{W}_i^{t+1})^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) + \alpha_{ij}^t < 0, \end{cases} \quad (16)$$

for $i = 1, 2, \dots, k$

Step (c): In the $(t+1)$ -th iteration, $\mathbf{Z}_{is}^{t+1}$, $s = 1, 2, \dots, N_i$, is solved by

$$\begin{aligned} \mathbf{Z}_{is}^{t+1} = \arg \min_{\mathbf{Z}_{is}} & \Omega \sum_{j=1}^{N_i} |\mathbf{Z}_{is}| \\ & + \frac{\rho}{2} \sum_{j=1}^{N_i} \|(\mathbf{W}_i^{t+1})^T (\mathbf{x}_s^i - \bar{\mathbf{x}}_i) - \mathbf{Z}_{is} + \beta_{ij}^t\|^2. \end{aligned}$$

Its solution is given through soft thresholding function

$$\mathbf{Z}_{is}^{t+1} = \Phi_{\Omega/\rho}[(\mathbf{W}_i^{t+1})^T (\mathbf{x}_s^i - \bar{\mathbf{x}}_i) + \beta_{ij}^t] \quad (17)$$

for $i = 1, 2, \dots, k$ and $j = 1, 2, \dots, N_i$, where

$$\Phi_{\kappa}(a) = \begin{cases} a - \kappa, & a > \kappa \\ 0, & |a| \leq \kappa \\ a + \kappa, & a < -\kappa. \end{cases}$$

Step (d): In the $(t+1)$ -th iteration, $\mathbf{D}^{t+1}$ is then solved by

$$\mathbf{D}^{t+1} = \arg \min_{\mathbf{D}} \frac{\rho}{2} \|\mathbf{D} - \mathbf{W}_i^{t+1} + \mathbf{I}^t\|_F^2.$$

Similarly, $\mathbf{D}$ can be given componentwisely by

$$\mathbf{D}^{t+1} = \mathbf{W}_i^{t+1} - \mathbf{I}^t. \quad (18)$$

Step (a)–Step (d) give the solution of primal variables. The dual variables can be easily updated by

$$\alpha_{ij}^{t+1} = \alpha_{ij}^t + \left(\frac{1}{N} \sqrt{N_i N_j} (\mathbf{W}_i^{t+1})^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) - \mathbf{B}_{ij}^{t+1} \right), \quad (19)$$

$$\beta_{ij}^{t+1} = \beta_{ij}^t + ((\mathbf{W}_i^{t+1})^T (\mathbf{x}_s^i - \bar{\mathbf{x}}_i) - \mathbf{Z}_{is}^{t+1}), \quad (20)$$

$$\mathbf{I}^{t+1} = \mathbf{I}^t + (\mathbf{D}^{t+1} - \mathbf{W}_i^{t+1}). \quad (21)$$

The above iteration procedure continues until some termination conditions are satisfied [26]. We summarize the above ADMM algorithm for solving model (7) with termination conditions in Algorithm 3.

Algorithm 3 ADMM algorithm for solving (7).

Input: Data set $T = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$; the positive tolerance parameters ϵ^{pri} and ϵ^{dual} .

Output: $\mathbf{W}_i^* = \mathbf{W}_i^{t+1}$, where $\mathbf{W}_i^*$ is the optimal solution of (7).

Process:

1. Set the iteration number $t = 0$ and initialize $\mathbf{D}^0 \in \mathbb{R}^{n \times d}$, $\mathbf{B}_{ij}^0 \in \mathbb{R}^d$, $\mathbf{Z}_{is}^0 \in \mathbb{R}^d$ and $\alpha_{ij}^0 \in \mathbb{R}^d$, $\beta_{ij}^0 \in \mathbb{R}^d$, $\mathbf{I}^0 \in \mathbb{R}^{n \times d}$ for $i = 1, \dots, k$, $j = 1, \dots, N_i$.
2. **for** $t = 1, 2, \dots$

- (a) Compute $\mathbf{W}_i^{t+1}$ according to $\begin{cases} \text{Algorithm 2,} & d < n \\ \text{Eq. (14),} & d = n; \end{cases}$

- (b) Compute $\mathbf{B}_{ij}^{t+1}$ according to Eq. (16);

- (c) Compute $\mathbf{Z}_{is}^{t+1}$ according to Eq. (17);

- (d) Compute $\mathbf{D}^{t+1}$ according to Eq. (18);

- (e) Update α_{ij}^{t+1} according to Eq. (19);

- (f) Update β_{ij}^{t+1} according to Eq. (20);

- (g) Update $\mathbf{I}^{t+1}$ according to Eq. (21);

until the primal residual

$$\|r^{t+1}\| = \max_{i,j} \left\{ \left\| \frac{1}{N} \sqrt{N_i N_j} (\mathbf{W}_i^{t+1})^T (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) - \mathbf{B}_{ij}^{t+1} \right\|, \right.$$

$$\left. \|(\mathbf{W}_i^{t+1})^T (\mathbf{x}_s^i - \bar{\mathbf{x}}_i) - \mathbf{Z}_{is}^{t+1}\|, \|\mathbf{W}_i^{t+1} - \mathbf{D}^{t+1}\|_F \right\} \leq \epsilon^{pri}$$

and the dual residual

$$\|s^{t+1}\| = \max_{i,j} \left\{ \|\rho(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)(\mathbf{B}_{ij}^{t+1} - \mathbf{B}_{ij}^t)\|, \right.$$

$$\left. \|\rho(\mathbf{x}_s^i - \bar{\mathbf{x}}_i)(\mathbf{Z}_{is}^{t+1} - \mathbf{Z}_{is}^t)\|, \|\rho(\mathbf{D}^{t+1} - \mathbf{D}^t)\|_F \right\} \leq \epsilon^{dual}.$$

end

4. Experimental results

In this section, we analyze the performance of our k SDC compared with some related partition-based clustering methods, including k -means [11], k PC [12], q -flat [13], k PPC [15], TWSVC [16], RTWSVC [32] and FRTWSVC [32] on eighteen benchmark data sets, as well as three segmentation images. All our experiments

Table 1

Accuracies (%) and computation time (second) of clustering methods on the benchmark data sets.

