ADAPTABLE SUBSPACE CLUSTERING COMBINE APPROPRIATE SELECTION AND K-MEANS CLUSTERING FRAMEWORK

J.MADHAVI
ASST PROFESSOR ,CSE
MAHAVEER INSTITUTE OF SCIENCE & TECHNOLOGY
jajala.madhavi@gmail.com

K GIRI BABU
ASST PROFESSOR ,CSE
MAHAVEER INSTITUTE OF SCIENCE & TECHNOLOGY
girik019@gmail.com

ABSTRACT

Subspace clustering is a technique which finds groups within different subspaces (a selection of one or more dimensions). suitable to the nonexistence of class labels, unsupervised feature selection is much more complicated than supervised feature selection. concerning as an important computing prototype, cloud computing is to address big and distributed databases and rather simple computation. In this model, data mining is one of the most important and fundamental problems. A huge amount of data is generated by sensors and other smart devices. Data mining for these big data is essential in various applications. K-means clustering is a difficult 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 containing a huge amount of noise, outliers and unnecessary features. A variation of K-means clustering algorithm, namely, adaptable subspace clustering, incorporates feature selection and K-means clustering into a integrated framework, which can select the advanced features and improve the clustering performance. tentative results verify the presented method has more robust and better performance on standard databases compared to the existing approaches.

I INTRODUCTION

In order to explain a new model for data exploitation, the locution "Big Data" has recently been emerging. These technologies are new in the field of Information Technology, tend to emerge very often and with a massive publicity, at the end it takes some time to be recognized. Big Data (also known as BD) is different in numerous ways such as volume (too big), velocity (faster arrival) variability (quick changes), veracity (much commotion), and variety (diversity). Using orthodox propositions and procedures this Big Data is processed in partial arrangements. Even the technologies introduced to support Big Data contain different variety of presentations, which eventually make it tough to stimulate the creation of tools and applications to help include data from many sources. This analysis hence identifies possible areas for uniformity within the Big Data technology vastness. Complicated and massive datasets have various types of different and important features that are closely in similarity with "Big Data". To administer these datasets is troublesome with the conventional information preparing frameworks. in addition, data storage, data transition, data visualization, data penetrating, data analysis, data security,

data privacy violations and sharing propose different rising challenges that the "Big Data" reinforce. To potentially grasp the supplementary information sets the Big Data is appropriate as a observation that highlights the deficiency of ability of usual information structures. The emergence of Big Data model transpire when the compute of the data is either in take it easy .it forcedly induces the management of data in the system engineering design to become a important driver. Basically the Big Data Model represents a paradigm shift in the data infrastructures i.e. from substantial systems with at a 90 degree angle mount into a parallel mounted system that coalesce an unbounded connected set of reserves. This change from perpendicular to parallel predicates some different problems in some dissimilar areas such as information deliverance, information orchestrating, and inactivity in the consistency across schematics, stack stabilizing, and process deficiencies and their interdependencies on single hand. On other hand, the Big Data model uses different contraptions to provide the clamber in data handling, but embodies the same shift again. The reason for this move is to bargain out codes and information across over inexactly coupled assets and match the scaling in information. As to produce additional knowledge about the data a different purpose of residing and retrieve huge amounts of data is to execute analysis. In the olden days, the assay was usually attained on an undirected sample of the data. The word "Big Data" contains assortment of distinctiveness, it is used in various contexts. To identify with where thought will appropriately assist backing the big data model, in order to find what the term really means we have to extend our knowledge to some extent of consonance. The "Big Data" is a gathering of information with unique brilliance (e.g. capacity, momentum, array, range, precision, etc.) that for a problem realm at any given moment can't be expertly handled using current advancements and strategies with a specific end goal to concentrate esteem. The above definition recognizes of Big Data from business knowledge and predictable value based administration while suggesting an expansive range of utilizations that incorporates them.

