Flexible Subspace Clustering: A Joint Feature Selection and K-Means Clustering Framework[☆]

Zhong-Zhen Long[a], Guoxia Xu[b], Jiao Du[c], Hu Zhu[d], Taiyu Yan[e], Yu-Feng Yu[f,*]

[a]*Shenzhen Securities Communication Co., Ltd., Shenzhen 518041, China.*
[b]*Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjovik, Norway.*
[c]*School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China.*
[d]*College of Telecommunication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China.*
[e]*Department of Imaging and Interventional Radiology, Chinese University of Hong Kong, Hong Kong, China.*
[f]*Department of Statistics, Guangzhou University, Guangzhou 510006, China.*

Abstract

Regarding as an important computing paradigm, cloud computing is to address big and distributed databases and rather simple computation. In this paradigm, data mining is one of the most important and fundamental problems. A large amount of data is generated by sensors and other intelligent devices. Data mining for these big data is crucial in various applications. K-means clustering is a typical technique to group the similar data into the same clustering, and has been commonly used in data mining. However, it is still a challenge to the data con-

[☆]This work was supported in part by National Natural Science Foundation of China under Grant 62006056 and Grant 61802148, in part by National Statistical Science Research Project of China under Grant 2020LY090, and in part by the Natural Science Foundation of Guangdong Province under Grant 2019A1515011266.

[*]Corresponding author

Email addresses: longzhzh@mail2.sysu.edu.cn (Zhong-Zhen Long), gxxu.re@gmail.com (Guoxia Xu), djiao@gzhu.edu.cn (Jiao Du), peter.hu.zhu@gmail.com (Hu Zhu), taiyuyan@163.com (Taiyu Yan), yuyufeng220@163.com (Yu-Feng Yu)

taining a large amount of noise, outliers and redundant features. In this paper, we propose a robust K-means clustering algorithm, namely, flexible subspace clustering. The proposed method incorporates feature selection and K-means clustering into a unified framework, which can select the refined features and improve the clustering performance. Moreover, for the purpose of enhancing the robustness, the $l_{2.p}$-norm is embedded into the objective function. We can flexibly choose appropriate p according to the different data and thus obtain more robust performance. Experimental results verify the presented method has more robust and better performance on benchmark databases compared to the existing approaches.
Keywords: Big data, K-means clustering, cloud computing, subspace learning.

1. Introduction

The cloud computing with big data is regarded as an important paradigm, which handles big and distributed databases and rather simple computation. Many interesting studies concentrate on cloud security [3], smart service [2] and mobile cloud computing [1]. However, most of these papers focus on hardware, data storage and management in clouds. Recently, the internet of things (IoT) is gaining increasing attention and many related studies are proposed in various applications such as quality prediction [4][5], dynamic resource discovery [6], bearing test [7] and feature recognition [8][9]. IoT and big data have an increasing impact on the future development of cloud computing. In IoT, enormous amount of data generated by sensors and other intelligent devices contain valuable information, but also encompass a large amount of noise, outliers and redundant features. Thus, data mining for these big data is crucial to be suitable for various applications. As one of the most important and fundamental technique in data mining, clustering

has been studied a lot and applied in many fields, such as resource scheduling in cloud computing [10], abnormal behavior detection in cloud [11], clinical observation [12], heterogeneous data analysis [13] and so on. Clustering is a kind of unsupervised learning, which groups the similar data points into the same cluster. As for the similarity, the most common used criterion is the distance, and K-means (KM) is a typical algorithm of this criterion.

The classical K-means distributes data points to k different clusters using l_2-norm distance. It's simple and easy to be solved, but easily affected by outliers and noises [14]. To overcome this problem, one direction is to use a distance measure that can be more robust. The use of l_p-norm is a successful extension. Hathaway et al. [15] conclude that $p = 1$ shows its property of robustness and choosing the value of p can provide better clustering results than fixing p as 1 or 2 but the model could be difficult to be solved. Salem et al. [16] adopt l_1-norm to evaluate the similarity between the observation and the centroid, which is shown efficiency and suitable to noisy data and outliers. Cai et al. [17] propose a multi-view K-means clustering based on $l_{2,1}$-norm. Liang et al. [18] propose a robust K-means using $l_{2,1}$-norm in the feature space and then extend it to the kernel space. The reform of the distance metric can improve the performance of K-means algorithm, which has been demonstrated in the above literatures.

However, with the development of science and technology, the data in real life is explosive. Big data sets generated from many fields contains a large amount of attributes, and some of which are noise and redundant attributes. It poses a remarkable challenge on the traditional clustering methods. For example, in face recognition applications, given a face image data of 128×128 resolution which is relatively small, it will generate a 16384-dimensional feature vector. This

kind of high-dimensional data always contains a large amount of noises, outliers and redundant features. It is difficult to cluster directly, and sometimes leads to high computational complexity and performance degradation [19], especially in K-means and its extensions. To deal with the curse of dimensionality and reduce the noise, outliers and redundant features, an intuitive approach is to conduct dimensionality reduction processing on the data before clustering. Many dimension reduction methods have been studied in the past decades, such as Principal Component Analysis (PCA) [20], Linear Discriminant Analysis (LDA) [21], sparse approximation to discriminant projection learning (SADPL) [22] and Locally Linear Embedding (LLE) [23]. PCAKM is a typical method that sequentially conducts PCA for dimension reduction and K-means for clustering [24]. Yin et al. [25] apply LLE to preprocess the data before performing K-means to make better use of the manifold information. These sequential methods can improve the computational efficiency, but the subspace got from the dimension reduction process may not be the optimal one for the clustering process, so that some researchers believe that the separation of dimension reduction and clustering may result in worse clustering performance [26].

Intuitively, if clustering is embedded into the process of dimension reduction, the performance of clustering may be improved. This kind of methods try to find the optimal structure of data in the low-dimensional feature space for clustering. They perform K-means and the subspace learning process simultaneously. For example, Ding et al. [28] construct an adaptive framework LDAKM, in which LDA and K-means are jointly implemented, that is, labels are generated by K-means algorithm, and the obtained labels are used by LDA to learn the subspace. Since LDA may fail when the number of samples is very small, several LDA's

extensions have been used to replace LDA, and get better results than LDAK-M [29], such as Maximum Margin Criterion (MMC) [30], Orthogonal Centroid Method (OCM) [31] and Orthogonal Least Squares Discriminant Analysis (OLS-DA) [32]. Hou et al. [29] consider the relation between PCA and K-means, and propose a general subspace clustering framework. This kind of algorithms have been proved to get better results than the sequential algorithms, but they also have some drawbacks. These algorithms all need to compute an approximate solution by eigenvalue decomposition, which will increase the computational burden so that when facing the high-dimensional data, these algorithms may fail. And since the optimal subspace is found by orthogonal linear transformation, it may have difficulty to understand the meaning of the obtained low-dimensional features. Wang et al. [27] construct a special feature selection matrix and propose a fast adaptive subspace clustering algorithm FAKM based on DEC, which can effectively select the most representative subspace without requiring eigenvalue decomposition. FAKM also performs adaptive learning to the K-means part.

Most methods mentioned above are based on the l_2-norm distance metric, which is known to be very sensitive to data outliers and noise. Therefore, it is meaningful to build a model with robust distance metric. Recently, $l_{2,p}$-norm is successfully used to replace l_2-norm as distance metric for improving the robustness, such as DCM [33] and $l_{2,p}$-PCA [34]. In $l_{2,p}$-PCA, $l_{2,p}$-norm is incorporated into PCA, and it is robust to outliers and can retain the desirable properties from big data. Inspired by FAKM and $l_{2,p}$-PCA, we propose a flexible subspace clustering method. Our method flexibly chooses appropriate p according to the data and thus obtains more robust clustering performance. Several experimental results on various datasets prove the effectiveness of the proposed algorithm.