Data set Sample no.×Feature no.	k-means AC/Time	kPC AC/Time	q-flat AC/Time	kPPC AC/Time	TWSVC AC/Time	RTWSVC AC/Time	FRTWSVC AC/Time	kSDC AC/Time
Compound	86.29	72.04	75.82 (2)	73.90	81.66	80.44	82.52	89.54 (2)
399 × 2	0.0034	1.3673	0.2993	1.3107	5.4263	21.3049	0.9545	0.2761
Aggregation	91.91	79.19	92.59 (2)	79.00	83.30	82.82	84.10	92.97 (2)
788 × 2	0.0055	28.4346	0.0111	28.1013	58.6749	993.2816	10.3082	0.0979
R15	98.21	92.15	98.44 (2)	92.00	96.53	93.07	92.91	98.72 (2)
600 × 2	0.0037	8.0559	0.0249	7.9376	39.5611	49.5294	19.8812	0.1067
Flame	72.67	69.93	69.93 (1)	71.55	72.67	69.93	70.46	72.67 (2)
240 × 2	0.0023	1.8229	0.0547	0.0916	0.3476	0.0283	3.4750	0.0185
Arrhythmia	65.72	32.31	64.52 (1)	65.20	32.31	32.31	32.31	65.86 (28)
452 × 278	0.0541	2.2077	0.2993	6.8241	4.7036	0.2504	0.1955	0.0222
Dermatology	69.76	60.50	88.56 (1)	70.36	71.55	60.50	60.50	81.89 (20)
366 × 34	0.0034	1.0116	0.0428	1.0744	7.8135	0.0537	0.0174	0.0570
Ecoli	82.19	33.11	83.75 (7)	66.46	77.26	34.33	34.33	83.75 (7)
336 × 7	0.0036	0.7731	0.0129	0.7643	16.7281	0.0197	0.0097	0.1345
Seeds	87.35	71.80	72.09 (6)	62.39	75.89	72.24	76.16	86.81 (7)
210 × 7	0.0022	0.2120	0.0030	0.2200	0.4705	5.0752	0.1465	0.0432
Zoo	87.49	54.12	90.26 (1)	84.06	88.69	54.12	54.12	89.94 (10)
101 × 16	0.0020	0.0371	0.0036	0.0615	0.0617	0.0027	0.0028	0.0976
Haberman	49.91	49.84	60.64 (3)	60.95	62.87	61.57	62.21	63.90 (2)
306 × 3	0.0021	0.5886	0.0019	0.5828	1.1871	1.1813	0.0330	0.0583
Housevotes	78.90	78.83	75.17 (14)	68.77	75.83	71.40	71.40	78.90 (16)
435 × 16	0.0026	1.7025	0.0032	1.6766	2.3004	0.0212	0.0085	0.0782
Musk	50.12	50.15	50.98 (1)	50.18	51.31	51.04	50.28	53.63 (7)
476 × 166	0.0041	2.5847	0.0380	2.6998	3.1540	1.7669	4.2137	0.1125
Sonar	50.22	49.80	51.44 (1)	49.99	49.93	51.26	50.06	51.81 (27)
208 × 60	0.0030	0.2132	0.0119	0.2458	0.2434	29.5494	0.5248	0.1072
Spectf	53.95	49.49	49.49 (1)	50.51	50.16	51.93	51.39	51.93 (8)
80 × 44	0.0016	0.0334	0.0011	0.0503	0.0390	0.3785	0.0319	0.0317
Wpbc	56.03	52.95	63.61 (29)	57.95	56.39	53.48	57.15	64.15 (13)
198 × 34	0.0015	0.1816	0.0013	0.2304	0.2185	0.1742	0.5075	0.0222
German	55.80	55.80	58.61 (20)	57.96	56.08	57.96	55.80	58.04 (8)
1000 × 20	0.0023	15.8678	0.0162	78.6714	104.9822	2.2655	0.6696	0.1153
Hepatitis	62.77	55.56	67.02 (3)	71.90	66.27	67.02	73.66	67.79 (15)
155 × 19	0.0221	0.0010	0.0273	0.5869	0.2202	0.0456	0.0108	0.0208
Australian	50.66	49.93	52.23 (2)	50.53	60.97	50.00	49.93	52.62 (12)
690 × 14	0.0129	0.0012	0.0389	27.5401	63.0056	1.0531	0.1325	0.0468

are carried out on a PC machine with Intel 3.30 GHz CPU and 4 GB RAM memory under Matlab 2017b platform. kPC and kPPC obtain their solutions by solving eigenvalue problems, and are realized by using a simple MATLAB function “eig”. q-flat is solved through the singular value decomposition (SVD). TWSVC is solved through two quadratic problems (QPs). Instead of using MATLAB function “quadprog”, to speed up the learning speed, we use successive overrelaxation (SOR) technique to obtain the solution of TWSVC. RTWSVC solves a series of QPs by using SOR, and FRTWSVC solves a series of systems of linear equations by simply using the division operation “/”. With regard to parameter selection, the trade-off parameter c in k-PC, PPC, TWSVC, RTWSVC and FRTWSVC, and kernel parameter μ for all the methods, are selected from the set $\{2^{-8}, 2^{-7}, \dots, 2^7\}$. The random initialization is fused for kmeans, and the NNG [33] initialization technique is used for other methods, and its parameter p is selected from $\{1, 2, 3, 4, 5\}$. The optimal parameter was selected by using the grid searching technique for all the investigated methods. Once the optimal parameters are selected, they are employed to learn the final clusters.

Since clustering accuracy (AC) [34] is the most recognized measurement for labeled data, and Dice Ratio (DR) is the most applied measurement for unlabeled data, in our experiments, to measure the performance of these methods, we use AC on benchmark data sets and DR on segmentation images as measurement. Define TP (true positive) as the number of similar sample classified into the same cluster, TN (true negative) as the number of dissimilar sample classified into different clusters, FP (false positive) as the number of dissimilar sample classified into the same cluster, and FN (false negative) as the number of similar

sample classified into different clusters. Then AC is defined as

$$AC = \frac{TP + TN}{TP + FP + TN + FN}.$$

If $S1$ and $S2$ are two segmentation results of the same image, then DR is defined as

$$DR = \frac{2|S1 \cap S2|}{|S1| + |S2|}.$$

4.1. Benchmark data sets

4.1.1. Performance analysis

We first evaluate the performance of each method in terms of accuracy on benchmark data sets. For q-flat and our kSDC, the accuracy under optimal reduced dimension is employed. Table 1 records the accuracies for all the clustering methods. The results show that: (i) The proposed kSDC owns the highest accuracies on more data sets than other methods, while on other data sets, kSDC is comparable to q-flat or k-means, which demonstrates the effectiveness of kSDC. To clearly see the superiority of our method, we also compute the rank of each method for each data set, and the corresponding results are presented in Table 2. From the table, we see that on most of the data sets, we have the highest rank; while on most of the rest data sets, we are second the best. We also has the best average rank. The result supports the above argument. (ii) Choosing appropriate reduced dimensions is necessary for both q-flat and our kSDC. To see this phenomenon clearly, we describe the accuracy variation along different dimensions in Fig. 1 on several data sets, including “Compound”,