A definitive objective of handling Big Data is to get separated esteem that can be trusted (in light of the fact that the central information can be trusted). This is done through the usage of higher examination next to the entire evaluate of data disregarding scale. Parsing these target edges the regard exchange for massive Information utilizes cases.

Formulation

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

$\min_{F,Z} \sum_{i=1}^{n} \sum_{k=1}^{c} F_{ik} ||X_i \quad Z_k|||_2^2$ (1)

----------(1)

s.t F∈ $\{0,1\}^{n \times c}$, F_1=1

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

$\min_{F,Z} ||X - ZF^T||_F^2$,

s.t F∈ $\{0,1\}^{n \times c}$, F_1=1 (2)

Considering that the high-dimensional data could contain a huge amount of noises, outliers and unnecessary features. It leads to high computational complexity and performance degradation. The direct idea is to find a transformation matrix W ∈ RD×d which transforms the high-dimensional features to a low-dimensional feature space Y = WT X, where Y = [y1,⋯ , yn] ∈ Rd×n. Following the feature selection, we use the column vectors wi as follow

$W_i = [0, \ldots, 0, 1, 0 \ldots, 0]^T$ ----------(3)

Then the feature selection matrix W can be represented as

W = [wI(1), wI(2), ..., wI(d)]----------(4)

where I is a permutation of {1, 2,..., D}. It can be seen that the transformation matrix 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

$\max \mathrm{Tr}_{W,G,F}(W^T S_t W) - \lambda ||W^T X - GF^T||_2^P$

s.t W∈ $\{0,1\}^{D \times d}$,rank(W)=d,W^T1=1

F∈ $\{0,1\}^{n \times c}$, F_1=1 ……(5)

Here $S_t = \sum_{i=0}^{n} X_i X_i^T$ is the total scatter matrix. G = [g1,⋯ , gc] ∈ R^{d*c} is c centroid vectors in the low-dimensional space. It should be noted that is also a joint model of subspace learning and clustering. It uses the pattern of $||M|| \sum_i \left(\frac{(1+\sigma)||m_{i||_2^2})}{||m_i||^{2+\sigma}} \right)$ to construct the K-means clustering, here M is an arbitrary matrix, mi is the i-th column and σ is a parameter. Different from FAKM, the model in (5) has more robust performance since it adopts l2,p - norm to construct the K-means clustering and can flexibly choose appropriate p according to the different data. means clustering, here M is an arbitrary matrix, mi is the i-th column and σ is a parameter. Different from FAKM, the

model in (5) has more robust performance since it adopts l2,p - norm to construct the K-means clustering and can flexibly choose appropriate p according to the different data.

2. Optimization

Since our objective function (5) involves l2,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 l2,p -norm. Similar techniques are used in [35] to solve the problem of the minimization of LDA with regular term based on l2,p -norm ($0 < p \leq 2$). Inspired by these papers, we propose an effective iterative algorithm to solve our objective function.

Let $d_i=\frac{p}{2}||W^Tx_i-Gf_i||_2^{p-2}$, then (5) can be transformed to

$\max Tr_{W,G,F}(W^TS_tW)-2\lambda/p \sum_{i=1}^{n} d_i||W^Tx_i-Gf_i||_2^2$ (6)

Since λ is an arbitrary constant, for convenience, we will still mark 2λ/p as λ. Denote as a diagonal matrix with its i-th diagonal element as di , and U = [u1,u2,...,un] = $W^T X_i – GF^T$, where ui $\in R^d$ is the i-th column of U. We have

$max_{W,G,F,\Delta}(W^TS_tW)-\lambda Tr(U^T\Delta U)$---------(7)

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

Step1. Fixing W,G and and Optimizing F

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

$\min_F \sum_{i=1}^{n} d\ ||W^Tx_i-Gf_i||_2^2$

$=\min_F \sum_{i=1}^{n}\sum_{k=1}^{c} F_{ik}||W^Tx_i-G||_2^2F_{ik}$.------(8)