The main contributions of our paper are listed as follows.

- The proposed algorithm combines the feature selection and clustering into a single framework jointly.
- The use of $l_{2,p}$-norm on K-means makes our algorithm robust to noise and redundant features of big data.
- The proposed approach is neither convex nor Lipschitz continuous, thus it is difficult to be solved directly. We propose an iterative algorithm to optimize it.

The rest of the paper is organized as follows. We propose our model and derive an efficient algorithm to optimize the model in Section 2. In Section 3, the proposed model is evaluated on the benchmark databases. Finally, we draw the conclusion in Section 4.

2. The Proposed Method

In this section, we introduce the details about the proposed method for clustering. The main content will be separated into the following several parts including the formulation of the proposed approach, an efficient algorithm, convergence and computational complexity analysis.

2.1. Formulation

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in R^{D \times n}$ be a high-dimensional data matrix, and $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_c] \in R^{D \times c}$ be c centroid vectors. $\mathbf{F} \in \{0, 1\}^{n \times c}$ denotes the indicator matrix, here $F_{ik} = 1$ if $\mathbf{x}_i$ belongs to the k-th cluster, otherwise $F_{ik} = 0$.

Following [37][38], we can obtain the K-means formulation as

$$\min_{\mathbf{F},\mathbf{Z}} \sum_{i=1}^{n}\sum_{k=1}^{c} F_{ik} \parallel \mathbf{x}_i - \mathbf{z}_k \parallel_2^2, \tag{1}$$

$$\text{s.t. } \mathbf{F} \in \{0,1\}^{n\times c}, \mathbf{F1} = \mathbf{1}.$$

Taking a simple algebra, the objective in (1) becomes

$$\min_{\mathbf{F},\mathbf{Z}} \parallel \mathbf{X} - \mathbf{Z}\mathbf{F}^T \parallel_F^2, \tag{2}$$

$$\text{s.t. } \mathbf{F} \in \{0,1\}^{n\times c}, \mathbf{F1} = \mathbf{1}.$$

Considering that the high-dimensional data could contain a large amount of noises, outliers and redundant features. It leads to high computational complexity and performance degradation. The direct idea is to find a transformation matrix $\mathbf{W} \in R^{D\times d}$ which transforms the high-dimensional features to a low-dimensional feature space $\mathbf{Y} = \mathbf{W}^T\mathbf{X}$, where $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n] \in R^{d\times n}$. Following the feature selection [39][27], we use the column vectors $\mathbf{w}_i$ as follow

$$\mathbf{w}_i = [\underbrace{0, ..., 0}_{i-1}, 1, \underbrace{0, ..., 0}_{D-i}]^T. \tag{3}$$

Then the feature selection matrix $\mathbf{W}$ can be represented as

$$\mathbf{W} = [\mathbf{w}_{I(1)}, \mathbf{w}_{I(2)}, ..., \mathbf{w}_{I(d)}], \tag{4}$$

where I is a permutation of $\{1, 2, ..., D\}$. It can be seen that the transformation matrix $\mathbf{W}$ is sparse and column-full-rank.

To achieve the goal of feature selection and K-means clustering simultaneously, we incorporate the subspace learning and K-means clustering into a unified

framework as

$$\begin{aligned} &\max_{\mathbf{W},\mathbf{G},\mathbf{F}} Tr(\mathbf{W}^T\mathbf{S}_t\mathbf{W}) - \lambda \parallel \mathbf{W}^T\mathbf{X} - \mathbf{G}\mathbf{F}^T \parallel_2^p \\ \text{s.t. } &\mathbf{W} \in \{0,1\}^{D\times d}, \text{rank}(\mathbf{W}) = d, \mathbf{W}^T\mathbf{1} = \mathbf{1}, \\ &\mathbf{F} \in \{0,1\}^{n\times c}, \mathbf{F}\mathbf{1} = \mathbf{1}, \end{aligned} \tag{5}$$

where $\mathbf{S}_t = \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T$ is the total scatter matrix. $\mathbf{G} = [\mathbf{g}_1, \cdots, \mathbf{g}_c] \in R^{d\times c}$ is c centroid vectors in the low-dimensional space. It should be noted that FAKM [27] is also a joint model of subspace learning and clustering. It uses the pattern of $\parallel \mathbf{M} \parallel_\sigma = \sum_i \frac{(1+\sigma)\|\mathbf{m}_i\|_2^2}{\|\mathbf{m}_i\|_2+\sigma}$ to construct the K-means clustering, here $\mathbf{M}$ is an arbitrary matrix, $\mathbf{m}_i$ is the i-th column and σ is a parameter. Different from FAKM, the model in (5) has more robust performance since it adopts $l_{2,p}$-norm to construct the K-means clustering and can flexibly choose appropriate p according to the different data.

2.2. Optimization

Since our objective function (5) involves $l_{2,p}$-norm, it is difficult to get its closed-form solution directly. In [34], an iterative algorithm is proposed to solve the objective function in the form of $l_{2,p}$-norm. Similar techniques are used in [35] to solve the problem of the minimization of LDA with regular term based on $l_{2,p}$-norm $(0 < p \leq 2)$. Inspired by these papers, we propose an effective iterative algorithm to solve our objective function.

Let $d_i = \frac{p}{2}\|\mathbf{W}^T\mathbf{x}_i - \mathbf{G}\mathbf{f}_i\|_2^{p-2}$, then (5) can be transformed to

$$\max_{\mathbf{W},\mathbf{G},\mathbf{F}} Tr(\mathbf{W}^T\mathbf{S}_t\mathbf{W}) - 2\lambda/p \sum_{i=1}^{n} d_i\|\mathbf{W}^T\mathbf{x}_i - \mathbf{G}\mathbf{f}_i\|_2^2. \tag{6}$$

Since λ is an arbitrary constant, for convenience, we will still mark $2\lambda/p$ as λ.

Denote Δ as a diagonal matrix with its i-th diagonal element as d_i, and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n] = \mathbf{W}^T\mathbf{X} - \mathbf{G}\mathbf{F}^T$, where $\mathbf{u}_i \in \mathbb{R}^d$ is the i-th column of $\mathbf{U}$. We have

$$\max_{\mathbf{W},\mathbf{G},\mathbf{F},\Delta} Tr(\mathbf{W}^T\mathbf{S}_t\mathbf{W}) - \lambda Tr(\mathbf{U}^T\Delta\mathbf{U}). \tag{7}$$

Since the objective function in (7) is not jointly convex with all the variables, and Δ is dependent on $\mathbf{W}, \mathbf{F}$ and $\mathbf{G}$, we propose the following iterative algorithm to alternatively update $\mathbf{W}, \mathbf{G}, \mathbf{F}$ and Δ.

Step 1: Fixing $\mathbf{W}, \mathbf{G}$ and Δ and Optimizing $\mathbf{F}$.

When $\mathbf{W}, \mathbf{G}$ and Δ are fixed, the first term in (7) is constant, and we only need to minimize the second term. The optimization problem becomes

$$\min_{\mathbf{F}} \sum_{i=1}^{n} d_i \|\mathbf{W}^T\mathbf{x}_i - \mathbf{G}\mathbf{f}_i\|_2^2 = \min_{\mathbf{F}} \sum_{i=1}^{n} d_i \sum_{k=1}^{c} \|\mathbf{W}^T\mathbf{x}_i - \mathbf{G}\|_2^2 F_{ik}, \tag{8}$$

Since $\mathbf{G}$ is fixed and $\mathbf{F}$ is the cluster indicator matrix, according to the algorithm in [29] and [27], the optimized $\mathbf{F}$ can be derived from

$$\mathbf{F}_{ij} = \begin{cases} 1, & j = arg\min_k \|\mathbf{W}^T\mathbf{x}_i - \mathbf{g}_k\|_2^2, \\ 0, & Otherwise. \end{cases} \tag{9}$$

Step 2: Fixing Δ and $\mathbf{F}$ and Optimizing $\mathbf{W}$ and $\mathbf{G}$.

