



A feature group weighting method for subspace clustering of high-dimensional data

Xiaojun Chen^a, Yunming Ye^{a,*}, Xiaofei Xu^b, Joshua Zhexue Huang^c

^a Shenzhen Graduate School, Harbin Institute of Technology, China

^b Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin, China

^c Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

ARTICLE INFO

Article history:

Received 7 June 2010

Received in revised form

23 June 2011

Accepted 28 June 2011

Available online 6 July 2011

Keywords:

Data mining

Subspace clustering

k-Means

Feature weighting

High-dimensional data analysis

ABSTRACT

This paper proposes a new method to weight subspaces in feature groups and individual features for clustering high-dimensional data. In this method, the features of high-dimensional data are divided into feature groups, based on their natural characteristics. Two types of weights are introduced to the clustering process to simultaneously identify the importance of feature groups and individual features in each cluster. A new optimization model is given to define the optimization process and a new clustering algorithm FG-*k*-means is proposed to optimize the optimization model. The new algorithm is an extension to *k*-means by adding two additional steps to automatically calculate the two types of subspace weights. A new data generation method is presented to generate high-dimensional data with clusters in subspaces of both feature groups and individual features. Experimental results on synthetic and real-life data have shown that the FG-*k*-means algorithm significantly outperformed four *k*-means type algorithms, i.e., *k*-means, W-*k*-means, LAC and EWKM in almost all experiments. The new algorithm is robust to noise and missing values which commonly exist in high-dimensional data.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

The trend we see with data for the past decade is towards more observations and high dimensions [1]. Large high-dimensional data are usually sparse and contain many classes/clusters. For example, large text data in the vector space model often contains many classes of documents represented in thousands of terms. It has become a rule rather than the exception that clusters in high-dimensional data occur in subspaces of data, so subspace clustering methods are required in high-dimensional data clustering. Many subspace clustering algorithms have been proposed to handle high-dimensional data, aiming at finding clusters from subspaces of data, instead of the entire data space [2,3]. They can be classified into two categories: *hard subspace clustering* that determines the exact subspaces where the clusters are found [4–10], and *soft subspace clustering* that assigns weights to features, to discover clusters from subspaces of the features with large weights [11–24].

Many high-dimensional data sets are the results of integration of measurements on observations from different perspectives so that the features of different measurements can be grouped. For example, the features of the nucleated blood cell data [25] were divided into groups of density, geometry, “color” and texture, each

representing one set of particular measurements on the nucleated blood cells. In a banking customer data set, features can be divided into a demographic group representing demographic information of customers, an account group showing the information about customer accounts, and the spending group describing customer spending behaviors. The objects in these data sets are categorized jointly by all feature groups but the importance of different feature groups varies in different clusters. The group level difference of features represents important information to subspace clusters and should be considered in the subspace clustering process. This is particularly important in clustering high-dimensional data because the weights on individual features are sensitive to noise and missing values while the weights on feature groups can smooth such sensitivities. Moreover, the introduction of weights to feature groups can eliminate the unbalanced phenomenon caused by the difference of the populations among feature groups. However, the existing subspace clustering algorithms fail to make use of feature group information in clustering high-dimensional data.

In this paper, we propose a new soft subspace clustering method for clustering high-dimensional data from subspaces in both feature groups and individual features. In this method, the features of high-dimensional data are divided into feature groups, based on their natural characteristics. Two types of weights are introduced to simultaneously identify the importance of feature groups and individual features in categorizing each cluster. In this way, the clusters are revealed in subspaces of both feature groups and individual features. A new optimization model is given to define

* Corresponding author.

E-mail addresses: xjchen.hitsz@gmail.com (X. Chen), yeyunming@hit.edu.cn (Y. Ye), xiaofei@hit.edu.cn (X. Xu), zx.huang@sia.ac.cn (J.Z. Huang).

the optimization process in which two types of subspace weights are introduced. We propose a new iterative algorithm FG- k -means to optimize the optimization model. The new algorithm is an extension to k -means, adding two additional steps to automatically calculate the two types of subspace weights.

We present a data generation method to generate high-dimensional data with clusters in subspaces of feature groups. This method was used to generate four types of synthetic data sets for testing our algorithm. Two real-life data sets were also selected for our experiments. The results on both synthetic data and real-life data have shown that in most experiments FG- k -means significantly outperforms the other four k -means algorithms, i.e., k -means, W- k -means [19], LAC [20] and EWKM [21]. The results on synthetic data sets revealed that FG- k -means was robust to noise and missing values. We also conducted an experiment on feature selection with FG- k -means and the results demonstrated that FG- k -means can be used for feature selection.

The remainder of this paper is organized as follows. In Section 2 we state the problem of finding clusters in subspaces of feature groups and individual features. The FG- k -means clustering algorithm is presented in Section 3. In Section 4, we review some related work. Section 5 presents experiments to investigate the properties of two types of subspace weights in FG- k -means. A data generation method is presented in Section 6 for the generation of our synthetic data. The experimental results on synthetic data are presented in Section 7. In Section 8 we present experimental results on two real-life data sets. Experimental results on feature selection are presented in Section 9. We draw conclusions in Section 10.

2. Problem statement

The problem of finding clusters in subspaces of both feature groups and individual features from high-dimensional data can be stated as follows. Let $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ be a high-dimensional data set of n objects and $\mathbf{A} = \{A_1, A_2, \dots, A_m\}$ be the set of m features representing the objects in \mathbf{X} . Let $\mathbf{G} = \{G_1, G_2, \dots, G_T\}$ be a set of T subsets of \mathbf{A} where $G_t \neq \emptyset$, $G_t \subset \mathbf{A}$, $G_t \cap G_s = \emptyset$ and $\bigcup_{t=1}^T G_t = \mathbf{A}$ for $t \neq s$ and $1 \leq t, s \leq T$. Assume that \mathbf{X} contains k clusters of $\{C_1, C_2, \dots, C_k\}$. We want to discover the set of k clusters from subspaces of \mathbf{G} and identify the subspaces of the clusters from two weight matrices $W = [w_{l,t}]_{k \times T}$ and $V = [v_{l,j}]_{k \times m}$, where $w_{l,t}$ indicates the weight that is assigned to the t -th feature group in the l -th cluster and $\sum_{t=1}^T w_{l,t} = 1$ ($1 \leq l \leq k$), and $v_{l,j}$ indicates the weight that is assigned to the j -th feature in the l -th cluster and $\sum_{j \in G_t} v_{l,j} = 1$, $\sum_{j=1}^m v_{l,j} = T$ ($1 \leq l \leq k$, $1 \leq t \leq T$).