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

$F_{ij}=\begin{cases}\mathbf{1}, & \boldsymbol{j = arg\ min_k}\ ||W^Tx_i-g_k||_2^2 \\ \mathbf{0}, & \boldsymbol{otherwise}\end{cases}$ ----------(9)

Step 2. Fixing Δ and F and Optimizing W and G

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

$C(W,G)= Tr(W^TS_tW)-\lambda Tr(U^T\Delta U)$ (10)

Where $U= (W^Tx-GF^T)$

We take a derivative of $C(W,G)$ over G

$\frac{\partial C(W,G)}{\partial G} = -\lambda\ \partial Tr((W^Tx-GF^T)^T\Delta(W^Tx-GF^T)) / \partial G,$

$= -\lambda\ \partial Tr((GF^T-WTx)^T\Delta(GF^T- W^Tx)) / \partial G,$ (11)

$= -\lambda\ \partial Tr((GF^T\Delta FG^T)^T-2Tr(GF^T X^TW)) / \partial G,$

$= -2\ \lambda\ (GF^T\Delta F -W^TX\Delta F)$

Let the above equation equal to zero and we have

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

Substituting G into $C(W,G)$ we have

$C(W)= Tr(W^{T}S_{t}W) - \lambda Tr((W^{T}x-GF^{T)T}\Delta(W^{T}x-GF^{T))}$

$= Tr(W^{T}S_{t}W) -- \lambda Tr((W^{T}x\Delta X^{T}W - W^{T}XF(F^{T}\Delta F)^{-1}F^{T}\Delta^{T}X^{T}W$ (13)

$= Tr(W^{T}(S_{t}- \lambda X\Lambda X^{T}+ \lambda X\Delta F (F^{T}\Delta F)^{-1} F^{T}\Delta X^{T})W),$

$=Tr(W^{T}MW),$

WHERE $M= S_{t}- \lambda X\Lambda X^{T}+ \lambda X\Delta F (F^{T}\Delta F)^{-1} F^{T}\Delta X^{T})W$

therefore the problem to optimize W becomes

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

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

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

$$d_{i} = \frac{p}{2}\|\mathbf{W}^{T}\mathbf{x}_{i} - \mathbf{G}\mathbf{f}_{i}\|_{2}^{p-2}.$$

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 F is fixed, the algorithm can quickly adjust W and G to adapt to the F. In other words, when we need to update the F in the next step, the optimal 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 and is employed. In each step of updating F, we will randomly initialize F several times. If the value of the objective function $\|\mathbf{W}^{T}\mathbf{X} - \mathbf{G}\mathbf{F}^{T}\|\mathbf{2}^{F}$ is smaller than that of the previous F, then updating F according to the random initialization. Otherwise, updating 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 $\mathbf{F}^{1}_{i+1}$, $\mathbf{F}^{2}_{i+1}$, ..., $\mathbf{F}^{t}_{i+1}$ by random initialization, where t is the number of random initialization. We update F according to the following rules

$$\mathbf{F}^{*}_{i+1} = \begin{cases} \mathbf{F}^{j}_{i+1}, & \|(\mathbf{W}^{*}_{i})^{T}\mathbf{X}-\mathbf{G}^{*}_{i}(\mathbf{F}^{j}_{i+1})^{T}\|^{2}_{F} \\ & <\|(W^{*}_{i})^{T}X-G^{*}_{i}(F^{j}_{i+1})^{T}\|^{2}_{F} \\ F^{*}, & \text{otherwise} \end{cases} \quad \text{--------} \quad (16)$$

where F^{*} is defined as

$$F_{ij}=\begin{cases} 1, & j = arg\min_{k} ||_{W^{T}X_{i}-g_{k}}||_{2}^{2} \\ 0, & otherwise \end{cases} \text{------}$$

------(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 $X \in R^{D\times n}$, the reduced dimension number d, the number of clusters c, regularization parameter λ, and the distance metric parameter p.