When Δ and $\mathbf{F}$ are fixed, the closed-form solution of $\mathbf{W}$ and $\mathbf{G}$ can be derived as follows. Denote

$$\mathcal{L}(\mathbf{W}, \mathbf{G}) = Tr(\mathbf{W}^T\mathbf{S}_t\mathbf{W}) - \lambda Tr(\mathbf{U}^T\Delta\mathbf{U}), \tag{10}$$

where $\mathbf{U} = \mathbf{W}^T\mathbf{X} - \mathbf{G}\mathbf{F}^T$.

We take a derivative of $\mathcal{L}(\mathbf{W}, \mathbf{G})$ over $\mathbf{G}$

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{W}, \mathbf{G})}{\partial \mathbf{G}} &= -\lambda \frac{\partial Tr((\mathbf{W}^T\mathbf{X} - \mathbf{G}\mathbf{F}^T)^T \Delta (\mathbf{W}^T\mathbf{X} - \mathbf{G}\mathbf{F}^T))}{\partial \mathbf{G}}, \\ &= -\lambda \frac{\partial Tr((\mathbf{G}\mathbf{F}^T - \mathbf{W}^T\mathbf{X})^T \Delta (\mathbf{G}\mathbf{F}^T - \mathbf{W}^T\mathbf{X}))}{\partial \mathbf{G}}, \\ &= -\lambda \frac{\partial Tr(\mathbf{G}\mathbf{F}^T\Delta\mathbf{F}\mathbf{G}^T) - 2Tr(\mathbf{G}\mathbf{F}^T\Delta\mathbf{X}^T\mathbf{W})}{\partial \mathbf{G}}, \\ &= -2\lambda(\mathbf{G}\mathbf{F}^T\Delta\mathbf{F} - \mathbf{W}^T\mathbf{X}\Delta\mathbf{F}).\end{aligned} \tag{11}$$

Let the above equation equals to zero, and we have

$$\mathbf{G} = \mathbf{W}^T\mathbf{X}\Delta\mathbf{F}(\mathbf{F}^T\Delta\mathbf{F})^{-1}. \tag{12}$$

Substituting $\mathbf{G}$ into $\mathcal{L}(\mathbf{W}, \mathbf{G})$, we have

$$\begin{aligned}\mathcal{L}(\mathbf{W}) &= Tr(\mathbf{W}^T\mathbf{S}_t\mathbf{W}) - \lambda Tr((\mathbf{W}^T\mathbf{X} - \mathbf{G}\mathbf{F}^T)^T\Delta(\mathbf{W}^T\mathbf{X} - \mathbf{G}\mathbf{F}^T), \\ &= Tr(\mathbf{W}^T\mathbf{S}_t\mathbf{W}) - \lambda Tr(\mathbf{W}^T\mathbf{X}\Delta\mathbf{X}^T\mathbf{W} - \mathbf{W}^T\mathbf{X}\mathbf{F}(\mathbf{F}^T\Delta\mathbf{F})^{-1}\mathbf{F}^T\Delta^T\mathbf{X}^T\mathbf{W}), \\ &= Tr(\mathbf{W}^T(\mathbf{S}_t - \lambda\mathbf{X}\Delta\mathbf{X}^T + \lambda\mathbf{X}\Delta\mathbf{F}(\mathbf{F}^T\Delta\mathbf{F})^{-1}\mathbf{F}^T\Delta\mathbf{X}^T)\mathbf{W}), \\ &= Tr(\mathbf{W}^T\mathbf{M}\mathbf{W}),\end{aligned} \tag{13}$$

where $\mathbf{M} = \mathbf{S}_t - \lambda\mathbf{X}\Delta\mathbf{X}^T + \lambda\mathbf{X}\Delta\mathbf{F}(\mathbf{F}^T\Delta\mathbf{F})^{-1}\mathbf{F}^T\Delta\mathbf{X}^T$.

Therefore the problem to optimize $\mathbf{W}$ becomes

$$\max_{\mathbf{W}} Tr(\mathbf{W}^T\mathbf{M}\mathbf{W}) = \max_{\mathbf{W}} \sum_{i=1}^{d} Tr(\mathbf{w}_i^T\mathbf{M}\mathbf{w}_i), \tag{14}$$

According to the definition of $\mathbf{W}$ in (4), we can optimize $\mathbf{W}$ by locating the first d largest diagonal elements of matrix $\mathbf{M}$.

Step 3: Updating Δ by calculating its i-th diagonal element as

$$d_i = \frac{p}{2}\|\mathbf{W}^T\mathbf{x}_i - \mathbf{G}\mathbf{f}_i\|_2^{p-2}. \tag{15}$$

It is important to note that there is a problem when using the above alternative algorithm. Although the above solving strategy can guarantee convergence, its result is not satisfactory. Like the traditional K-means method, there are a lot of local optimizations which depend on initialization. Considering the above update rules, when $\mathbf{F}$ is fixed, the algorithm can quickly adjust $\mathbf{W}$ and $\mathbf{G}$ to adapt to the $\mathbf{F}$. In other words, when we need to update the $\mathbf{F}$ in the next step, the optimal $\mathbf{F}$ is the same as before. That is to say, the algorithm has fast convergence speed and the optimal solution depends on the initial value. In order to avoid the local optimal problem, the update rule proposed in [29] and [27] is employed. In each step of updating $\mathbf{F}$, we will randomly initialize $\mathbf{F}$ several times (20 times in our experiment). If the value of the objective function $\|\mathbf{W}^T\mathbf{X} - \mathbf{G}\mathbf{F}^T\|_F^2$ is smaller than that of the previous $\mathbf{F}$, then updating $\mathbf{F}$ according to the random initialization. Otherwise, updating $\mathbf{F}$ by (9). That is, assume that in the i-th iteration, we have gotten $\mathbf{F}_i^*$, $\mathbf{W}_i^*$ and $\mathbf{G}_i^*$. In the $(i+1)$-th iteration, we will get $\mathbf{F}_{i+1}^1, \mathbf{F}_{i+1}^2, ..., \mathbf{F}_{i+1}^t$ by random initialization, where t is the number of random initialization. We update $\mathbf{F}$ according to the following rules

$$\mathbf{F}_{i+1}^* = \begin{cases} \mathbf{F}_{i+1}^j, & \begin{array}{l}\|(\mathbf{W}_i^*)^T\mathbf{X} - \mathbf{G}_i^*(\mathbf{F}_{i+1}^j)^T\|_F^2 \\ < \|(\mathbf{W}_i^*)^T\mathbf{X} - \mathbf{G}_i^*(\mathbf{F}_i^*)^T\|_F^2,\end{array} \\ \mathbf{F}^*, & Otherwise, \end{cases} \tag{16}$$

where $\mathbf{F}^*$ is defined as

$$\mathbf{F}_{ij}^* = \begin{cases} 1, & j = arg\min_k \|(\mathbf{W}_i^*)^T x_i - (\mathbf{g}_i^*)_k\|_2^2, \\ 0, & Otherwise. \end{cases} \tag{17}$$

The pseudo code of optimizing the proposed algorithm is listed in Algorithm 1.