Fig. 1 illustrates the relationship of the feature set \mathbf{A} and the feature group set \mathbf{G} in a data set \mathbf{X} . In this example, the data contains 12 features in the feature set \mathbf{A} . The 12 features are divided into three groups $\mathbf{G} = \{G_1, G_2, G_3\}$, where $G_1 = \{A_1, A_3, A_7\}$, $G_2 = \{A_2, A_5, A_9, A_{10}, A_{12}\}$, $G_3 = \{A_4, A_6, A_8, A_{11}\}$. Assume \mathbf{X} contains three clusters in different subspaces of \mathbf{G} that are identified in the 3×3 weight matrix as shown in Fig. 2. We can see that cluster C_1 is mainly characterized by feature group G_1 because the weight for G_1 in this cluster is 0.7, and is much larger than the weights for the other two groups. Similarly, cluster C_3 is categorized by G_3 . However, cluster C_2 is categorized jointly by three feature groups because the weights for the three groups are similar.

If we consider \mathbf{G} as a set of individual features in data \mathbf{X} , this problem is equivalent to the soft subspace clustering in



Fig. 1. Aggregation of individual features to feature groups.

	G_1 (A_1, A_3, A_7)	G_2 ($A_2, A_5, A_9, A_{10}, A_{12}$)	G_3 (A_4, A_6, A_8, A_{11})
C_1	0.7	0.2	0.1
C_2	0.3	0.4	0.3
C_3	0.2	0.3	0.5

Fig. 2. Subspace structure revealed from feature group weight matrix.

[15–18,20,21]. As such, we can consider this method as a generalization of these soft subspace clustering methods. If soft subspace clustering is conducted directly on subspaces in individual features, the group level differences of features are ignored. The weights on subspaces in individual features are sensitive to noise and missing values. Moreover, there may exist unbalanced phenomenon so that the feature group with more features will gain more weights than the feature group with less features. Instead of subspace clustering on individual features, we aggregate features into feature groups and conduct subspace clustering in subspaces of both feature groups and individual features so the subspace clusters can be revealed in subspaces of feature groups and individual features. The weights on feature groups are then less sensitive to noise and missing values. The unbalanced phenomenon caused by the difference of the populations among feature groups can be eliminated by the introduction of weights to feature groups.

3. The FG- k -means algorithm

In this section, we present an optimization model for finding clusters of high-dimensional data from subspaces of feature groups and individual features and propose FG- k -means, a soft subspace clustering algorithm for high-dimensional data.

3.1. The optimization model

To cluster \mathbf{X} into k clusters in subspaces of both feature groups and individual features, we propose the following objective function to optimize in the clustering process:

$$P(U, Z, V, W) = \sum_{l=1}^k \left[\sum_{i=1}^n \sum_{t=1}^T \sum_{j \in G_t} u_{i,l} w_{l,t} v_{l,j} d(x_{i,j}, z_{l,j}) + \lambda \sum_{t=1}^T w_{l,t} \log(w_{l,t}) + \eta \sum_{j=1}^m v_{l,j} \log(v_{l,j}) \right] \quad (1)$$

subject to

$$\begin{cases} \sum_{l=1}^k u_{i,l} = 1, & u_{i,l} \in \{0,1\}, & 1 \leq i \leq n \\ \sum_{l=1}^k w_{l,t} = 1, & 0 < w_{l,t} < 1, & 1 \leq t \leq T \\ \sum_{j \in G_t} v_{l,j} = 1, & 0 < v_{l,j} < 1, & 1 \leq l \leq k \\ & 1 \leq t \leq T \end{cases} \quad (2)$$

where

- U is a $n \times k$ partition matrix whose elements $u_{i,l}$ are binary where $u_{i,l} = 1$ indicates that the i -th object is allocated to the l -th cluster.

- $Z = \{Z_1, Z_2, \dots, Z_k\}$ is a set of k vectors representing the centers of the k clusters.
- $V = [v_{l,j}]_{k \times m}$ is a weight matrix where $v_{l,j}$ is the weight of the j -th feature on the l -th cluster. The elements in V satisfy $\sum_{j \in G_t} v_{l,j} = 1$ for $1 \leq l \leq k$ and $1 \leq t \leq T$.
- $W = [w_{l,t}]_{k \times T}$ is a weight matrix where $w_{l,t}$ is the weight of the t -th feature group on the l -th cluster. The elements in W satisfy $\sum_{t=1}^T w_{l,t} = 1$ for $1 \leq l \leq k$.
- $\lambda > 0, \eta > 0$ are two given parameters. λ is used to adjust the distribution of W and η is used to adjust the distribution of V .
- $d(x_{i,j}, z_{l,j})$ is a distance or dissimilarity measure between object i and the center of cluster l on the j -th feature. If the feature is numeric, then

$$d(x_{i,j}, z_{l,j}) = (x_{i,j} - z_{l,j})^2 \quad (3)$$

If the feature is categorical, then

$$d(x_{i,j}, z_{l,j}) = \begin{cases} 0 & (x_{i,j} = z_{l,j}) \\ 1 & (x_{i,j} \neq z_{l,j}) \end{cases} \quad (4)$$

The first term in (1) is a modification of the objective function in [21], by weighting subspaces in both feature groups and individual features instead of subspaces only in individual features. The second and third terms are two negative weight entropies that control the distributions of two types of weights through two parameters λ and η . Large parameters make the weights more evenly distributed, otherwise, more concentrated on some subspaces.