Output: Transformation matrix W, cluster indicator matrix F, and cluster centroid matrix G.

1: Initialize Δ as identity matrix, and randomly initialize W and G.

2: **while** Not convergent do

3: Update F by (16);

4: Update G by (12);

5: Update W by locating the d largest diagonal elements of the matrix M in (14);

6: Update Δ by calculating its diagonal elements by $d_i = p/2||W^T xi - _{Gfi||2}{}^{p-2;}$

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 . For any nonzero vectors e^{t+1}, $e^t \in R^m$, when $0 < p \leq 2$, we have:

$$\frac{\|e^{t+1}\|_2^p}{\|e^t\|_2^p} - \frac{p}{2}\frac{\|e^{t+1}\|_2^2}{\|e^t\|_2^2} - 1 + \frac{p}{2} \leq 0. \quad (18)$$

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

Proof. When W,G and Δ are fixed, optimizing the problem in (7) is equal to solving the traditional K-means on $W^T X$ with fixed centroid. Thus the optimized solution is unique. According to (3), W_i is a vector with only one element being 1 and the rest being 0. Obviously, the derived W by locating the d largest diagonal elements of St $- \Lambda x\delta x^T + \lambda X\Delta F(F^T\Delta F)^{-1}F^T\Delta X^T$ maximizes the objective function in (14). When F and Δ are fixed, G is dependent on W, and the global solution of 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 W_t, G_t in the t-th iteration. In the (t + 1)-th iteration, we fix W_t, G_t and Δ_t, and get the optimized F_{t+1} by (16). According to Theorem 1 and the updating rule in (9), we have

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

Then we fix Δ_t and F_{t+1}, and update G and 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

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

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

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

which can be wrote as

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

According to Lemma 1, we have

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

which holds for each index i, thus we have

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

that is

$$-\sum_i \|\mathbf{u}_i^t\|_2^p + \frac{p}{2}\sum_i \frac{\|\mathbf{u}_i^t\|_2^2}{\|\mathbf{u}_i^t\|_2^{2-p}} \le -\sum_i \|\mathbf{u}_i^{t+1}\|_2^p + \frac{p}{2}\sum_i \frac{\|\mathbf{u}_i^{t+1}\|_2^2}{\|\mathbf{u}_i^t\|_2^{2-p}} \quad (25)$$

Combining (22) and (25), we have

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

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 $Tr(XX^T)$, Algorithm 1 will monotonically increase the objective function until it converges.

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 $O(dcn+C^2n)$ and computing matrix M's diagonal elements to optimize 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 Tt, then the computational complexity of our algorithm is $O(Tt(Tk(DCN) + DCN + C^2n + Dn + dlogd) \sim O(Dn)$.

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.

Table 1

Summary of the different datasets

Datasets	**Classes(c)**	**Samples(n)**	**Total features(D)**
Cars	3	392	8
Wine	3	178	13
Lonospher e	2	351	34

Ecoli	8	366	343
Usps	10	1854	256
Umist	20	575	644
Coil-20	**20**	**1440**	**3076**

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 l2,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 , 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 datasets1: We evaluate our algorithm on four datasets: Cars, Wine, Ionosphere, and Ecoli.

USPS Digit Dataset2: 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 Dataset3: 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.