Algorithm 1 The Algorithm to Solve Problem (5)

Input: The input data $\mathbf{X} \in \mathbb{R}^{D\times n}$, the reduced dimension number d, the number of clusters c, regularization parameter λ, and the distance metric parameter p.

Output: Transformation matrix $\mathbf{W}$, cluster indicator matrix $\mathbf{F}$, and cluster centroid matrix $\mathbf{G}$.

1: Initialize Δ as identity matrix, and randomly initialize $\mathbf{W}$ and $\mathbf{G}$.

2:**while** Not convergent **do**

3: Update $\mathbf{F}$ by (16);

4: Update $\mathbf{G}$ by (12);

5: Update $\mathbf{W}$ by locating the d largest diagonal elements of the matrix $\mathbf{M}$ in (14);

6: Update Δ by calculating its diagonal elements by $d_i = \frac{p}{2}\|\mathbf{W}^T\mathbf{x}_i - \mathbf{G}\mathbf{f}_i\|_2^{p-2}$;

7: **end while**

2.3. Convergence Analysis

In this section, we prove the convergence of the proposed algorithm. First, we give the following Lemma:

Lemma 1 [34]: For any nonzero vectors $\mathbf{e}^{t+1}$, $\mathbf{e}^t \in \mathbb{R}^m$, when $0 < p \leq 2$, we have:

$$\frac{\|\mathbf{e}^{t+1}\|_2^p}{\|\mathbf{e}^t\|_2^p} - \frac{p}{2}\frac{\|\mathbf{e}^{t+1}\|_2^2}{\|\mathbf{e}^t\|_2^2} - 1 + \frac{p}{2} \leq 0. \tag{18}$$

Theorem 1: When $\mathbf{W}, \mathbf{G}$ and Δ are fixed, the derived $\mathbf{F}$ in (9) is the global solution to the problem (7). Similarly, when $\mathbf{F}$ and Δ are fixed, the derived $\mathbf{G}$ in (12) and the derived $\mathbf{W}$ by locating the d largest diagonal elements of $\mathbf{S}_t - \lambda\mathbf{X}\Delta\mathbf{X}^T + \lambda\mathbf{X}\Delta\mathbf{F}(\mathbf{F}^T\Delta\mathbf{F})^{-1}\mathbf{F}^T\Delta\mathbf{X}^T$ are also the global solutions to the problem in (7).

Proof: When $\mathbf{W}, \mathbf{G}$ and Δ are fixed, optimizing the problem in (7) is equal to solving the traditional K-means on $\mathbf{W}^T\mathbf{X}$ with fixed centroid. Thus the optimized solution is unique.

According to (3), $\mathbf{w}_i$ is a vector with only one element being 1 and the rest being 0. Obviously, the derived $\mathbf{W}$ by locating the d largest diagonal elements of $\mathbf{S}_t - \lambda\mathbf{X}\Delta\mathbf{X}^T + \lambda\mathbf{X}\Delta\mathbf{F}(\mathbf{F}^T\Delta\mathbf{F})^{-1}\mathbf{F}^T\Delta\mathbf{X}^T$ maximizes the objective function in (14). When $\mathbf{F}$ and Δ are fixed, $\mathbf{G}$ is dependent on $\mathbf{W}$, and the global solution of $\mathbf{W}$ can be derived from the process above.

To sum up, the theorem is proved.

Theorem 2: The procedure in Algorithm 1 monotonically increases the objective function of the problem in (5) in each iteration.

Proof: Assume that we have derived the updated $\mathbf{W}_t$, $\mathbf{G}_t$ in the t-th iteration. In the $(t+1)$-th iteration, we fix $\mathbf{W}_t$, $\mathbf{G}_t$ and Δ_t, and get the optimized $\mathbf{F}_{t+1}$ by (16). According to Theorem 1 and the updating rule in (9), we have

$$\begin{aligned} &Tr(\mathbf{W}_t^T\mathbf{S}_t\mathbf{W}_t) - \lambda\|\mathbf{W}_t^T\mathbf{X} - \mathbf{G}_t\mathbf{F}_t^T\|_{2,p}^p \\ \leq\ &Tr(\mathbf{W}_t^T\mathbf{S}_t\mathbf{W}_t) - \lambda\|\mathbf{W}_t^T\mathbf{X} - \mathbf{G}_t\mathbf{F}_{t+1}^t\|_{2,p}^p. \end{aligned} \tag{19}$$

Then we fix Δ_t and $\mathbf{F}_{t+1}$, and update $\mathbf{G}$ and $\mathbf{W}$ by maximizing (10). Let $f(\mathbf{W}) = Tr(\mathbf{W}_t^T\mathbf{S}_t\mathbf{W}_t)$, $\mathbf{u}_i^t = \mathbf{W}_t^T\mathbf{x}_i - \mathbf{G}_t(\mathbf{f}_i)_{t+1}$, and $\mathbf{u}_i^{t+1} = \mathbf{W}_{t+1}^T\mathbf{x}_i - \mathbf{G}_{t+1}(\mathbf{f}_i)_{t+1}$, we have

$$f(\mathbf{W}_t) - \lambda\sum_i d_i^t\|\mathbf{u}_i^t\|_2^2 \leq f(\mathbf{W}_{t+1}) - \lambda\sum_i d_i^t\|\mathbf{u}_i^{t+1}\|_2^2. \tag{20}$$

Since $d_i^t = \frac{p}{2}\|\mathbf{W}_t^T\mathbf{x}_i - \mathbf{G}_t(\mathbf{f}_i)_t\|_2^{p-2}$, thus we have

$$f(\mathbf{W}_t) - \lambda\sum_i \frac{p}{2}\|\mathbf{u}_i^t\|_2^p \leq f(\mathbf{W}_{t+1}) - \lambda\sum_i \frac{p}{2}\|\mathbf{u}_i^t\|_2^{p-2}\|\mathbf{u}_i^{t+1}\|_2^2, \tag{21}$$

which can be wrote as

$$f(\mathbf{W}_t) - \lambda \sum_i \frac{p}{2} \frac{\|\mathbf{u}_i^t\|_2^2}{\|\mathbf{u}_i^t\|_2^{2-p}} \leq f(\mathbf{W}_{t+1}) - \lambda \sum_i \frac{p}{2} \frac{\|\mathbf{u}_i^{t+1}\|_2^2}{\|\mathbf{u}_i^t\|_2^{2-p}}, \tag{22}$$

According to Lemma 1, we have

$$\frac{p}{2} \frac{\|\mathbf{u}_i^{t+1}\|_2^2}{\|\mathbf{u}_i^t\|_2^2} \|\mathbf{u}_i^t\|_2^p \geq \|\mathbf{u}_i^{t+1}\|_2^p - (1 - \frac{p}{2}) \|\mathbf{u}_i^t\|_2^p, \tag{23}$$

which holds for each index i, thus we have

$$\frac{p}{2} \sum_i \frac{\|\mathbf{u}_i^{t+1}\|_2^2}{\|\mathbf{u}_i^t\|_2^2} \|\mathbf{u}_i^t\|_2^p \geq \sum_i \|\mathbf{u}_i^{t+1}\|_2^p - (1 - \frac{p}{2}) \sum_i \|\mathbf{u}_i^t\|_2^p, \tag{24}$$

that is

$$-\sum_i \|\mathbf{u}_i^t\|_2^p + \frac{p}{2} \sum_i \frac{\|\mathbf{u}_i^t\|_2^2}{\|\mathbf{u}_i^t\|_2^{2-p}} \leq -\sum_i \|\mathbf{u}_i^{t+1}\|_2^p + \frac{p}{2} \sum_i \frac{\|\mathbf{u}_i^{t+1}\|_2^2}{\|\mathbf{u}_i^t\|_2^{2-p}}, \tag{25}$$

Combining (22) and (25), we have

$$f(\mathbf{W}_t) - \lambda \sum_i \|\mathbf{u}_i^t\|_2^p \leq f(\mathbf{W}_{t+1}) - \lambda \sum_i \|\mathbf{u}_i^{t+1}\|_2^p. \tag{26}$$

To sum up, Algorithm 1 monotonically increases the objective function of the problem in (5) in each iteration. Since (5) has an obvious upper bound $Tr(\mathbf{X}\mathbf{X}^T)$, Algorithm 1 will monotonically increase the objective function until it converges.