3.2. The FG- k -means clustering algorithm

We can minimize (1) by iteratively solving the following four minimization problems:

1. Problem P_1 : Fix $Z = \hat{Z}$, $V = \hat{V}$ and $W = \hat{W}$, and solve the reduced problem $P(U, \hat{Z}, \hat{V}, \hat{W})$;
2. Problem P_2 : Fix $U = \hat{U}$, $V = \hat{V}$ and $W = \hat{W}$, and solve the reduced problem $P(\hat{U}, \hat{Z}, \hat{V}, \hat{W})$;
3. Problem P_3 : Fix $U = \hat{U}$, $Z = \hat{Z}$ and $W = \hat{W}$, and solve the reduced problem $P(\hat{U}, \hat{Z}, V, \hat{W})$;
4. Problem P_4 : Fix $U = \hat{U}$, $Z = \hat{Z}$ and $V = \hat{V}$, and solve the reduced problem $P(\hat{U}, \hat{Z}, \hat{V}, W)$.

Problem P_1 is solved by

$$\begin{cases} u_{i,l} = 1 & \text{if } D_l \leq D_s \text{ for } 1 \leq s \leq k \\ & \text{where } D_s = \sum_{t=1}^T w_{s,t} \sum_{j \in G_t} v_{s,j} d(x_{i,j}, z_{s,j}) \\ u_{i,s} = 0 & \text{for } s \neq l \end{cases} \quad (5)$$

and problem P_2 is solved for the numerical features by

$$z_{l,j} = \frac{\sum_{i=1}^n u_{i,l} x_{i,j}}{\sum_{i=1}^n u_{i,l}} \quad \text{for } 1 \leq l \leq k \quad (6)$$

If the feature is categorical, then

$$z_{l,j} = a_j^l \quad (7)$$

where a_j^l is the mode of the categorical values of the j -th feature in cluster l [26].

The solution to problem P_3 is given by Theorem 1:

Theorem 1. Let $U = \hat{U}$, $Z = \hat{Z}$, $W = \hat{W}$ be fixed and $\eta > 0$. $P(\hat{U}, \hat{Z}, V, \hat{W})$ is minimized iff

$$v_{l,j} = \frac{\exp\left\{\frac{-E_{l,j}}{\eta}\right\}}{\sum_{h \in G_t} \exp\left\{\frac{-E_{l,h}}{\eta}\right\}} \quad (8)$$

where

$$E_{l,j} = \sum_{i=1}^n \hat{u}_{i,l} \hat{w}_{l,t} d(x_{i,j}, \hat{z}_{l,j}) \quad (9)$$

Here, t is the index of the feature group which the j -th feature is assigned to.

Proof. Given \hat{U} , \hat{Z} and \hat{W} , we minimize the objective function (1) with respect to $v_{l,j}$, the weight of the j -th feature on the l -th cluster. Since there exist a set of $k \times T$ constraints $\sum_{j \in G_t} v_{l,j} = 1$, we form the Lagrangian by isolating the terms which contain $\{v_{l,1}, \dots, v_{l,m}\}$ and adding the appropriate Lagrangian multipliers as

$$L_{[v_{l,1}, \dots, v_{l,m}]} = \sum_{t=1}^T \left[\sum_{j \in G_t} v_{l,j} E_{l,j} + \eta \sum_{j \in G_t} v_{l,j} \log(v_{l,j}) + \gamma_{l,t} \left(\sum_{j \in G_t} v_{l,j} - 1 \right) \right] \quad (10)$$

where $E_{l,j}$ is a constant in the t -th feature group on the l -th cluster for fixed \hat{U} , \hat{Z} and \hat{W} , and calculated by (9).

By setting the gradient of $L_{[v_{l,1}, \dots, v_{l,m}]}$ with respect to $\gamma_{l,t}$ and $v_{l,j}$ to zero, we obtain

$$\frac{\partial L_{[v_{l,1}, \dots, v_{l,m}]}}{\partial \gamma_{l,t}} = \sum_{j \in G_t} v_{l,j} - 1 = 0 \quad (11)$$

and

$$\frac{\partial L_{[v_{l,1}, \dots, v_{l,m}]}}{\partial v_{l,j}} = E_{l,j} + \eta(1 + \log(v_{l,j})) + \gamma_{l,t} = 0 \quad (12)$$

where t is the index of the feature group which the j -th feature is assigned to.

From (12), we obtain

$$v_{l,j} = \exp\left\{\frac{-E_{l,j} - \gamma_{l,t} - \eta}{\eta}\right\} = \exp\left\{\frac{-E_{l,j} - \eta}{\eta}\right\} \exp\left\{\frac{-\gamma_{l,t}}{\eta}\right\} \quad (13)$$

Substituting (13) into (11), we have

$$\sum_{j \in G_t} \exp\left\{\frac{-E_{l,j} - \eta}{\eta}\right\} \exp\left\{\frac{-\gamma_{l,t}}{\eta}\right\} = 1$$

It follows that

$$\exp\left\{\frac{-\gamma_{l,t}}{\eta}\right\} = \frac{1}{\sum_{j \in G_t} \exp\left\{\frac{-E_{l,j} - \eta}{\eta}\right\}}$$

Substituting this expression back into (13), we obtain

$$v_{l,j} = \frac{\exp\left\{\frac{-E_{l,j}}{\eta}\right\}}{\sum_{h \in G_t} \exp\left\{\frac{-E_{l,h}}{\eta}\right\}} \quad \square$$

The solution to problem P_4 is given by Theorem 2:

Theorem 2. Let $U = \hat{U}$, $Z = \hat{Z}$, $V = \hat{V}$ be fixed and $\lambda > 0$. $P(\hat{U}, \hat{Z}, \hat{V}, W)$ is minimized iff

$$w_{l,t} = \frac{\exp\left(\frac{-D_{l,t}}{\lambda}\right)}{\sum_{s=1}^T \exp\left(\frac{-D_{l,s}}{\lambda}\right)} \quad (14)$$

where

$$D_{l,t} = \sum_{i=1}^n \hat{u}_{i,l} \sum_{j \in G_t} \hat{v}_{l,j} d(x_{i,j}, \hat{z}_{l,j}) \quad (15)$$

Proof. Given \hat{U} , \hat{Z} and \hat{V} , we minimize the objective function (1) with respect to $w_{l,t}$, the weight of the t -th feature group on the l -th cluster. Since there exist a set of k constraints $\sum_{t=1}^T w_{l,t} = 1$, we form the Lagrangian by isolating the terms which contain $\{w_{l,1}, \dots, w_{l,T}\}$ and adding the appropriate Lagrangian multipliers as

$$L_{[w_{l,1}, \dots, w_{l,T}]} = \sum_{t=1}^T \left[w_{l,t} D_{l,t} + \lambda \sum_{t=1}^T w_{l,t} \log w_{l,t} + \gamma \left(\sum_{t=1}^T w_{l,t} - 1 \right) \right] \quad (16)$$

where $D_{l,t}$ is a constant of the t -th feature group on the l -th cluster for fixed \hat{U} , \hat{Z} and \hat{V} , and calculated by (15).