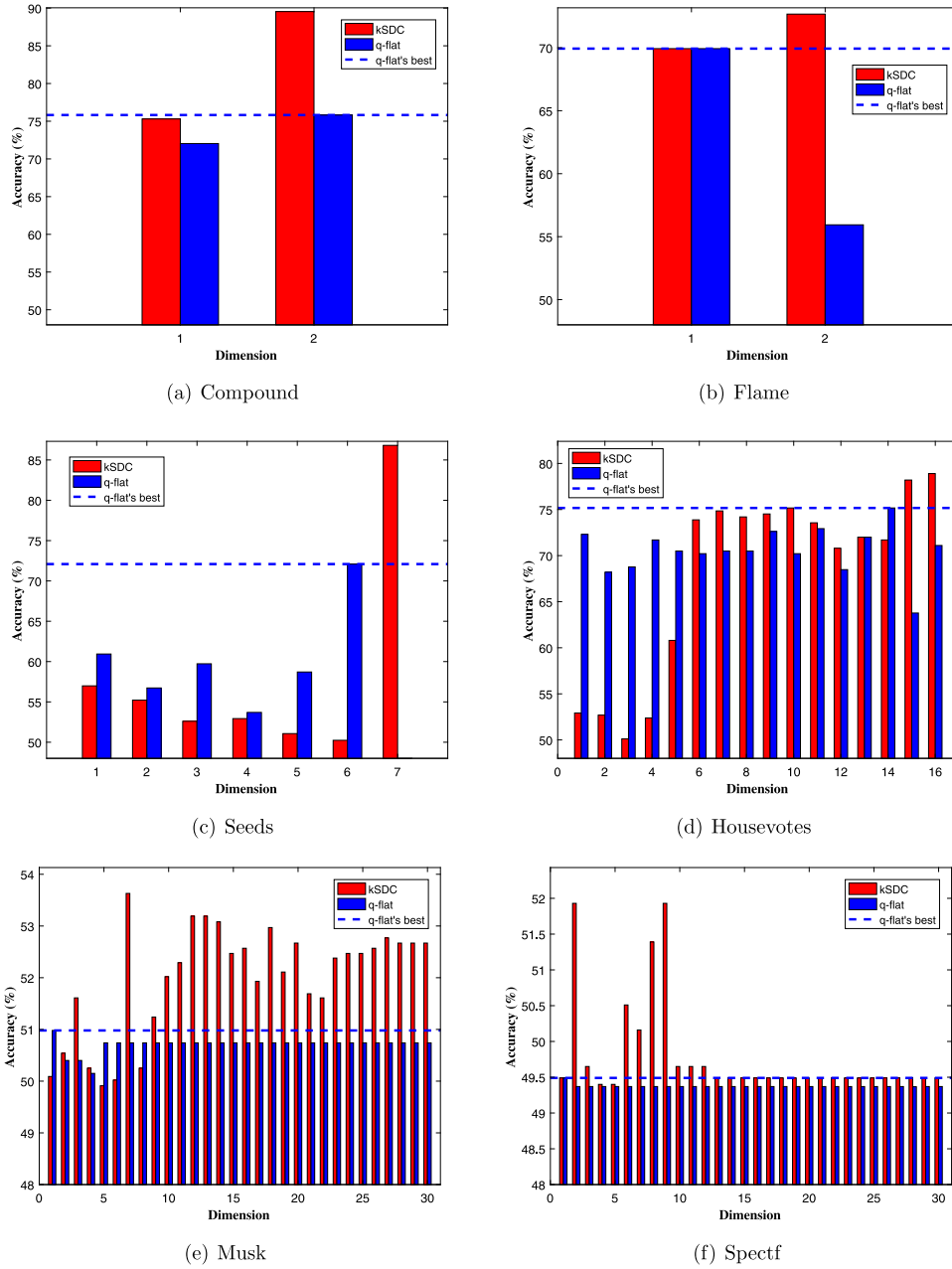


Fig. 1. Accuracies of q -flat and k SDC along different dimensions. The dash line represents the highest accuracy of q -flat under optimal reduced dimension.

“Flame”, “Seeds”, “Housevotes”, “Musk” and “Spectf”. For an n -dimensional data set, we plot the accuracies of q -flat and our k SDC when the reduced dimension is set to $d = 1, 2, \dots, n$, and also highlight the highest accuracy among all the dimensions for each data set. In specific, the dash line in Fig. 1 represents the highest accuracy of q -flat under optimal reduced dimension. From the figure, we see that no matter for q -flat or our k SDC, there is a big difference on accuracies for different reduced dimensions, and the highest accuracy of our k SDC is higher than the highest accuracy of q -flat. The results indicate that it is necessary to choose optimal dimension for different data sets, and our k SDC captures the intrinsic structures of complex data sets well by choosing a subspace of appropriate dimension and performs better than q -flat; (iii) For computation time, k SDC is slower than k -means and is comparable to q -flat, and is much faster than other methods. This shows the efficiency of our proposed method.

4.1.2. Statistical analysis

In order to obtain reliable assessments, we use the Friedman test [35,36] to make further comparison of these eight algorithms. The Friedman test with the corresponding post hoc tests was proven to be a simple, safe, and robust non-parametric test and has been applied in [37]. Therefore, we apply the Friedman test to detect whether the average ranks of all the methods are significantly different from the mean rank under the null hypothesis.

Let r_i^j be the rank of the j th algorithm on the i th data set, where $i = 1, 2, \dots, M$, $j = 1, 2, \dots, L$, M is the number of data sets and L is the number of algorithms. Let $R_j = \frac{1}{M} \sum_{i=1}^M r_i^j$ be the average rank of the j th algorithm on all data sets. Then the Friedman statistic [35]

$$\chi_F^2 = \frac{12M}{L(L+1)} \left[\sum_{j=1}^L R_j^2 - \frac{L(L+1)^2}{4} \right]$$

Table 2
Ranks of clustering methods on the benchmark data sets.

Data set	<i>k</i> -means Rank	<i>k</i> PC Rank	<i>q</i> -flat Rank	<i>k</i> PPC Rank	TWSVC Rank	RTWSVC Rank	FRTWSVC Rank	<i>k</i> SDC Rank
Compound	2	8	6	7	4	5	3	1
Aggregation	3	7	2	8	5	6	4	1
R15	3	7	2	8	4	5	6	1
Flame	2	7	7	4	2	7	5	2
Arrhythmia	2	6.5	4	3	6.5	6.5	6.5	1
Dermatology	5	7	1	4	3	7	7	2
Ecoli	3	8	1.5	5	4	6.5	6.5	1.5
Seeds	1	7	6	8	4	5	3	2
Zoo	4	7	1	5	3	7	7	2
Haberman	7	8	6	5	2	4	3	1
Housevotes	1.5	3	5	8	4	6.5	6.5	1.5
Musk	8	7	4	6	2	3	5	1
Sonar	4	8	2	6	7	3	5	1
Spectf	1	7.5	7.5	5	6	2.5	4	2.5
Wpbc	6	8	2	3	5	7	4	1
German	7	7	1	3.5	5	3.5	7	2
Hepatitis	7	8	4.5	2	6	4.5	1	3
Australian	4	7.5	3	5	1	6	7.5	2
Average rank	3.9167	7.1389	3.6389	5.3056	4.0833	5.2778	5.0556	1.5833

Table 3

Nemenyi test of all the methods on benchmark data sets. Here the difference of average rank is computed as the difference between the average rank of other method and the average rank of our *k*SDC.