2.4. Complexity Analysis

First we consider the computation complexity of Algorithm 1. It contains three main components, i.e., K-means in the subspace with computation complexity $O(dcn)$, the process of computing matrix G with computation complexity $O(dcn + c^2 n)$ and computing matrix $\mathbf{M}$'s diagonal elements to optimize $\mathbf{W}$ with computation complexity $O(Dn + D + dlogd)$. Denote the repeated initialization

times of **F** in (16) as T_k, and the number of iterations in the whole algorithm as T_t, then the computational complexity of our algorithm is $O(T_t(T_k(DCN)+DCN+c^2n+Dn+dlogd) \sim O(Dn)$. Next we consider the memory cost of Algorithm 1. Algorithm 1 mainly involves matrices such as **X**, **F**, **G**, etc. $O(Dn+cn+dc)$ is needed for storage. Thus, the calculation cost of our algorithm has a linear relationship with the dimension of the data. According to the above analysis, our algorithm can deal with high-dimensional data well.

2.5. Parameter Determination

Our method mainly involves three important parameters: the reduced dimension d, the balance parameter λ, and the p value of the $l_{2,p}$-norm used in the distance metric. Since the determination of the parameters is still an open problem in the related fields, we use heuristic and empirical methods to determine the parameters.

The first parameter d represents the number of features that can best represent the original data. When d is too large, the representation of the original data is still redundant and the curse of dimension still exists. When d is too small, there may be loss of information so that different clusters cannot be separated. In this paper, by changing the value of d, the parameters with the best accuracy are selected through grid search. The second parameter is the balance parameter λ. Obviously, this parameter balances the effect of dimensionality reduction and clustering on the value of objective function. The larger λ, the greater the impact of clustering is. Following the setting in [27], we search λ in the range of $[10^{-6}, 10^{-4}, 10^{-2}, ..., 10^2, 10^4, 10^6]$. The third parameter p affects the distance between data points in KM, and then influences the clustering results. We adjust the value between 0 and 2. The influence of different parameter values will be

discussed in the experimental section.

3. Experiments

3.1. Data Description and Evaluation Metric

3.1.1. Data Description

We conduct analytical experiments on seven datasets to evaluate the performance. For each dataset, we preprocess all the values by centralization. These datasets include:

UCI datasets [1]: We evaluate our algorithm on four datasets: Cars, Wine, Ionosphere, and Ecoli.

USPS Digit Dataset [2]: The dataset includes 9298 handwritten digital images, all of which are grayscale images of 16 pixels. We select 20% of the dataset for the experiment.

Umist Face Dataset [3]: 575 images in total, corresponding to 20 different people. Each category consists of 19 to 48 images.

COIL-20 Object Dataset [40]: It contains 20 objects and each object has 72 samples taken at pose intervals of five degrees. We first extract LBP features with 3076 dimensions and reduce the dimension to 300 for evaluating the performance of our method.

The detailed description of the aforementioned datasets is displayed in Table 1.

[1] http://archive.ics.uci.edu/ml/datasets.html

[2] https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/multiclass.html

[3] http://images.ee.umist.ac.uk/danny/database.html

Table 1: Summary of the different datasets.

Datasets	Classes (c)	Samples (n)	Total features (D)
Cars	3	392	8
Wine	3	178	13
Ionosphere	2	351	34
Ecoli	8	366	343
USPS	10	1854	256
Umist	20	575	644
COIL-20	20	1440	3076

3.1.2. Evaluation Metric

In order to evaluate the effectiveness of the proposed method, we will compare it with some relevant subspace clustering methods. Meanwhile, in order to express the effect of dimensionality reduction, we will also provide the results of K-means clustering for comparison. The detailed introduction is as follows:

- **KM** represents the traditional K-means algorithm, and its results will be used as the benchmark in the experiment.

- **PCAKM** means that PCA is first used to reduce the dimension of data, and then KM clustering is used for clustering.

- **DEC** [29] is a general discriminant subspace learning framework, which optimizes both PCA and KM simultaneously.

- **TRACK** [36] adopts LDA and KM clustering methods, and uses regularization technique of structured sparse induction criterion to select discriminant features.

- **FAKM** [27] combines feature selection with KM clustering, and uses an adaptive loss function in the objective function.

All the compared methods are implemented in MATLAB (R2016a). The computer processor is Intel(R)Core(TM) i7-7500T CPU @ 2.70GHz, and the memory is 8-GB. We used three indicators of accuracy (ACC), normalized mutual information (NMI) and purity to evaluate the clustering performance of all methods.

Denote g_i as the real label of x_i, q_i as the result of cluster process. Accuracy (ACC) is defined as follow

$$ACC = \frac{\sum_{i=1}^{n} \sigma(g_i, map(q_i))}{n}, \tag{27}$$

where $map(\cdot)$ is a mapping function to obtain the matching between real tags and clustering tags by Kuhn-Munkres algorithm. $\delta(x, y)$ is the Kronecker function

$$\delta(x, y) = \begin{cases} 1, & x = y, \\ 0, & Otherwise. \end{cases} \tag{28}$$

A larger value of accuracy (ACC) indicates a better clustering result.

Denote C as the real classes tag set of the sample, C' as the classes tag set obtained by clustering algorithm. Normalized mutual information (NMI) can be defined by the following formula

$$NMI(C, C') = \frac{MI(C, C')}{\max(H(L), H(C))}, \tag{29}$$

where $H(\cdot)$ represents the entropy. $MI(C, C')$ is the mutual information between C and C', as defined below

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)}, \tag{30}$$

here $p(c_i, c'_j)$ is the probability of a randomly selected sample belongs to both cluster c_i and c'_j. It is easy to observe that the value of normalized mutual information (NMI) is between 0 and 1. Similar to the accuracy rate (ACC), the larger the NMI, the better the clustering result.

Purity is a very simple clustering evaluation method, which is calculated by assigning the labels of a cluster to the most frequent classes. The mathematical definition is as follows

$$purity(C, C') = \frac{1}{N} \sum_{j} \max_{i} |c'_j \cap c_j|, \tag{31}$$

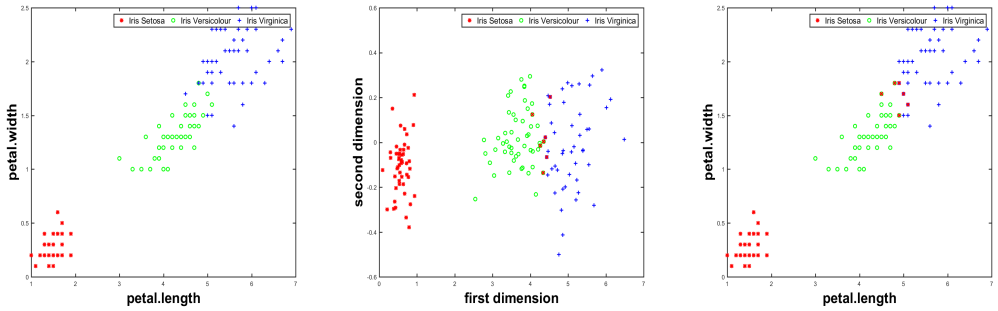
where N represents the total number of samples. Similarly, purity $\in [0, 1]$, the closer the value is to 1, the better the result.