Taking the derivative with respect to $w_{l,t}$ and setting it to zero yields a minimum of $w_{l,t}$ at (where we have dropped the argument γ):

$$\hat{w}_{l,t} = \frac{\exp\left(\frac{-D_{l,t}}{\lambda}\right)}{\sum_{s=1}^T \exp\left(\frac{-D_{l,s}}{\lambda}\right)} \quad \square$$

The FG- k -means algorithm that minimizes the objective function (1) using formulae (5)–(9), (14) and (15) is given as Algorithm 1.

Algorithm 1. FG- k -means.

Input: The number of clusters k and two positive parameters λ , η ;

Output: Optimal values of U , Z , V , W ;
Randomly choose k cluster centers Z^0 , set all initial weights in V^0 and W^0 to equal values;
 $t := 0$

repeat

Update U^{t+1} by (5);
Update Z^{t+1} by (6) or (7);
Update V^{t+1} by (8) and (9);
Update W^{t+1} by (14) and (15);
 $t := t + 1$

until the objective function (1) obtains its local minimum value;

In FG- k -means, the input parameters λ and η are used to control the distributions of the two types of weights W and V . We can easily verify that the objective function (1) can be minimized with respect to V and W iff $\eta \geq 0$ and $\lambda \geq 0$. Moreover, they are used as follows:

- $\eta > 0$. In this case, according to (8), v is inversely proportional to E . The smaller $E_{l,j}$, the larger $v_{l,j}$ and the more important the corresponding feature.
- $\eta = 0$. It will produce a clustering result with only one import feature in a feature group. It may not be desirable for high-dimensional data.
- $\lambda > 0$. In this case, according to (14), w is inversely proportional to D . The smaller $D_{l,t}$, the larger $w_{l,t}$ and the more important the corresponding feature group.
- $\lambda = 0$. It will produce a clustering result with only one import feature group. It may not be desirable for high-dimensional data.

In general, λ and η are set as positive real values.

Since the sequence of (P_1, P_2, \dots) generated by the algorithm is strictly decreasing, Algorithm 1 converges to a local minima.

The FG- k -means algorithm is an extension to the k -means algorithm by adding two additional steps to calculate two types of weights in the iterative process. It does not seriously affect the scalability of the k -means clustering process in clustering large data. If the FG- k -means algorithm needs r iterations to converge, we can easily verify that the computational complexity is $O(rknm)$. Therefore, FG- k -means has the same computational complexity like k -means.

4. Related work

To our knowledge, SYNCLUS is the first clustering algorithm that uses weights for feature groups in the clustering process [11]. The SYNCLUS clustering process is divided into two stages. Starting from an initial set of feature weights, SYNCLUS first uses the k -means clustering process to partition the data into k clusters. It then estimates a new set of optimal weights by optimizing a weighted mean-square, stress-like cost function. The two stages iterate until the clustering process converges to an optimal set of feature weights. SYNCLUS computes feature weights automatically and the feature group weights are given by users. Another weakness of SYNCLUS is that it is time-consuming [27] so it cannot process large data sets.

Huang et al. [19] proposed the W- k -means clustering algorithm that can automatically compute feature weights in the k -means clustering process. W- k -means extends the standard k -means algorithm with one additional step to compute feature weights at each iteration of the clustering process. The feature weight is inversely proportional to the sum of the within-cluster variances of the feature. As such, noise features can be identified and their affects on the clustering result are significantly reduced.

Friedman and Meulman [18] proposed a method to cluster objects on subsets of attributes. Instead of assigning a weight to each feature for the entire data set, their approach is to compute a weight for each feature in each cluster. Friedman and Meulman proposed two approaches to minimize its objective function. However, both approaches involve the computation of dissimilarity matrices among objects in each iterative step which has a high computational complexity of $O(m^2m)$ (where n is the number of objects, m is the number of features and r is the number of iterations). In other words, their method is not practical for large-volume and high-dimensional data.

Domeniconi et al. [20] proposed the Locally Adaptive Clustering (LAC) algorithm which assigns a weight to each feature in each cluster. They use an iterative algorithm to minimize its objective function. However, Liping et al. [21] have pointed out that “the objective function of LAC is not differentiable because of a maximum function. The convergence of the algorithm is proved by replacing the largest average distance in each dimension with a fixed constant value”.

Liping et al. [21] proposed the entropy weighting k -means (EWKM) which also assigns a weight to each feature in each cluster. Different from LAC, EWKM extends the standard k -means algorithm with one additional step to compute feature weights for each cluster at each iteration of the clustering process. The weight is inversely proportional to the sum of the within-cluster variances of the feature in the cluster. EWKM only weights subspaces in individual features. The new algorithm we present in this paper weights subspaces in both feature groups and individual features. Therefore, it is an extension to EWKM.

Hoff [28] proposed a multivariate Dirichlet process mixture model which is based on a Pólya urn cluster model for multivariate

means and variances. The model is learned by a Markov chain Monte Carlo process. However, its computational cost is prohibitive. Bouveyron et al. [22] proposed the GMM model which takes into account the specific subspaces around which each cluster is located, and therefore limits the number of parameters to estimate. Tsai and Chiu [23] developed a feature weights self-adjustment mechanism for k -means clustering on relational data sets, in which the feature weights are automatically computed by simultaneously minimizing the separations within clusters and maximizing the separations between clusters. Deng et al. [29] proposed an enhanced soft subspace clustering algorithm (ESSC) which employs both within-cluster and between-cluster information in the subspace clustering process. Cheng et al. [24] proposed another weighted k -means approach very similar to LAC, but allowing for incorporation of further constraints.