Method	10% critical CD value ($\alpha = 0.1$)	Difference of average ranks
<i>k</i> -means	2.2699	2.3333
<i>k</i> PC	2.2699	5.5556
<i>q</i> -flat	2.2699	2.0556
<i>k</i> PPC	2.2699	3.7222
TWSVC	2.2699	2.5000
RTWSVC	2.2699	3.6944
FRTWSVC	2.2699	3.4722
<i>k</i> SDC	2.2699	0.0000

obeys χ_F^2 distribution with $L - 1$ degrees of freedom, and

$$F_F = \frac{(M - 1)\chi_F^2}{M(L - 1) - \chi_F^2}$$

obeys F -distribution with $L - 1$ and $(L - 1)(M - 1)$ degrees of freedom [36]. In our case, we have $M = 18$ and $L = 7$, and it can be computed directly that for ranks of all the methods in Table 2, $F_F = 13.1121$. Note that the critical value of $F(7, 119)$ for significance level $\alpha = 0.1$ is 1.769. Clearly, the null hypothesis proposes that all the algorithms behave similarly is rejected since $F_F = 13.1121 > 1.769$, and hence the eight algorithms are statistically significantly different.

The above rejection requires us perform a posthoc Nemenyi test [38]. The Nemenyi test is used to compare each two algorithms. We regard the performance of two algorithms to be significantly different if their average ranks differ by at least the critical difference that is defined as

$$CD = q_\alpha \sqrt{\frac{L(L + 1)}{6M}},$$

where the calculation of critical value q_α is referred to Ref. [37]. In our case, $q_{0.1} = 2.780$ and $CD = 2.2699$. The results of Nemenyi test are shown in Table 3. It can be seen that the differences of ranks between other methods except *q*-flat and our *k*SDC are larger than CD value at the 10% significance level, and the rank difference between our *k*SDC and *q*-flat is very close to CD value at the 10% significance level. The result shows that on these benchmark data sets, our *k*SDC significantly outperforms *k*-means, *k*PC, *k*PPC, TWSVC, RTWSVC and FRTWSVC, and is comparable to *q*-flat in terms of accuracy, which demonstrates the superiority of our *k*SDC.

4.2. Image segmentation

We here compare the performance of *k*SDC with the other methods on the image segmentation data. All the methods are implemented on the simulated data that is available from Brain-Web simulator repository [39]. For clustering performance comparison, the quantitative index Dice Ratio (DR) is considered:

$$DR = \frac{2|S1 \cap S2|}{|S1| + |S2|}, \quad (22)$$

where $S1$ represents the set of pixels obtained by the algorithm, while $S2$ represents the set of pixels from the real image. The larger the Dice ratio, the better the clustering performance.

We first apply all the methods to a brain MR image of 160×160 pixels and realize segmentation. The original image and the image segmentation results are shown in Fig. 2. From Fig. 2, we observe that *q*-flat and *k*SDC achieve good performance, while *k*SDC shows more reasonable segmentation result. To test the robustness of various methods, we then pollute the above image by adding the Salt & Pepper noise of strength 0.5. The number of

Table 4
Dice ratios (%) of clustering methods on the MR image.

Noise type	<i>k</i> -means DR	<i>k</i> PC DR	<i>q</i> -flat DR	<i>k</i> PPC DR	TWSVC DR	RTWSVC DR	FRTWSVC DR	<i>k</i> SDC DR
Original image	72.65	77.94	80.57	80.20	67.79	53.25	66.48	81.79
Salt & Pepper 0.5	79.16	73.73	82.01	85.49	68.72	66.44	80.94	89.62
Salt & Pepper 3	98.59	73.28	71.71	60.19	96.57	80.50	86.75	98.70
Salt & Pepper 18	82.47	89.53	72.49	64.25	96.44	81.86	69.11	97.30
Gaussian 0.5	90.33	82.44	83.44	64.14	93.81	90.62	88.20	98.57
Gaussian 3	92.49	82.40	72.68	64.22	89.41	82.14	88.74	95.20
Gaussian 18	97.36	71.94	68.65	65.33	88.93	83.77	83.40	95.34

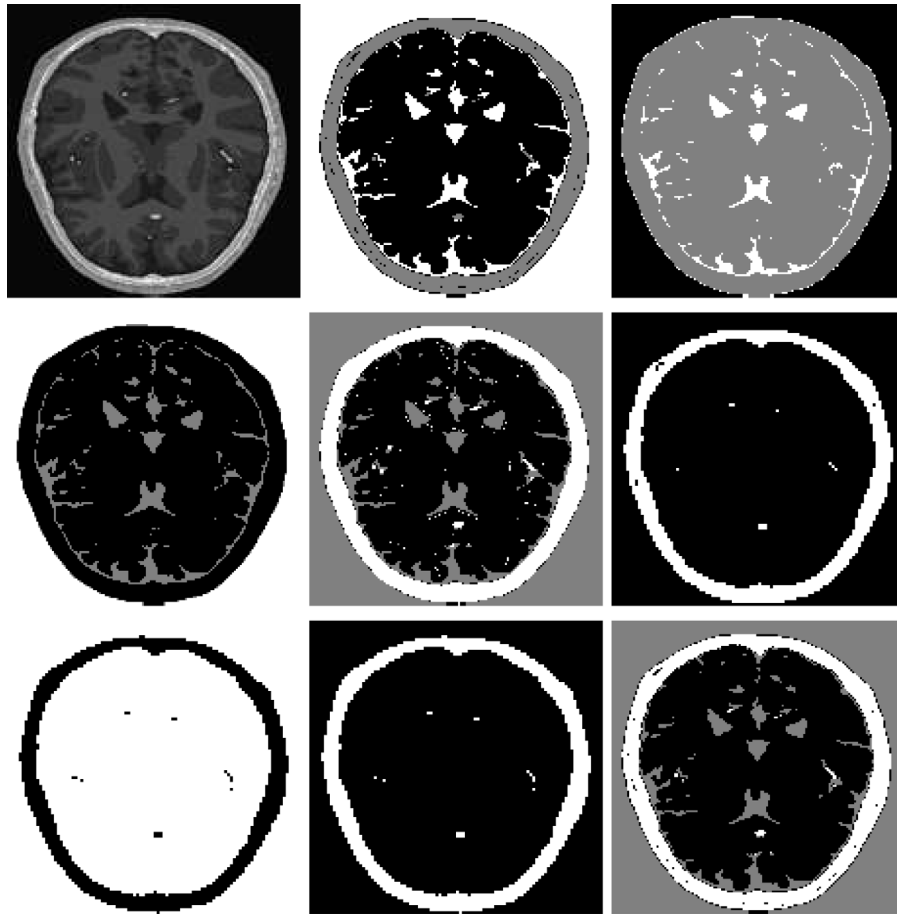


Fig. 2. Segmentation results of eight algorithms on a real brain MR image. Images from top-left to bottom-right in order are original image, images obtained from *k*-means, *k*-PC, *q*-flat, *k*PPC, TWSVC, RTWSVC, FRTWSVC and *k*SDC.

Table 5

Ranks of clustering methods on the MR image.