3.2. Toy Example on Iris

To show the visual effectiveness, we first conduct a small experiment on Iris dataset [4]. The dataset consists of three categories (setosa, versicolor and Virginia). The petal length and petal width are chosen for experiment to show a visualization example. DEC is used to compare with our method, and d is set as 2. We first cluster the iris data, and then use the obtained optimal transformation matrix to project the original data into a two-dimensional space. The clustering results are shown in Fig 1, where the samples of the wrong cluster are marked with red 'x'.

As we can see, our method has fewer error markers than DEC. In addition, From Fig. 1.(c), it can be seen that the features selected by our method are consistent with the two features that can distinguish the various types of samples visually, namely, the length and width of petals. From the Fig. 1.(b), we can see

[4] http://archive.ics.uci.edu/ml/datasets/Iris



(a) (b) (c)

Figure 1: Clustering results on the Iris dataset, the dimension is reduced to 2. (a) Original data. (b) Clustering results of DEC. (c) Clustering results of our methods.

Table 2: Comparison of clustering results (ACC%)

Methods	Cars	Wine	Ionosphere	Ecoli	USPS	Umist	COIL-20
KM	44.79 ± 0.13	64.80 ± 6.44	70.75 ± 1.60	55.67 ± 7.69	62.03 ± 3.89	41.67 ± 2.23	62.78 ± 0.04
PCAKM	44.82 ± 0.12	67.64 ± 5.44	71.11 ± 0.14	68.93 ± 6.41	63.91 ± 1.64	42.10 ± 2.32	59.38 ± 3.27
TRACK	45.66 ± 0.00	70.22 ± 0.00	71.88 ± 0.14	63.01 ± 5.42	65.70 ± 0.27	47.97 ± 4.02	54.04 ± 3.08
DEC	47.68 ± 0.08	70.22 ± 0.00	71.23 ± 0.00	62.08 ± 3.85	64.96 ± 0.09	44.54 ± 1.73	$\mathbf{67.74 \pm 2.81}$
FAKM	59.18 ± 0.17	$\mathbf{88.20 \pm 0.00}$	72.31 ± 3.63	69.73 ± 6.11	66.98 ± 3.37	48.43 ± 1.98	67.02 ± 3.33
OURS	$\mathbf{62.50 \pm 1.31}$	$\mathbf{88.20 \pm 0.00}$	$\mathbf{74.93 \pm 0.00}$	$\mathbf{72.05 \pm 2.25}$	$\mathbf{67.49 \pm 2.79}$	$\mathbf{48.54 \pm 3.10}$	67.23 ± 1.98

that DEC has a completely different structure. Therefore, our approach better p-reserves the structure of the original data than that of DEC by selecting the most representative features.

3.3. Comparison of Clustering Results

In this section, we show the clustering results of different methods on different datasets. Grid search is conducted for different parameters according to the above mentioned, and the best combination of parameters is selected to repeat the experiment for 10 times and the average value is taken. The results are shown in

Table 3: Comparison of clustering results (NMI%)

Methods	Cars	Wine	Ionosphere	Ecoli	USPS	Umist	COIL-20
KM	19.35 ± 0.33	41.61 ± 1.49	12.30 ± 2.99	49.09 ± 4.00	61.73 ± 2.67	62.93 ± 2.27	73.28 ± 1.97
PCAKM	19.45 ± 0.32	42.27 ± 1.57	13.01 ± 0.00	57.38 ± 3.51	62.34 ± 0.68	64.07 ± 2.27	71.82 ± 2.09
TRACK	$\mathbf{30.39 \pm 3.78}$	43.56 ± 2.68	13.49 ± 0.48	55.29 ± 6.66	63.60 ± 0.79	64.43 ± 2.27	66.30 ± 1.92
DEC	19.10 ± 0.00	42.87 ± 0.00	13.12 ± 0.00	56.54 ± 2.58	62.90 ± 0.66	65.77 ± 2.27	$\mathbf{75.94 \pm 1.29}$
FAKM	19.10 ± 7.17	$\mathbf{65.69 \pm 0.00}$	12.85 ± 9.72	57.59 ± 1.58	63.60 ± 0.88	$\mathbf{66.84 \pm 2.27}$	75.60 ± 1.63
OURS	$\mathbf{30.39 \pm 0.00}$	$\mathbf{65.69 \pm 0.00}$	$\mathbf{18.86 \pm 0.00}$	$\mathbf{58.52 \pm 1.46}$	$\mathbf{64.08 \pm 1.12}$	66.74 ± 2.27	75.65 ± 1.13

Table 4: Comparison of clustering results (purity%)

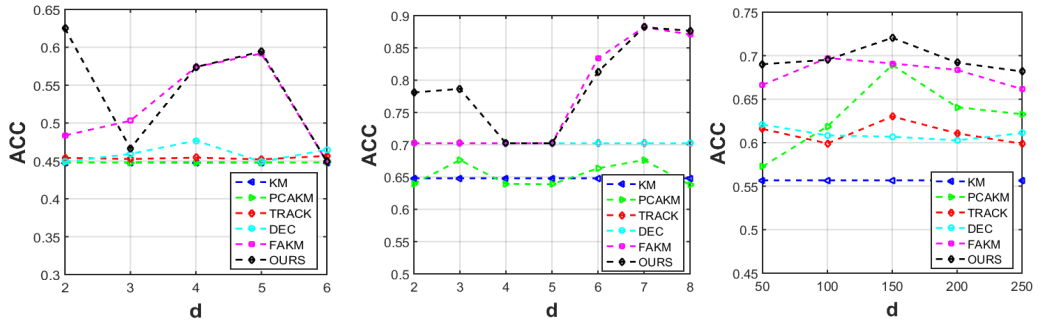
Methods	Cars	Wine	Ionosphere	Ecoli	USPS	Umist	COIL-20
KM	65.05 ± 0.00	69.52 ± 0.84	70.75 ± 1.60	76.60 ± 3.17	70.76 ± 3.22	49.90 ± 2.36	66.06 ± 2.90
PCAKM	65.05 ± 0.00	69.89 ± 1.12	71.11 ± 0.00	80.59 ± 2.89	71.51 ± 1.49	50.63 ± 2.36	63.28 ± 2.82
TRACK	65.03 ± 0.00	70.22 ± 0.00	71.88 ± 0.27	80.86 ± 7.43	73.19 ± 1.54	52.89 ± 2.36	57.78 ± 2.28
DEC	65.05 ± 0.00	70.22 ± 0.00	71.23 ± 0.00	82.17 ± 2.39	72.40 ± 1.58	52.94 ± 2.36	$\mathbf{70.39 \pm 2.57}$
FAKM	67.85 ± 7.17	$\mathbf{88.20 \pm 0.00}$	72.30 ± 6.38	81.69 ± 7.87	73.40 ± 1.80	$\mathbf{56.94 \pm 2.36}$	70.04 ± 2.32
OURS	$\mathbf{69.03 \pm 0.12}$	$\mathbf{88.20 \pm 0.00}$	$\mathbf{75.73 \pm 0.00}$	$\mathbf{82.83 \pm 0.14}$	$\mathbf{73.59 \pm 2.21}$	56.50 ± 2.36	70.24 ± 1.52

Tables 2-4.