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 discriminate 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 The computer processor is Intel® Core™ i7-7500T CPU 2.70 GHz, 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

$$ACC=\frac{\sum_{i=1}^{n}\sigma(gi,map(qi))}{n},$$

(27)

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

$$\boldsymbol{\delta(x,y)=\begin{cases}1, & x=y\\ 0, & otherwise\end{cases}}$$

(28)

A larger value of accuracy (ACC) indicates a better clustering result. Denote C as the real classes tag set of the sample, Cas the classes tag set obtained by clustering algorithm. Normalized mutual information (NMI) can be defined by the following formula

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

(29)

where H(·) represents the entropy. M I(C, C') is the mutual information between C and C' , as defined below

$$MI(C, C') \sum_{ci\epsilon C, c'j\epsilon C} p\,(ci, c'j).\log 2p\,\frac{p(ci, c'j)}{p(ci)p(c'j)'}$$

(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

$$\text{Purity}\quad (C,C')=\frac{1}{N}\sum_j \max\ |C'j,\cap Cj|$$ ------ (31)

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

$\max_i|C'_{j\cap} C_j|$

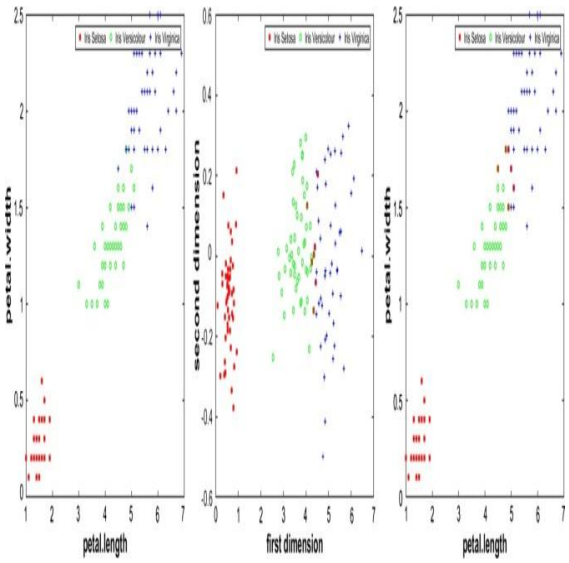


Fig. 1. Clustering results on the Iris dataset, the dimension is reduced to 2. (a) Original data. (b) Clustering results of DEC. (c) Clustering results of our methods. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 2 Comparison of clustering results (ACC %)

Methods	Cars	Wine	Ionosphere	Ecoli	Usps	Umist	Coil
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	67.74 ± 2.81
FAKM	59.18 ± 0.17	88.20 ± 0.00	72.31 ± 3.63	69.73 ± 6.11	66.98 ± 3.37	48.43 ± 1.98	67.02 ± 3.33
OURS	62.50 ± 1.31	88.20 ± 0.00	74.93 ± 0.00	72.05 ± 2.25	67.49 ± 2.79	48.54 ± 3.10	67.23 ± 1.98

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 that DEC has a completely different structure. Therefore, our approach better preserves 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 Tables2-4.

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 FAKM on the Cars dataset are lower than KM, these meth-ods 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.

–Compared to DEC, we can see that our method achieves better results on the most of datasets due to the robustness of $l2,p$-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 sim-ilar or better results, which indicates that the objective func-tion based on $l2,p$-

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. 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

Table3

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	**30.39 ± 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	75.94 ± 1.29
FAKM	19.10 ± 7.17	65.69 ± 0.00	12.85 ±9.72	57.59 ± 1.58	63.60 ± 0.88	66.84 ± 2.27	75.60 ± 1.63
OURS	30.39 ± 0.00	65.69 ± 0.00	18.86 ± 0.00	58.52 ± 1.46	64.08 ± 1.12	66.74 ± 2.27	75.65 ± 1.13