Generally speaking, none of the above methods takes weights of subspaces in both individual features and feature groups into consideration.

5. Properties of FG- k -means

We have implemented FG- k -means in java and the source code can be found at <http://code.google.com/p/k-means/>. In this section, we use a real-life data set to investigate the relationship between the two types of weights w , v and three parameters k , λ and η in FG- k -means.

5.1. Characteristics of the Yeast Cell Cycle data set

The Yeast Cell Cycle data set is microarray data from yeast cultures synchronized by four methods: α factor arrest, elutriation, arrest of a *cdc15* temperature-sensitive mutant and arrest of a *cdc28* temperature-sensitive mutant [30]. Further, it includes data for the B-type cyclin Clb2p and G1 cyclin Cln3p induction experiments. The data set is publicly available at <http://genome-www.stanford.edu/cellcycle/>. The original data contains 6178 genes. In this investigation, we selected 6076 genes on 77 experiments and removed those which had incomplete data. We used the following five feature groups:

- G_1 : contains four features from the B-type cyclin Clb2p and G1 cyclin Cln3p induction experiments;
- G_2 : contains 18 features from the α factor arrest experiment;
- G_3 : contains 24 features from the elutriation experiment;
- G_4 : contains 17 features from the arrest of a *cdc15* temperature-sensitive mutant experiment;
- G_5 : contains 14 features from the arrest of a *cdc28* temperature-sensitive mutant experiment.

5.2. Controlling weight distributions

We set the number of clusters k as {3,4,5,6,7,8,9,10}, λ as {1,2,4,8,12,16,24,32,48,64,80} and η as {1,2,4,8,12,16,24,32,48,64,80}. For each combination of k , λ and η , we ran FG- k -means to produce 100 clustering results and computed the average variances of W and V in the 100 results. Figs. 3–5 show these variances.

From Fig. 3(a), we can see that when η was small, the variances of V decreased with the increase of k . When η was big, the variances of V became almost constant. From Fig. 3(b), we can see λ has similar behavior.

To investigate the relationship among V , W and λ , η , we show results with $k=5$ in Figs. 4(a) and (b), 5(a) and (b). From Fig. 4(a), we can see that the changes of λ did not affect the variance of V too much. We can see from Fig. 4(b) that as λ increased, the

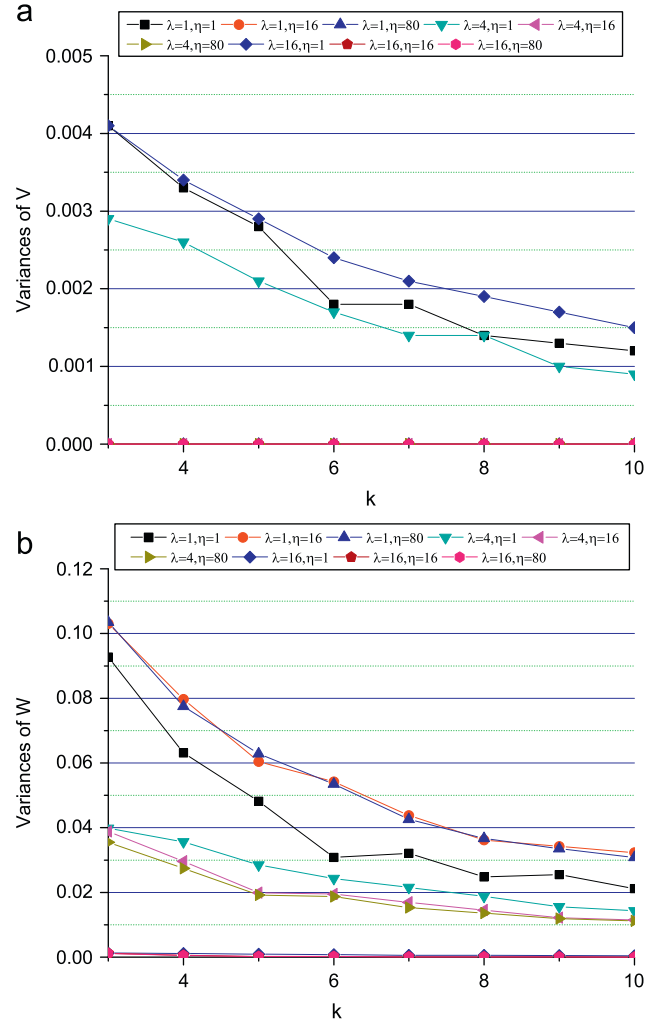


Fig. 3. The variances of W and V of FG- k -means on the Yeast Cell Cycle data set against k . (a) Variances of V against k . (b) Variances of W against k .

variance of W decreased rapidly. This result can be explained from formula (14): as λ increases, W becomes flatter. From Fig. 5(a), we can see that as η increased, the variance of V decreased rapidly. This result can be explained from formula (8): as η increases, V becomes flatter. Fig. 5(b) shows that the effect of η on the variance of W was not obvious.

From above analysis, we summarize the following method to control two types of weight distributions in FG- k -means by setting different values of λ and η :

- Big λ makes more subspaces in feature groups contribute to the clustering while small λ makes only important subspaces in feature groups contribute to the clustering.
- Big η makes more subspaces in individual features contribute to the clustering while small η makes only important subspaces in individual features contribute to the clustering.

6. Data generation method

For testing the FG- k -means clustering algorithm, we present a method in this section to generate high-dimensional data with clusters in subspaces of feature groups.