From the tables, we can get the following observations:

- Most KM-based subspace clustering algorithms have better performance than KM on each dataset, which shows the effectiveness of this kind of algorithm. Although the NMIs of DEC and FAKM on the Cars dataset are lower than KM, these methods still achieve a smaller gap with KM when the dimension is reduced and the calculation cost of subsequent learning tasks is greatly reduced.

- DEC achieves better results than PCAKM on all datasets except Ecoli because it builds a more general discriminant clustering framework.



(a) (b) (c)

Figure 2: Clustering results (ACC) of the compared methods on the different d. (a) Cars. (b) Wine. (c) Ecoli.

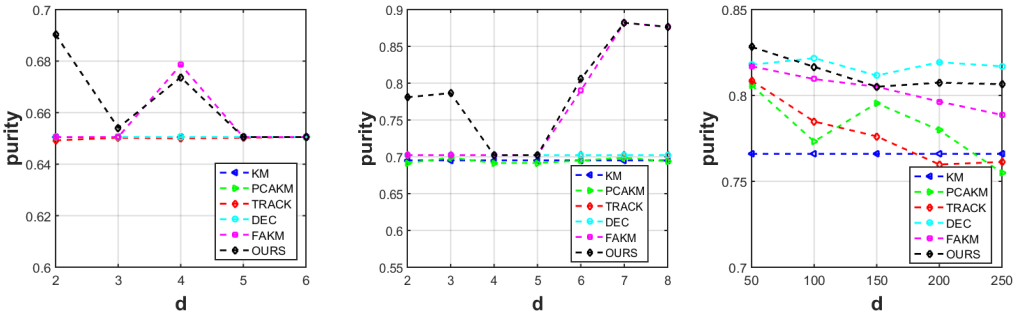
- Compared to DEC, we can see that our method achieves better results on the most of datasets due to the robustness of $l_{2,p}$-norm as a distance metric.

- Compared with TRACK, which also combines feature selection and clustering, our method also has better performance. The reason may be that our method is more flexible in balancing the scatter matrix.

- FAKM defines an adaptive objective function to improve the robustness of the method. In comparison, our method has similar or better results, which indicates that the objective function based on $l_{2,p}$-norm is more robust.

3.4. Impact of Dimension Reduction

In addition, we also study the effect of the reduced dimension d on different datasets by different methods, and the parameter setting is the same as above, each experiment is repeated ten times, and the mean value is recorded. The results are shown in Fig. 2 and Fig. 3.



(a) (b) (c)

Figure 3: Clustering results (purity) of the compared methods on the different d. (a) Cars. (b) Wine. (c) Ecoli.

Through observation, the following conclusions can be drawn:

- Not all the methods can achieve better results when d is increased, which indicates that dimension reduction can effectively improve the performance of clustering.

- When only a small dimension is reserved, the performance of some subspace clustering methods will decline because of the excessive information loss.

- Our method tends to perform better on smaller dimensions than other methods. In addition, the optimal results are usually obtained on the smaller dimensions, which indicates that our method can effectively select the most important features in the data.

- Our method can get the better results in most cases.

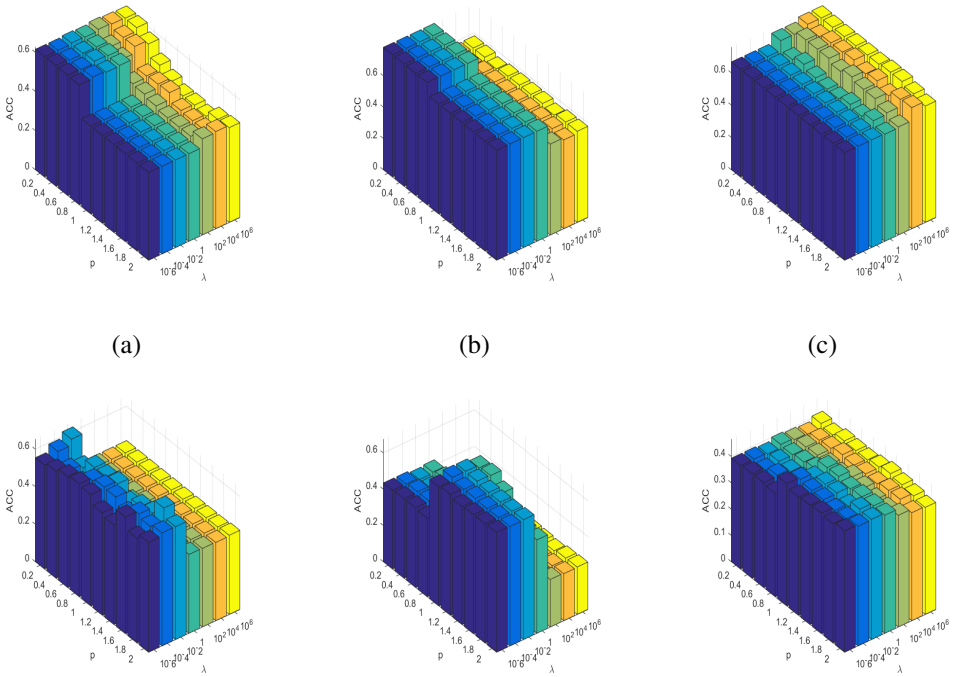


Figure 4: Parameters sensitivity analysis. (a) Cars; (b) Wine; (c) Ionosphere; (d) Ecoli; (e) Usps; (f) Umist

3.5. Parameter Analysis

In order to understand how the parameters λ and p affect the results of the clustering experiment, we fix the value of d in the experiment and carry out the parameters sensitivity experiment. The results are shown in Fig. 4.

As can be seen from Fig. 4, λ and p both have great influence on the final clustering accuracy. Let's first discuss the impact of λ. From the experimental results, we notice that the clustering performance is sensitive to λ. For example, in Fig. 4.(b), i.e., the Wine dataset, the result of $\lambda < 1$ is significantly better than that of $\lambda > 1$. At the same time, it can be found that if we can choose a value close to the λ that get the optimal result, we can get a good result, but it is also affected by the value of parameter p. We can see that the p also affects the result by the different value range. Take Fig. 4.(a) and (b) as examples, when p approximately belongs to $(0, 1)$, the clustering results are better. In Fig. 4.(c) and (e), it is approximately within the range of $(1, 2)$ for higher accuracies. The above observation is very helpful for parameter selection, that is, the parameter value can be approximately determined by finding which range of results are better.

4. Conclusion

In this paper, we propose a flexible subspace clustering model. Specifically, we first incorporate feature selection and K-means clustering into a single framework, which can select the refined features and improve the clustering performance. Second, we embed the $l_{2,p}$-norm into the framework to enhance the robustness and retain the desirable properties from big data. Finally, considering the proposed model is neither convex nor Lipschitz continuous, we develop an effective algorithm to solve it. In addition, we also theoretically prove the convergence

of the proposed algorithm. Experimental results verify the presented method has more robust and better performance on benchmark databases compared to the existing approaches.

It should be noted that the proposed method could obtain more robust results than the existing methods due to the flexibility of selecting p value. However, the proposed model can only choose the parameter p manually for different datasets. Recently, many adaptive learning approaches [41] [42] are successfully used in data mining and pattern recognition. Can the idea be used for our clustering model to adjust the parameter p automatically according to characters of different datasets? If the answer if yes, how to design the adaptive scheme? Our future work will focus on the topic.

References

[1] Yao D, Yu C, Yang L T, Jin H. Using Crowdsourcing to Provide QoS for Mobile Cloud Computing. IEEE Transactions on Cloud Computing, 2019, 7(2): 344-356.

[2] Sakr S, Elgammal A. Towards a Comprehensive Data Analytics Framework for Smart Healthcare Services. Big Data Research, 2016, 4: 44-58.