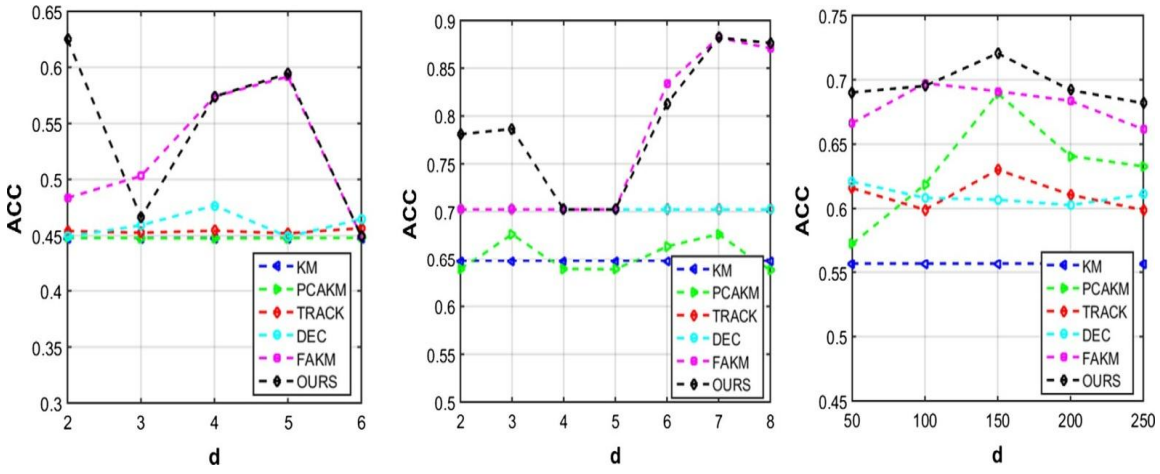
Table4

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	70.39 ±2.57
FAKM	67.85 ±7.17	88.20 ±0.00	72.30 ±6.38	81.69 ±7.87	73.40±1.80	56.94 ±2.36	70.04±2.32
OURS	69.03 ±0.12	88.20 ±0.00	75.73 ±0.00	82.83 ±0.14	73.59 ±2.21	56.50 ±2.36	70.24 ±1.52

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 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. 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



(a) (b) (c)

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

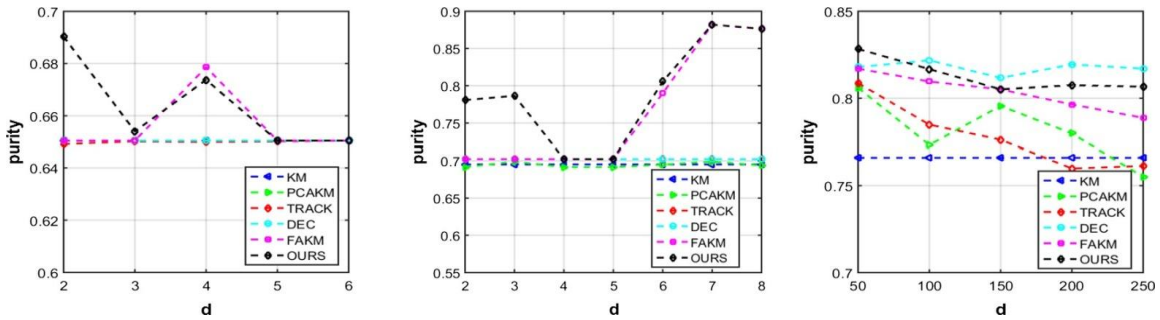


Fig. 3. Clustering results (purity) of the compared methods on the different *d*. (a) Cars. (b) Wine. (c) Ecoli

(a) (b) (c)