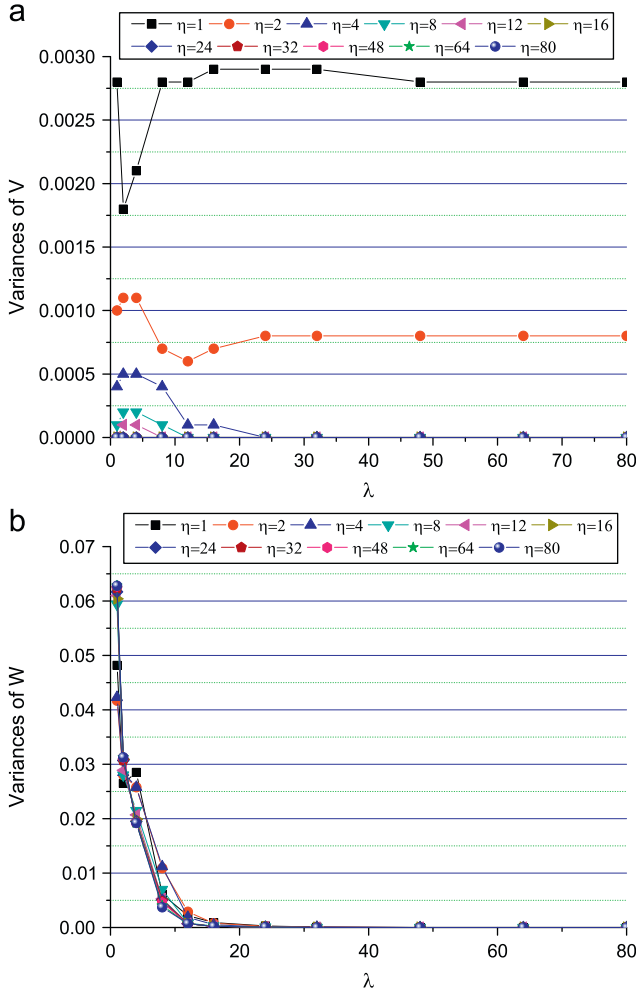


Fig. 4. The variances of W and V of FG- k -means on the Yeast Cell Cycle data set against λ . (a) Variances of V against λ . (b) Variances of W against λ .

6.1. Subspace data generation

Although several methods for generating high-dimensional data have been proposed, for example in [21,31,32], these methods were not designed to generate high-dimensional data containing clusters in subspaces of feature groups. Therefore, we have to design a new method for data generation.

In designing the new data generation method, we first consider that high-dimensional data \mathbf{X} is horizontally and vertically partitioned into $k \times T$ sections where k is the number of clusters in \mathbf{X} and T is the number of feature groups. Fig. 6(a) shows an example of high-dimensional data partitioned into three clusters and three feature groups. There are totally nine data sections. We want to generate three clusters that have inherent cluster features in different vertical sections.

To generate such data, we define a generator that can generate data with specified characteristics. The output from the data generator is called a data area which represents a subset of objects and a subset of features in \mathbf{X} . To generate different characteristics of data, we define three basic types of data areas:

- **Cluster area (C):** Data generated has a multivariate normal distribution in the subset of features.
- **Noise area (N):** Data generated are noise in the subset of features.

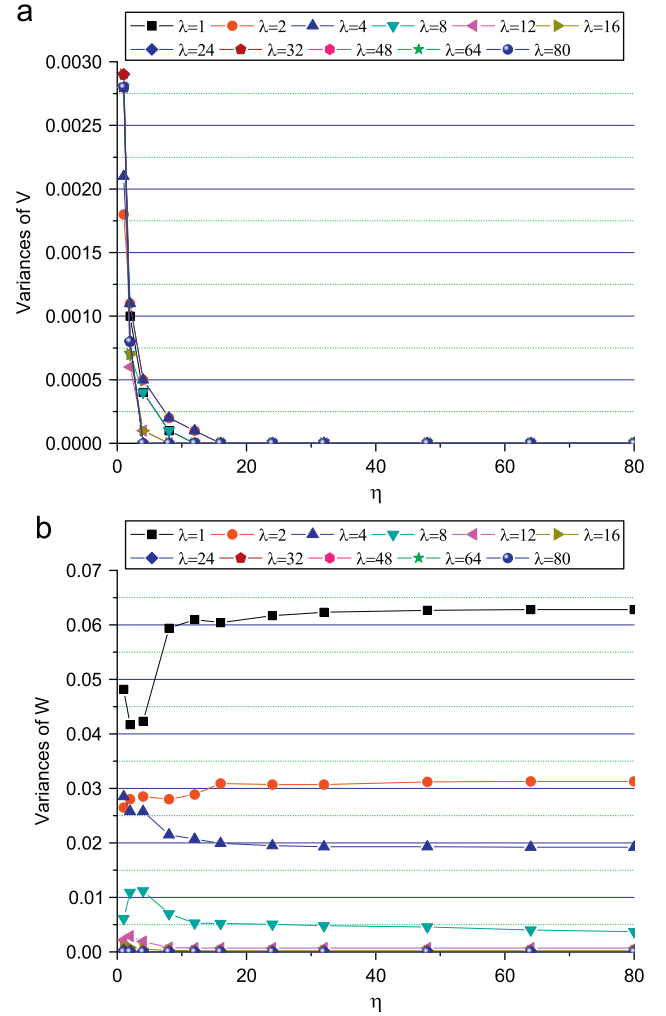


Fig. 5. The variances of W and V of FG- k -means on the Yeast Cell Cycle data set against η . (a) Variances of V against η . (b) Variances of W against η .

- **Missing value area (M):** Data generated are missing values in the subset of features. Here, we consider the area which only contains zero values as a special case of missing value area.

We generate high-dimensional data in two steps. We first use the data generator to generate cluster areas for the partitioned data sections. For each cluster, we generate the cluster areas in the three data sections with different covariances. According to Theorem 1, the larger the covariance, the smaller the group weight. Therefore, the importance of the feature groups to the cluster can be reflected in the data. For example, the darker sections in Fig. 6(a) show the data areas generated with small covariances, therefore, having bigger feature group weights and being more important in representing the clusters. The data generated in this step is called error-free data.

Given an error-free data, in the second step, we choose some data areas to generate noise and missing values by either replacing the existing values with the new values or appending the noise values to the existing values. In this way, we generate data with different levels of noise and missing values.