[3] Hendre A, Joshi K P. A Semantic Approach to Cloud Security and Compliance. International Conference on Cloud Computing, 2015: 1081-1084.

[4] Ren L, Meng Z, Wang X, Zhang L, Yang L T , A Data-driven Approach of Product Quality Prediction for Complex Production Systems, IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2020.3001054, 2020.

[5] Ren L, Meng Z, Wang X, Luan R, Yang L T, A Wide-Deep-Sequence Model based Quality Prediction Method in Industrial Process Analysis, IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2020.3001602, 2020.

[6] Li Z, Chen R, Liu L, Min G. Dynamic Resource Discovery Based on Preference and Movement Pattern Similarity for Large-Scale Social Internet of Things. IEEE Internet of Things Journal, 2016, 3(4): 581-589.

[7] Wang X, Yang L T, Wang Y, Ren L, Deen M J. ADTT: A Highly-Efficient Distributed Tensor-Train Decomposition Method for IIoT Big Data. IEEE Transactions on Industrial Informatics, doi: 10.1109/tii.2020.2967768, 2020.

[8] Li G, Wu H, Jiang G, Xu S, Liu H. Dynamic Gesture Recognition in the Internet of Things. IEEE Access, 2019, 7: 23713-23724.

[9] Wang X, Yang L T, Song L, Wang H, Ren L, Deen M J. A Tensor-based Multi-Attributes Visual Feature Recognition Method for Industrial Intelligence. IEEE Transactions on Industrial Informatics, doi: 10.1109/TI-I.2020.2999901, 2020.

[10] Wang X, Wang Y, Zhe H, Juan D. The Research on Resource Scheduling Based on Fuzzy Clustering in Cloud Computing. International Conference on Intelligent Computation Technology and Automation, 2015: 1025-1028.

[11] Zhang X, Meng F, Xu J. PerfInsight: A Robust Clustering-Based Abnormal Behavior Detection System for Large-Scale Cloud. International Conference on Cloud Computing, 2018:896-899.

[12] Estiri H , Omran B A , Murphy S N . kluster : An Efficient Scalable Procedure for Approximating the Number of Clusters in Unsupervised Learning. Big Data Research, 2018, 13: 38-51.

[13] Zhang Q, Yang L T, Chen Z, Li P. High-order possibilistic c-means algorithms based on tensor decompositions for big data in IoT. Information Fusion, 2018, 39: 72-80.

[14] Hodge V J, Austin J. A Survey of Outlier Detection Methodologies. Artificial Intelligence Review, 2004, 22(2): 85-126.

[15] Hathaway R J, Bezdek J C, Hu Y. Generalized fuzzy c-means clustering strategies using L/sub p/ norm distances. IEEE Transactions on Fuzzy Systems, 2000, 8(5): 576-582.

[16] Salem S B, Naouali S, Chtourou Z, et al. A fast and effective partitional clustering algorithm for large categorical datasets using a k-means based approach. Computers & Electrical Engineering, 2018, 68: 463-483.

[17] Cai X, Nie F, Huang H. Multi-view K-means clustering on big data. International Joint Conference on Artificial Intelligence, 2013: 2598-2604.

[18] Liang D, Peng Z, Lei S, Wang H, Fan M, Wang W, Shen Y. Robust multiple kernel K-means using $\ell_{2,p}$ norm. International Joint Conference on Artificial Intelligence, 2015, 3476-3482.

[19] Chang X, Nie F, Wang S, Yang Y, Zhou X, Zhang C. Compound rank- k projections for bilinear analysis. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(7): 1502-1513.

[20] Jolliffe, IT (1986). Principal Component Analysis. New York: Springer-Verlag.

[21] Duda, Richard O, Hart, Peter E, Stork, David G. Pattern Classification (2nd Edition). John Wiley, 2000.

[22] Yu Y F, Ren C X, Jiang M, et al. Sparse approximation to discriminant projection learning and application to image classification. Pattern Recognition, 2019, 96: 1-10.

[23] Roweis S T and Saul L K, Nonlinear dimensionality reduction by locally linear embedding, Science, 2000, vol.290, no.5500, pp.2323-2326.

[24] Hou C, Nie F, Jiao Y, Zhang C, Wu Y. Learning a subspace for clustering via pattern shrinking. Information Processing and Management, 2013, 49(4): 871-883.

[25] Yin X, Chen S, Hu E. Regularized soft K-means for discriminant analysis. Neurocomputing, 2013, 103: 29-42.

[26] Wang X, Chen R, Hong C, Zhang Z. Unsupervised feature analysis with sparse adaptive learning. Pattern Recognition Letters, 2018, 102: 89-94.

[27] Wang X, Chen R, Yan F, et al. Fast adaptive K-means subspace clustering for high-dimensional data. IEEE Access, 2019, 7: 42639-42651.

[28] Ding C, Li T. Adaptive dimension reduction using discriminant analysis and K -means clustering. International Conference on Machine Learning, 2007: 521-528.

[29] Hou C, Nie F, Yi D, Tao D. Discriminative embedded clustering: A framework for grouping high-dimensional data. IEEE Transactions on Neural Networks and Learning Systems, 2015, 26(6): 1287-1299.

[30] Li H, Jiang T, Zhang K. Efficient and robust feature extraction by maximum margin criterion. IEEE Transactions on Neural Networks and Learning Systems, 2006, 17(1): 157-165.

[31] Park H, Jeon M, Rosen J B, et al. Lower dimensional representation of text data based on centroids and least squares. Bit Numerical Mathematics, 2003, 43(2): 427-448.

[32] Nie F, Xiang S, Liu Y, Hou C, Zhang C. Orthogonal vs. uncorrelated least squares discriminant analysis for feature extraction. Pattern Recognition Letters, 2012, 33(5): 485-491.

[33] Yu Y F, Xu G, Huang K K, Zhu H, Chen L, Wang H. Dual Calibration Mechanism based $L_{2,p}$-Norm for Graph Matching. IEEE Transactions on Circuits and Systems for Video Technology, doi: 10.1109/TCSVT.2020.3023781.

[34] Wang Q, Gao Q, Gao X, Nie F. $\ell_{2,p}$ -Norm Based PCA for Image Recognition. IEEE Transactions on Image Processing, 2018, 27(3): 1336-1346.

[35] Tao H, Hou C, Nie F, Jiao Y, Yi D. Effective discriminative feature selection with nontrivial solution. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(4): 796-808.

[36] Wang D, Nie F, Huang H. Unsupervised feature selection via unified trace

ratio formulation and K -means clustering (TRACK). European Conference on Machine Learning, 2014, 306-321.

[37] Nguyen B, De Baets B. Kernel-based distance metric learning for supervised k-means clustering. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(10): 3084-3095.

[38] Peng J, Wei Y. Approximating K-means-type clustering via semidefinite programming. Siam Journal on Optimization, 2007, 18(1): 186-205.

[39] Luo M, Nie F, Chang X, Yang Y, Hauptmann A. G. and Zheng Q. Adaptive unsupervised feature selection with structure regularization. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(4): 944-956.

[40] Yu K, Zhang T, Gong Y. Nonlinear Learning using Local Coordinate Coding. Advances in Neural Information Processing Systems, 2009, 2223-2231.

[41] Ren C X, Liang B H, Ge P, Zhai Y M, Lei Z. Domain Adaptive Person Re-Identification via Camera Style Generation and Label Propagation. IEEE Transactions on Information Forensics and Security, 2020, 15: 1290-1302.

[42] Yan F , Wang X D , Zeng Z Q , et al. Adaptive Multi-view Subspace Clustering for High-dimensional Data. Pattern Recognition Letters, 2019, 130: 299-305.