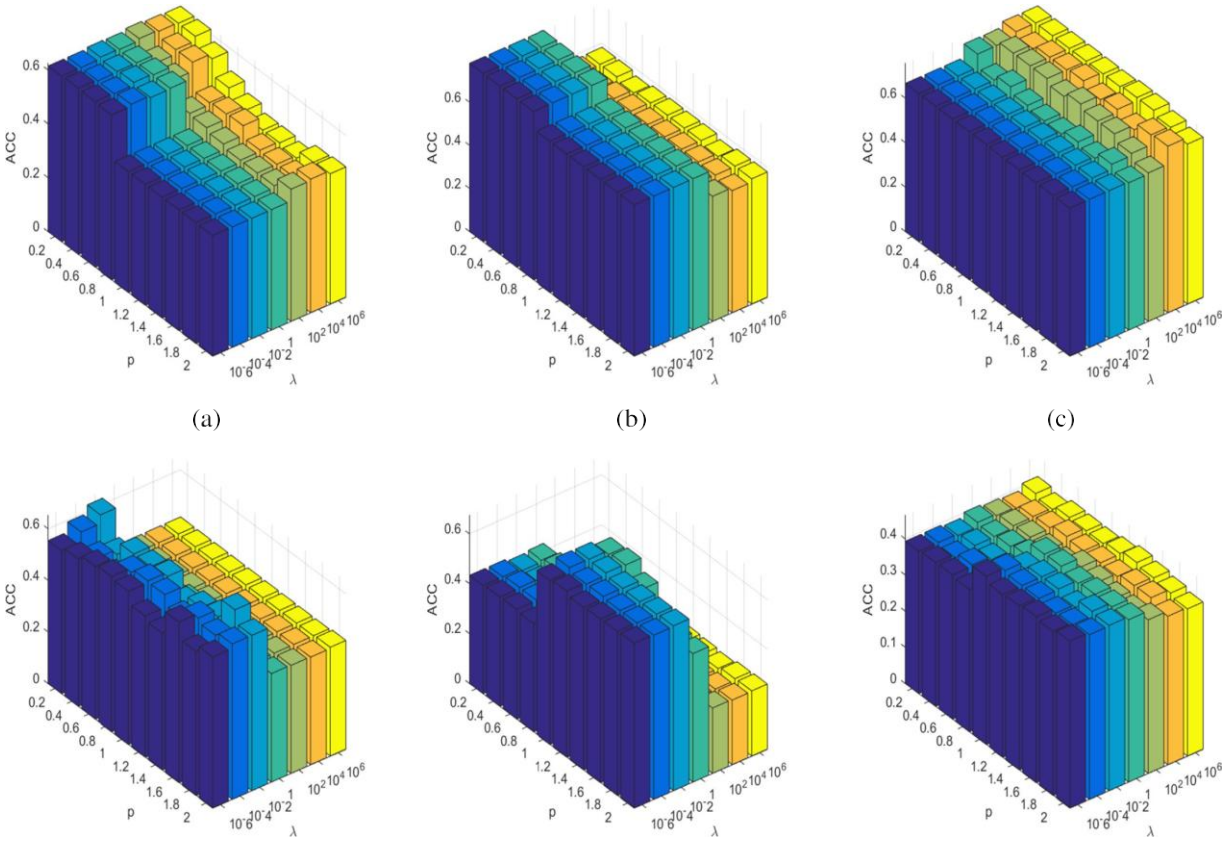


Fig. 4. Parameters sensitivity analysis. (a) Cars; (b) Wine; (c) Ionosphere; (d) Ecoli; (e) Usps; (f)Umist.

4. Conclusion

In this paper, we propose adaptable subspace clustering model. particularly, we first incorporate feature selection and K-means clustering into a single framework, which can select the advanced features and improve the clustering performance. Second, we embed the l2, p-norm into the framework to enhance the robustness and preserve the desirable properties from big data. At last, considering the proposed model is neither convex nor Lipchitz continuous, we develop an effective algorithm to solve it. In addition, we also theoretically prove the convergence of the proposed method. Experimental results verify the obtainable method has more robust and good performance on standard databases compared to the existing method.

It should be recognized that the proposed model 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 are successfully used in data mining and pattern recognition.