Data set	<i>k</i> -means Rank	<i>k</i> PC Rank	<i>q</i> -flat Rank	<i>k</i> PPC Rank	TWSVC Rank	RTWSVC Rank	FRTWSVC Rank	<i>k</i> SDC Rank
Original image	5	4	2	3	6	8	7	1
Salt & Pepper 0.5	5	6	3	2	7	8	4	1
Salt & Pepper 3	2	6	7	8	3	5	4	1
Salt & Pepper 18	4	3	6	8	2	5	7	1
Gaussian 0.5	4	7	6	8	2	3	5	1
Gaussian 3	2	5	7	8	3	6	4	1
Gaussian 18	1	6	7	8	3	4	5	2
Average rank	3.2857	5.2857	5.4286	6.4286	3.7143	5.5714	5.1429	1.1429

Table 6

Nemenyi test of all the methods on a MR image. Here the difference of average rank is computed as the difference between the average rank of other method and the average rank of our *k*SDC.

Method	10% critical CD value ($\alpha = 0.1$)	Difference of average ranks
<i>k</i> -means	3.6399	2.1429
<i>k</i> PC	3.6399	4.1429
<i>q</i> -flat	3.6399	4.2857
<i>k</i> PPC	3.6399	4.0588
TWSVC	3.6399	5.2857
RTWSVC	3.6399	2.5714
FRTWSVC	3.6399	4.4286
<i>k</i> SDC	3.6399	0.0000

clusters is set to 3. Fig. 3 illustrates the corresponding segmentation results. Clearly, *k*SDC is the most robust method comparing

to other methods. To show the results clearer, we also compute DR for all the methods and list them in Table 4. The results support our conclusion.

To further verify the robustness of our proposed *k*SDC, we add Salt & Pepper noise of larger levels 3 and 18 on the original image. We also consider Gaussian noise with different variances 0.5, 3, and 18 on the image. The results are listed in Table 4. It is clearly illustrated that the proposed *k*SDC algorithm gives better denoising performance than the other methods. Therefore, our *k*SDC is more robust.

Similar to the benchmark data sets, we use the Friedman test to make further comparison. It can be computed directly that for ranks of all the methods in Table 2, $F_F = 5.5187$. Note that the critical value of $F(7, 42)$ for significance level $\alpha = 0.1$ is 1.865. Clearly, $F_F = 5.5187 > 1.865$, and we can thus reject the null hypothesis that all the algorithms behave similarly. Therefore, we

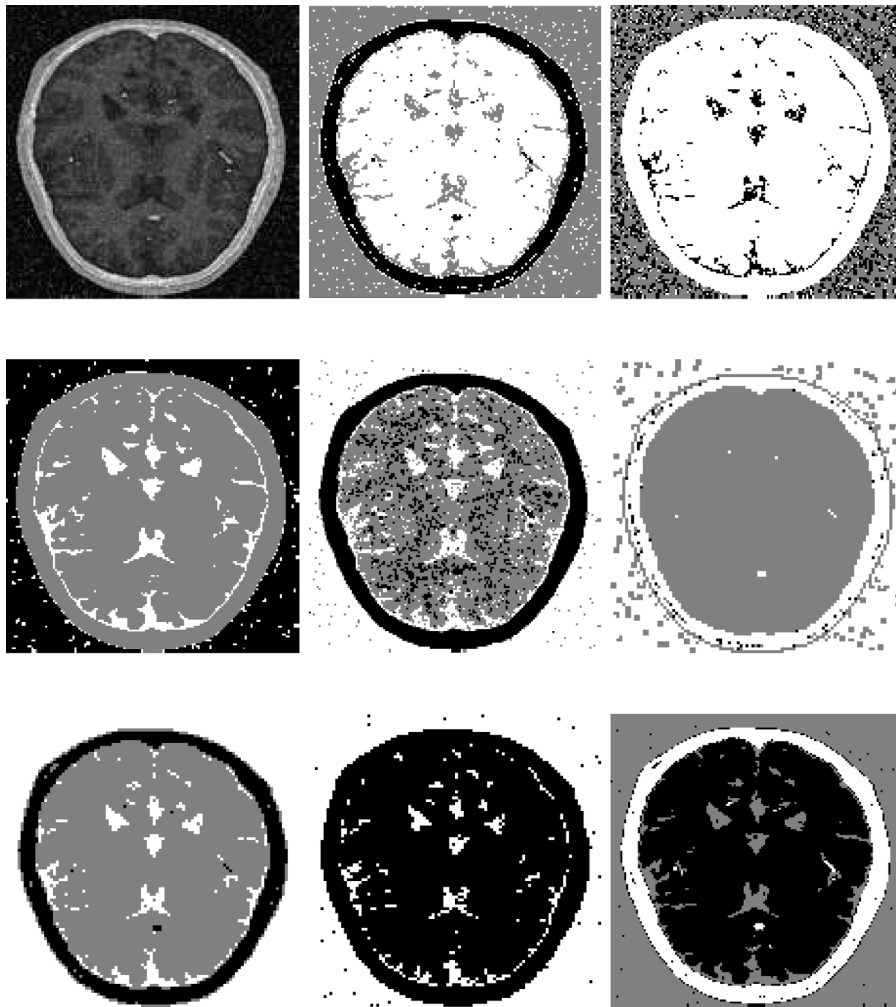


Fig. 3. Segmentation results of different algorithms on a real brain MR image with Salt & Pepper noise of strength 0.5. Images from top-left to bottom-right in order are noise image, images obtained from k -means, k -PC, q -flat, k PPC, TWSVC, RTWSVC, FRTWSVC and k SDC.

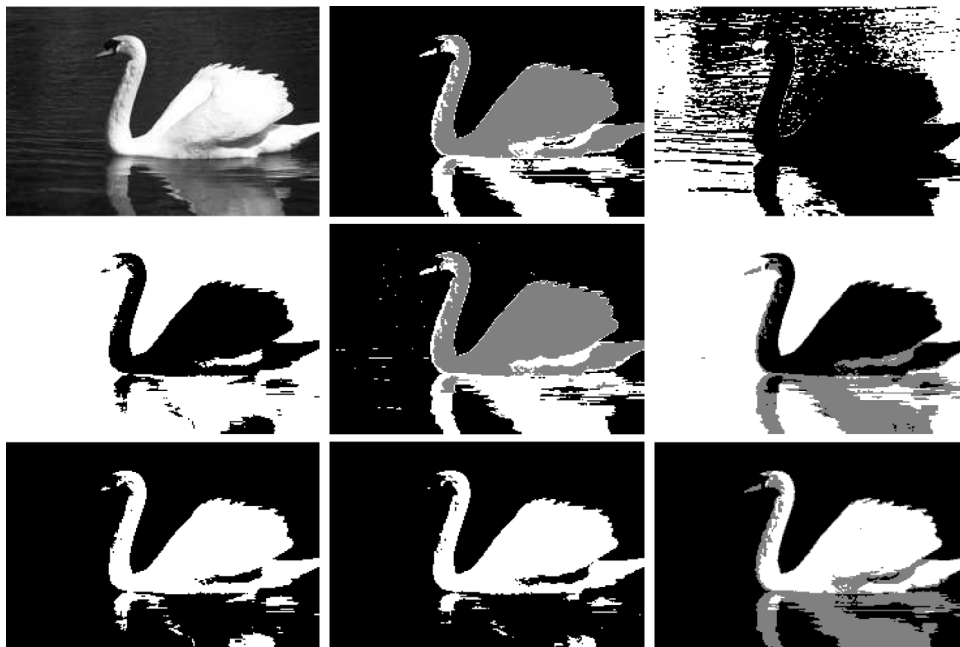


Fig. 4. Segmentation results of eight algorithms on a swan image. Images from top-left to bottom-right in order are original image, images obtained from k -means, k -PC, q -flat, k PPC, TWSVC, RTWSVC, FRTWSVC and k SDC.