Fig. 6(b) shows an example of high-dimensional data generated from the error-free data of Fig. 6(a). In this data, all features in G_3 are replaced with noise values. Missing values are introduced to feature A_{12} of feature group G_2 in cluster C_2 . Feature A_2 in feature group G_2 is replaced with noise. The data section of cluster C_3 in feature group G_2 is replaced with noise and feature A_7 in

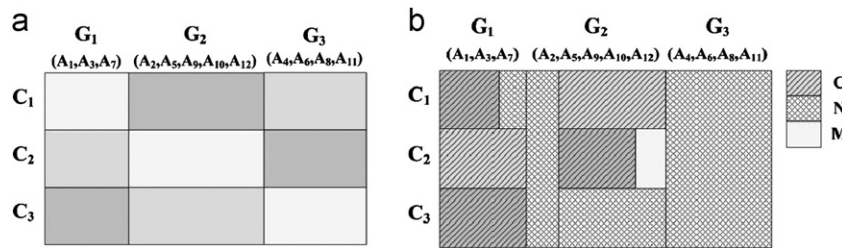


Fig. 6. Examples of subspace structure in data with clusters in subspaces of feature groups. (a) Subspace structure of error-free data. (b) Subspace structure of data with noise and missing values. (C: cluster area, N: noise area, M: missing value area).

feature group G_1 is replaced with noise in cluster C_1 . This introduction of noise and missing values makes the clusters in this data difficult to recover.

6.2. Data quality measure

We define several measures to measure the quality of the generated data. The noise degree is used to evaluate the percentage of noise data in a data set, which is calculated as

$$\epsilon(X) = \frac{\text{no.of data elements with noise values}}{\text{total no. of data elements in } X} \quad (17)$$

The missing value degree is used to evaluate the percentage of missing values in a data set, which is calculated as

$$\rho(X) = \frac{\text{no. of the data elements with missing values}}{\text{total no. of the data elements in } X} \quad (18)$$

7. Synthetic data and experimental results

Four types of synthetic data sets were generated with the data generation method. We ran FG- k -means on these data sets and compared the results with four clustering algorithms, i.e., k -means, W- k -means [19], LAC [20] and EWKM [21].

7.1. Characteristics of synthetic data

Table 1 shows the characteristics of the four synthetic data sets. Each data set contains three clusters and 6000 objects in 200 dimensions which are divided into three feature groups. D_1 is the error-free data, and the other three data sets were generated from D_1 by adding noise and missing values to the data elements. D_2 contains 20% noise. D_3 contains 12% as missing values. D_4 contains 20% noise and 12% as missing values. These data sets were used to test the robustness of clustering algorithms.

7.2. Experiment setup

With the four synthetic data sets listed in Table 1, we carried out two experiments. The first was conducted on four clustering algorithms excluding k -means, and the second was conducted on all five clustering algorithms. The purpose of the first experiment was to select proper parameter values for comparing the clustering performance of five algorithms in the second experiment.

In order to compare the classification performance, we used precision, recall, F-measure and accuracy to evaluate the results. Precision is calculated as the fraction of correct objects among those that the algorithm believes belonging to the relevant class. Recall is the fraction of actual objects that were identified. F-measure is the harmonic mean of precision and recall and accuracy is the proportion of correctly classified objects.

Table 1

Characteristics of four synthetic data sets.

Data sets (X)	n	m	k	T	$\epsilon(X)$	$\rho(X)$
D_1	6000	200	3	3	0	0
D_2					0.2	0
D_3					0	0.12
D_4					0.2	0.12

In the first experiment, we set the parameter values of three clustering algorithms with 30 positive integers from 1 to 30 (β in W- k -means, h in LAC and γ in EWKM). For FG- k -means, we set η as 30 positive integers from 1 to 30 and λ as 10 values of {1, 2, 3, 4, 5, 8, 10, 14, 16, 20}. For each parameter setting, we ran each clustering algorithm to produce 100 clustering results on each of the four synthetic data sets. In the second experiment, we first set the parameter value for each clustering algorithm by selecting the parameter value with the best result in the first experiment. Since the clustering results of the five clustering algorithms were affected by the initial cluster centers, we randomly generated 100 sets of initial cluster centers for each data set. With each initial setting, 100 results were generated from each of five clustering algorithms on each data set.

To statistically compare the clustering performance with four evaluation indices, the paired t -test comparing FG- k -means with the other four clustering methods was computed from 100 clustering results. If the p -value was below the threshold of the statistical significance level (usually 0.05), then the null hypothesis was rejected in favor of an alternative hypothesis, which typically states that the two distributions differ. Thus, if the p -value of two approaches was less than 0.05, the difference of the clustering results of the two approaches was considered to be significant, otherwise, insignificant.

7.3. Results and analysis

Figs. 7–10 draw the average clustering accuracies of four clustering algorithms in the first experiment. From these results, we can observe that FG- k -means produced better results than the other three algorithms on all four data sets, especially on D_3 and D_4 . FG- k -means produced the best results with small values of λ on all four data sets. This indicates that the four data sets have obvious subspaces in feature groups. However, FG- k -means produced the best results with medium values of η on D_1 and D_2 , but with large values of η on D_3 and D_4 . This indicates that the weighting of subspaces in individual features faces considerable challenges when the data contain noise, and especially when the data contain missing values. Under such circumstance, the weights of subspaces in feature groups were more effective than the weights of subspaces in individual features. Among the other three algorithms, W- k -means produced relatively better results