References

1. *D. Yao, C. Yu, L.T. Yang, H. Jin, Using crowdsourcing to provide QoS for mobile cloud computing, IEEE Trans. Cloud Comput. 7 (2) (2019) 344–356.*
2. *S. Sakr, A. Elgammal, Towards a comprehensive data analytics framework for smart healthcare services, Big Data Res. 4 (2016) 44–58.*
3. *A. Hendre, K.P. Joshi, A semantic approach to cloud security and compliance, in: International Conference on Cloud Computing, 2015, pp. 1081–1084.*
4. *L. Ren, Z. Meng, X. Wang, L. Zhang, L.T. Yang, A data-driven approach of prod- uct quality prediction for complex production systems, IEEE Trans. Ind. Inform. (2020), https://doi.org/10.1109/TII.2020.3001054.*
5. *L. Ren, Z. Meng, X. Wang, R. Luan, L.T. Yang, A wide-deep-sequence model based quality prediction method in industrial process analysis, IEEE Trans. Neu- ral Netw. Learn. Syst. (2020), https://doi.org/10.1109/TNNLS.2020.3001602.*
6. *Z. Li, R. Chen, L. Liu, Min G. Dynamic, Resource discovery based on preference and movement pattern similarity for large-scale social internet of things, IEEE Int. Things J. 3 (4) (2016) 581–589.*
7. *X. Wang, L.T. Yang, Y. Wang, L. Ren, M.J. Deen, ADTT: a highly-efficient distributed tensor-train decomposition method for IIoT big data, IEEE Trans. Ind. Inform. (2020), https://doi.org/10.1109/tii.2020.2967768.*
8. *G. Li, H. Wu, G. Jiang, S. Xu, H. Liu, Dynamic gesture recognition in the internet of things, IEEE Access 7 (2019) 23713–23724.*
9. *X. Wang, L.T. Yang, L. Song, H. Wang, L.*

Ren, M.J. Deen, A tensor-based multi-attributes visual feature recognition method for industrial intelligence, IEEE Trans. Ind. Inform. (2020), https://doi.org/10.1109/TII.2020.2999901.

10. *X. Wang, Y. Wang, H. Zhe, D. Juan, The research on resource scheduling based on fuzzy clustering in cloud computing, in: International Conference on Intel- ligent Computation Technology and Automation, 2015, pp. 1025–1028.*
11. *X. Zhang, F. Meng, Xu J. PerfInsight, A robust clustering-based abnormal be-havior detection system for large-scale cloud, in: International Conference on Cloud Computing, 2018, pp. 896–899.*
12. *H. Estiri, B.A. Omran, S.N. Murphy, Kluster: an efficient scalable procedure for approximating the number of clusters in unsupervised learning, Big Data Res. 13 (2018) 38–51.*
13. *Q. Zhang, L.T. Yang, Z. Chen, P. Li, High-order possibilistic c-means algorithms based on tensor decompositions for big data in IoT, Inf. Fusion 39 (2018) 72–80.*
14. *V.J. Hodge, J. Austin, A survey of outlier detection methodologies, Artif. Intell. Rev. 22 (2) (2004) 85–126.*
15. *R.J. Hathaway, J.C. Bezdek, Y. Hu, Generalized fuzzy c-means clustering strate- gies using L_p norm distances, IEEE Trans. Fuzzy Syst. 8 (5) (2000) 576–582.*
16. *S.B. Salem, S. Naouali, Z. Chtourou, et al., A fast and effective partitional clus- tering algorithm for large categorical datasets using a k-means based approach, Comput. Electr. Eng. 68 (2018) 463–483.*
17. *X. Cai, F. Nie, H. Huang, Multi-view K-means clustering on big data, in: International Joint Conference on Artificial Intelligence, 2013, pp. 2598–2604.*
18. *D. Liang, Z. Peng, S. Lei, H. Wang, M. Fan, W. Wang, Y. Shen, Robust multiple kernel K-means using 4_2,p norm, in: International Joint Conference on Artificial Intelligence, 2015, pp. 3476–3482.*
19. *X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, C. Zhang, Compound rank-k projec- tions for bilinear analysis, IEEE Trans. Neural Netw. Learn. Syst. 27 (7) (2016) 1502–1513.*