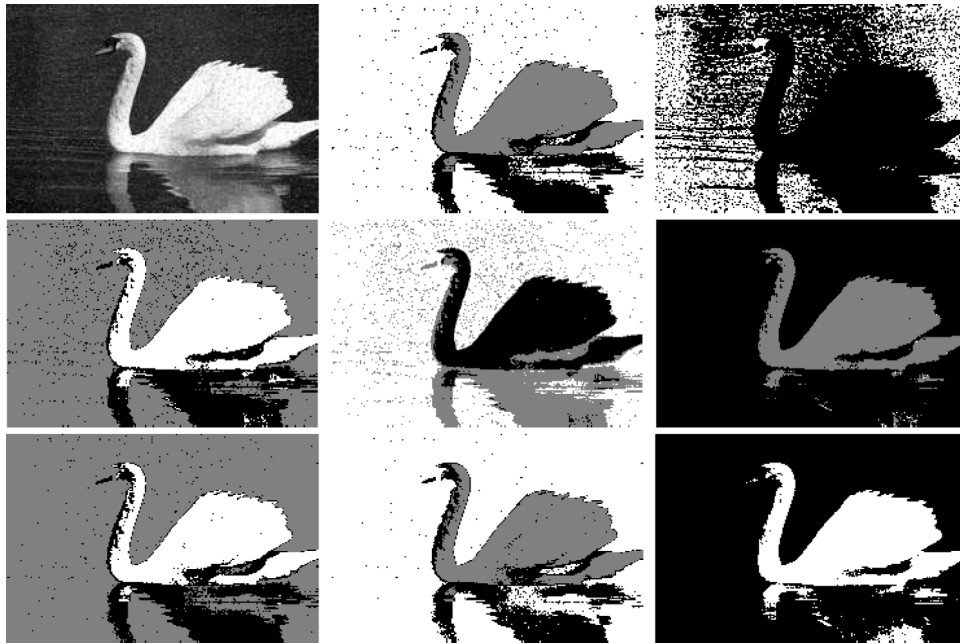


Fig. 5. Segmentation results of different algorithms on a swan image with Salt & Pepper noise of strength 5. Images from top-left to bottom-right in order are noise image, images obtained from *k*-means, *k*-PC, *q*-flat, *k*PPC, TWSVC, RTWSVC, FRTWSVC and *k*SDC.

Table 7

Dice ratios(%) of clustering methods on the swan image.

Noise type	<i>k</i> -means DR	<i>k</i> PC DR	<i>q</i> -flat DR	<i>k</i> PPC DR	TWSVC DR	RTWSVC DR	FRTWSVC DR	<i>k</i> SDC DR
Original image	79.82	66.69	81.41	79.59	83.91	81.56	81.87	86.37
Salt & Pepper 5	78.43	70.14	76.34	75.12	83.97	80.45	80.61	88.34
Salt & Pepper 10	71.47	69.23	70.18	71.71	73.74	72.40	68.53	74.11
Salt & Pepper 20	87.73	70.83	81.86	80.34	83.59	82.99	80.45	88.29
Gaussian 5	78.25	72.39	79.14	76.50	83.74	80.37	75.74	81.33
Gaussian 10	79.99	66.01	81.48	73.45	82.45	72.30	66.35	87.89
Gaussian 20	79.95	67.56	71.87	72.02	81.98	79.27	77.93	80.19

Table 8

Ranks of clustering methods on the swan image.

Data set	<i>k</i> -means Rank	<i>k</i> PC Rank	<i>q</i> -flat Rank	<i>k</i> PPC Rank	TWSVC Rank	RTWSVC Rank	FRTWSVC Rank	<i>k</i> SDC Rank
Original image	6	8	5	7	2	4	3	1
Salt & Pepper 5	5	8	6	7	2	4	3	1
Salt & Pepper 10	5	7	6	4	2	3	8	1
Salt & Pepper 20	2	8	5	7	3	4	6	1
Gaussian 5	5	8	4	6	1	3	7	2
Gaussian 10	4	8	3	5	2	6	7	1
Gaussian 20	3	8	7	6	1	4	5	2
Average rank	4.2857	7.8571	5.1429	6.0000	1.8571	4.0000	5.5714	1.2857

now perform a posthoc Nemenyi test. In this case, $q_{0.1} = 2.780$ and $CD = 3.6399$. The results of Nemenyi test that are shown in Table 3 indicate that *k*SDC significantly outperforms *k*PC, *q*-flat, *k*PPC, TWSVC and FRTWSVC, and is comparable to *k*-means and RTWSVC in terms of accuracy (see Tables 5 and 6).

To further verify the effectiveness of our method, we then perform experiments on a swan image and a human image from the Berkeley Segmentation Dataset.¹ The images are of 481×321 pixels and 321×481 pixels respectively, and we consider their corresponding gray images. Their corresponding segmentation results are shown in Figs. 4 and 6. To test the robustness of various methods, we also pollute the above two original images

by adding the Salt & Pepper noise of levels 5, 10, 20, and Gaussian noise of mean 0 and variances 5, 10, 20. Fig. 5 shows the segmentation result of the swan image with Salt & Pepper noise of strength 5, and Fig. 7 shows the segmentation result of the human image with Salt & Pepper noise of strength 10. We also report all the DR values on all the image cases in Tables 7 and 10. The ranks of all the methods on these images are presented in Tables 8 and 11. We then perform the Friedman test on the ranks. For the swan image, $F_F = 21.0789$, and for the face image, $F_F = 11.1978$. The results show that all the methods are statistically different on these image data. Therefore, we further apply a posthoc Nemenyi test on ranks to make deep comparison, and the corresponding results are shown in Tables 9 and 12. In summary, the above obtained results demonstrate the following observation: (i) *k*SDC owns the highest DR values on

¹ <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>



Fig. 6. Segmentation results of eight algorithms on a human image. Images from top-left to bottom-right in order are original image, images obtained from *k*-means, *k*-PC, *q*-flat, *k*PPC, TWSVC, RTWSVC, FRTWSVC and *k*SDC.

most of the image data; (ii) Different methods may have different behavior regarding ranks on different image data. However, our *k*SDC always has the highest average rank; (iii) The segmentation results support the robustness of the proposed method; (iv) The statistical test shows that our *k*SDC performs statistically different than other methods.

5. Conclusion

A robust *k*-subspace discriminative clustering method named *k*SDC has been studied in this paper. *k*SDC realizes discriminative clustering by choosing the most discriminative subspace. The L1-norm criterion of *k*SDC makes it more robust. *k*SDC is solved though an effective ADMM algorithm, and the experimental results on benchmark data and image segmentation data support the effectiveness of *k*SDC. This paper shows the prospect of the

Table 9

Nemenyi test of all the methods on swan image. Here the difference of average rank is computed as the difference between the average rank of other method and the average rank of our *k*SDC.

Method	10% critical CD value ($\alpha = 0.1$)	Difference of average ranks
<i>k</i> -means	3.6399	3.0000
<i>k</i> PC	3.6399	6.5714
<i>q</i> -flat	3.6399	3.8571
<i>k</i> PPC	3.6399	4.7143
TWSVC	3.6399	0.5714
RTWSVC	3.6399	2.7143
FRTWSVC	3.6399	4.2857
<i>k</i> SDC	3.6399	0.0000

extension the discriminant and local information in partition-based subspace clustering. In the future, we will study discriminant analysis for different types of clustering.

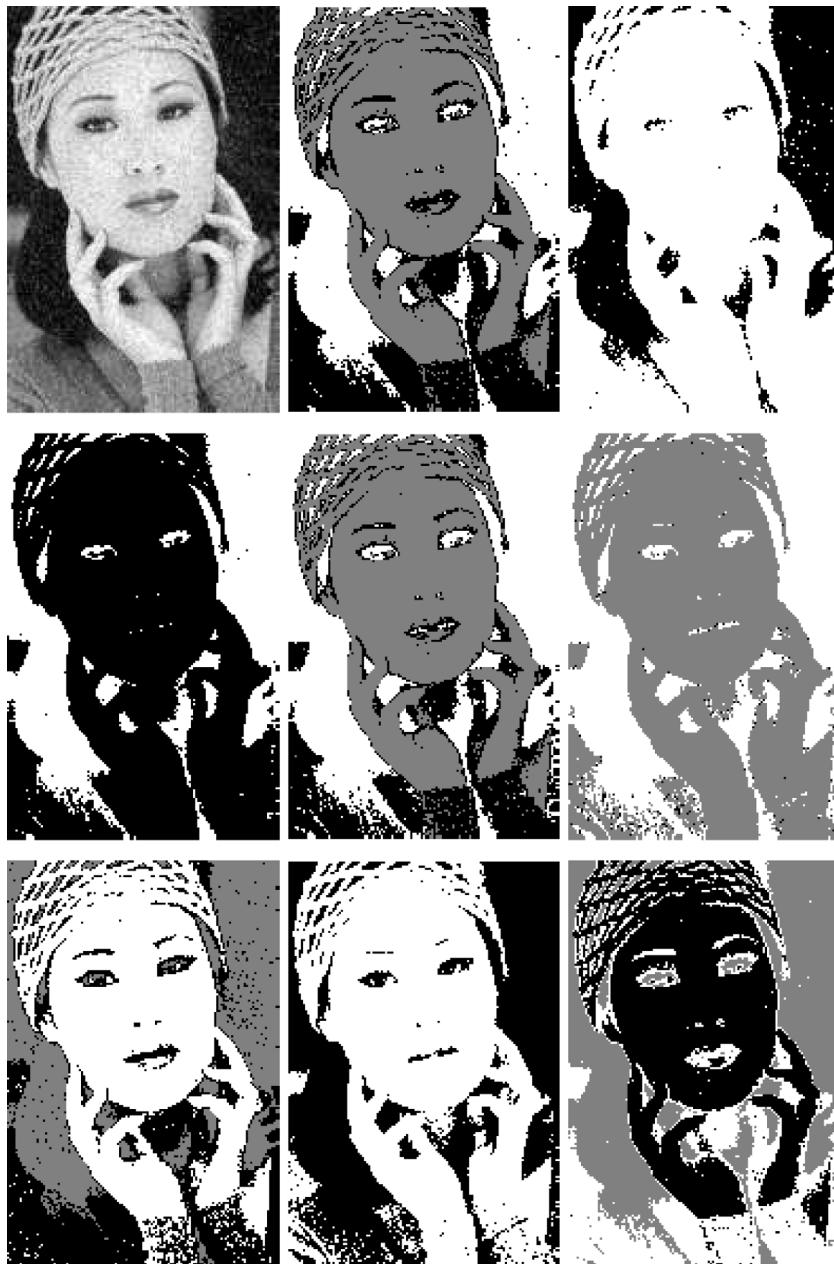


Fig. 7. Segmentation results of different algorithms on a human image with Salt & Pepper noise of strength 10. Images from top-left to bottom-right in order are noise image, images obtained from *k*-means, *k*-PC, *q*-flat, *k*PPC, TWSVC, RTWSVC, FRTWSVC and *k*SDC.

Table 10

Dice ratios(%) of clustering methods on the human image.

Noise type	<i>k</i> -means DR	<i>k</i> PC DR	<i>q</i> -flat DR	<i>k</i> PPC DR	TWSVC DR	RTWSVC DR	FRTWSVC DR	<i>k</i> SDC DR
Original image	84.80	71.57	76.37	73.67	77.09	75.47	79.28	83.80
Salt & Pepper 5	79.00	77.30	79.48	76.60	78.19	75.96	77.01	79.92
Salt & Pepper 10	83.03	66.98	79.49	70.56	69.74	73.54	73.67	84.58
Salt & Pepper 20	84.17	72.63	80.58	82.99	84.94	76.71	83.21	86.25
Gaussian 5	78.25	72.39	79.14	76.50	83.74	80.37	75.74	81.33
Gaussian 10	80.59	76.31	72.19	70.74	79.89	75.04	80.36	82.46
Gaussian 20	82.34	70.61	80.35	74.17	83.17	77.72	82.62	83.54

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to <https://doi.org/10.1016/j.asoc.2019.105858>.

Acknowledgments

This work is supported by the Natural Science Foundation of Hainan Province, PR China (No. 118QN181), the National Natural Science Foundation of China (No. 61703370, No. 61866010, No. 11871183, No. 11501310 and No. 61603338), the Natural Science

Table 11
Ranks of clustering methods on the human image.

Data set	k-means Rank	kPC Rank	q-flat Rank	kPPC Rank	TWSVC Rank	RTWSVC Rank	FRTWSVC Rank	kSDC Rank
Original image	1	8	5	7	4	6	3	2
Salt & Pepper 5	3	5	2	7	4	8	6	1
Salt & Pepper 10	2	8	3	6	7	5	4	1
Salt & Pepper 20	3	8	6	5	2	7	4	1
Gaussian 5	5	8	4	6	1	3	7	2
Gaussian 10	2	5	7	8	4	6	3	1
Gaussian 20	4	8	5	7	2	6	3	1
Average rank	2.8571	7.1429	4.5714	6.5714	3.4286	5.8571	4.2857	1.2857

Table 12

Nemenyi test of all the methods on the human image. Here the difference of average rank is computed as the difference between the average rank of other method and the average rank of our kSDC.

Method	10% critical CD value ($\alpha = 0.1$)	Difference of average ranks
k-means	3.6399	1.5714
kPC	3.6399	5.8571
q-flat	3.6399	3.2857
kPPC	3.6399	5.2857
TWSVC	3.6399	2.1429
RTWSVC	3.6399	4.5714
FRTWSVC	3.6399	3.0000
kSDC	3.6399	0.0000

Foundation of Zhejiang Province, PR China (No. LQ17F030003, No. LY18G010018 and No. LY16A010020), and the Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region, PR China (No. NJYT-19-B01).

References

- [1] M.W. Berry, Survey of Text Mining: Clustering, Classification, and Retrieval, Vol. 1, Springer, Berlin, Germany, 2004.
- [2] L.J. Van't Veer, H. Dai, M.J. Van De Vijver, et al., Gene expression profiling predicts clinical outcome of breast cancer, *Nature* 415 (6871) (2002) 530–536.
- [3] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, *Science* 344 (6191) (2014) 1492–1496.
- [4] J. Wang, M. She, Probabilistic latent semantic analysis for multichannel biomedical signal clustering, *IEEE Signal Process. Lett.* 23 (12) (2016) 1821–1824.
- [5] K. Sun, W. Tao, A constrained radial agglomerative clustering algorithm for efficient structure from motion, *IEEE Signal Process. Lett.* 25 (7) (2018) 1089–1093.
- [6] G.B. Coleman, H.C. Andrews, Image segmentation by clustering, *Proc. IEEE* 67 (5) (1979) 773–785.
- [7] T.X. Pham, P. Siarry, H. Oulhadj, Integrating fuzzy entropy clustering with an improved PSO for MRI brain image segmentation, *Appl. Soft Comput.* 65 (2018) 230–242.
- [8] N. Mahata, S. Kahali, S.K. Adhikari, et al., Local contextual information and Gaussian function induced fuzzy clustering algorithm for brain MR image segmentation and intensity inhomogeneity estimation, *Appl. Soft Comput.* 68 (2018) 586–596.
- [9] T. Lei, X. Jia, Y. Zhang, et al., Superpixel-based fast fuzzy C-means clustering for color image segmentation, *IEEE Trans. Fuzzy Syst.* (2018).
- [10] C. Gerloff, A. Kemper, C. Kilger, et al., Partition-based clustering in object bases: From theory to practice, in: D.B. Lomet (Ed.), *Foundations of Data Organization and Algorithms*, FODO 1993, in: *Lecture Notes in Computer Science*, vol. 730, Springer, Berlin, Heidelberg, 1993.
- [11] A.K. Jain, R.C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Upper Saddle River, NJ, USA, 1988.
- [12] P.S. Bradley, O.L. Mangasarian, *k*-Plane clustering, *J. Global Optim.* 16 (1) (2000) 23–32.
- [13] P. Tseng, Nearest *q*-flat to *m* points, *J. Optim. Theory Appl.* 105 (1) (2000) 249–252.
- [14] R.A. Fisher, The use of multiple measurements in taxonomic problems, *Ann. Eugen.* 7 (2) (1936) 179–188.
- [15] Y.H. Shao, L. Bai, Z. Wang, et al., Proximal plane clustering via eigenvalues, *Procedia Comput. Sci.* 17 (2013) 41–47.
- [16] Z. Wang, Y.H. Shao, L. Bai, et al., Twin support vector machine for clustering, *IEEE Trans. Neural Netw. Learn. Syst.* 26 (10) (2015) 2583–2588.
- [17] Z. Wang, Y.H. Shao, L. Bai, et al., Insensitive stochastic gradient twin support vector machines for large scale problems, *Inform. Sci.* 462 (2018) 114–131.
- [18] Z. Wang, Y.H. Shao, L. Bai, et al., A general model for plane-based clustering with loss function, 2019, arXiv preprint arXiv:1901.09178.
- [19] L. Bai, Y.H. Shao, Z. Wang, et al., Clustering by twin support vector machine and least square twin support vector classifier with uniform output coding, *Knowl.-Based Syst.* 163 (2019) 227–240.
- [20] Z.M. Yang, Y.R. Guo, C.N. Li, et al., Local *k*-proximal plane clustering, *Neural Comput. Appl.* 26 (1) (2015) 199–211.
- [21] Q. Ke, T. Kanade, Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, 2005, pp. 739–746.
- [22] C.N. Li, Y.H. Shao, N.Y. Deng, Robust L1-norm two-dimensional linear discriminant analysis, *Neural Netw.* 65 (2015) 92–104.
- [23] C.N. Li, Y.H. Shao, W. Yin, M.Z. Liu, Robust and sparse linear discriminant analysis via alternating direction method of multipliers, *IEEE Trans. Neural Netw. Learn. Syst.* (2019) <http://dx.doi.org/10.1109/TNNLS.2019.2910991>.
- [24] C. Kim, D. Klabjan, A simple and fast algorithm for L1-norm kernel PCA, *IEEE Trans. Pattern Anal. Mach. Intell.* (2019) <http://dx.doi.org/10.1109/TPAMI.2019.2903505>.
- [25] Q. Ye, H. Zhao, Z. Li, et al., L1-norm distance minimization-based fast robust twin support vector *k*-plane clustering, *IEEE Trans. Neural Netw. Learn. Syst.* 29 (9) (2017) 4494–4503.
- [26] C.N. Li, Y.H. Shao, Z. Wang, et al., Robust Bhattacharyya bound linear discriminant analysis through adaptive algorithm, *Knowl.-Based Syst.* 183 (2019) 104858.
- [27] K. Fukunaga, *Introduction To Statistical Pattern Recognition*, second ed., Academic Press, New York, 1991.
- [28] S. Boyd, N. Parikh, E. Chu, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.* 3 (1) (2011) 1–122.
- [29] S.A. Mulaik, *The Foundations of Factor Analysis*, McGraw-Hill, New York, 1972.
- [30] F. Nie, R. Zhang, X. Li, A generalized power iteration method for solving quadratic problem on the Stiefel manifold, *Sci. China Inf. Sci.* 60 (11) (2017) 112101:1–112101-10.
- [31] G.H. Golub, Van Loan C.F., *Matrix Computations*, third ed., The Johns Hopkins Univ. Press, Baltimore, MD, USA, 1996.
- [32] Q. Ye, H. Zhao, Z. Li, et al., L1-norm distance minimization-based fast robust twin support vector *k*-plane clustering, *IEEE Trans. Neural Netw. Learn. Syst.* 29 (9) (2018) 4494–4503.
- [33] D.T. Larose, *k*-Nearest neighbor algorithm, in: *Discovering Knowledge in Data: An Introduction To Data Mining*, Wiley, Warwick, U.K., 2005.
- [34] P.N. Tan, M. Steinbach, V. Kumar, *Introduction To Data Mining*, first ed., Addison-Wesley, Boston, MA, USA, 2005.
- [35] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *J. Amer. Statist. Assoc.* 32 (1937) 675–701.
- [36] R.L. Iman, J.M. Davenport, Approximations of the critical region of the Friedman statistic, *Commun. Stat.* (1980) 571–595.
- [37] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (Jan) (2006) 1–30.
- [38] P.B. Nemenyi, *Distribution-Free Multiple Comparisons* (Ph.D. thesis), Princeton University, 1963.
- [39] R.K.S. Kwan, A.C. Evans, G.B. Pike, MRI simulation-based evaluation of image-processing and classification methods, *IEEE Trans. Med. Imaging* 18 (11) (1999) 1085–1097.