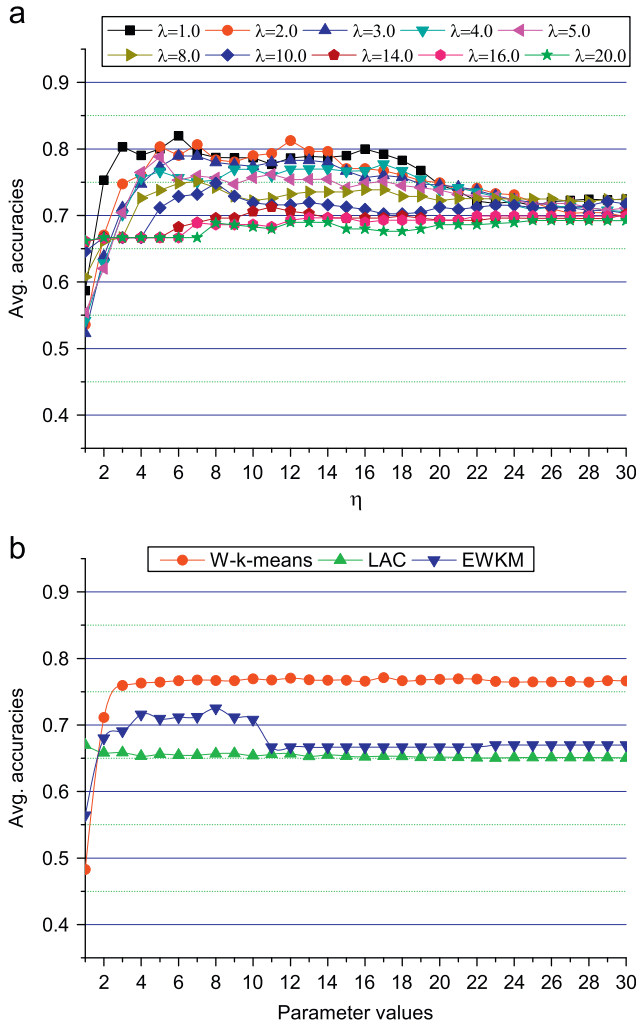


Fig. 7. The clustering results of four clustering algorithms versus their parameter values on D_1 . (a) Average accuracies of FG-k-means. (b) Average accuracies of other three algorithms.

than LAC and EWKM. On D_3 and D_4 , all three clustering algorithms produced bad results indicating that the weighting method in individual features was not effective when the data contain missing values.

In the second experiment, we set the parameters of four algorithms as shown in Table 2. Table 3 summarizes the total 2000 clustering results. We can see that FG-k-means significantly outperformed all other four clustering algorithms in almost all results. When data sets contained missing values, FG-k-means clearly had advantages. The weights for individual features could be misleading because missing values could result in a small variance of a feature in a cluster which would increase the weight of the feature. However, the missing values in feature groups were averaged so the weights in subspaces of feature groups would be less affected by missing values. Therefore, FG-k-means achieved better results on D_3 in all evaluation indices. When noise and missing values were introduced to the error-free data set, all clustering algorithms had considerable challenges in obtaining good clustering results from D_4 . LAC and EWKM produced similar results as the results on D_2 and D_3 , while W-k-means produced much worse results than the results on D_1 and D_2 . However, FG-k-means still produced good results. This indicates that FG-k-means was more robust in handling data with both noise and missing values, which commonly exist in high-dimensional data.

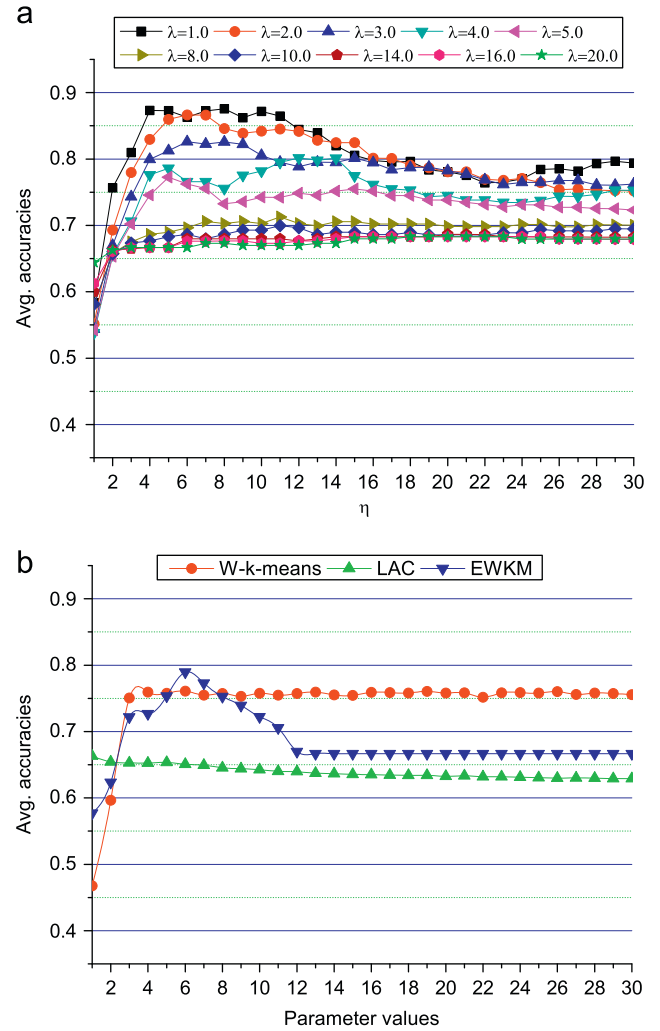


Fig. 8. The clustering results of four clustering algorithms versus their parameter values on D_2 . (a) Average accuracies of FG-k-means. (b) Average accuracies of other three algorithms.

Interestingly, W-k-means outperformed LAC and EWKM on D_2 . This could be caused by the fact that the weights of individual features computed from the entire data set were less affected by the noise values than the weights computed from each cluster.

To sum up, FG-k-means is superior to the other four clustering algorithms in clustering high-dimensional data with clusters in subspaces of feature groups. The results also show that FG-k-means is more robust to the noise and missing values.

7.4. Scalability comparison

To compare the scalability of FG-k-means with the other four clustering algorithms, we retained the subspace structure in D_4 and extended its dimensions from 50 to 500 to generate 10 synthetic data sets. Fig. 11 draws the average time costs of the five algorithms on the 10 synthetic data sets. We can see that the execution time of FG-k-means was more than only EWKM, and significantly less than the other three clustering algorithms. Although EWKM needs more time than k -means in one iteration, the introduction of subspace weights made EWKM faster to converge. Since FG-k-means is an extension to EWKM, the introduction of weights to subspaces of feature groups does not increase the computation in each iteration so much. This result indicates that FG-k-means scales well to high-dimensional data.

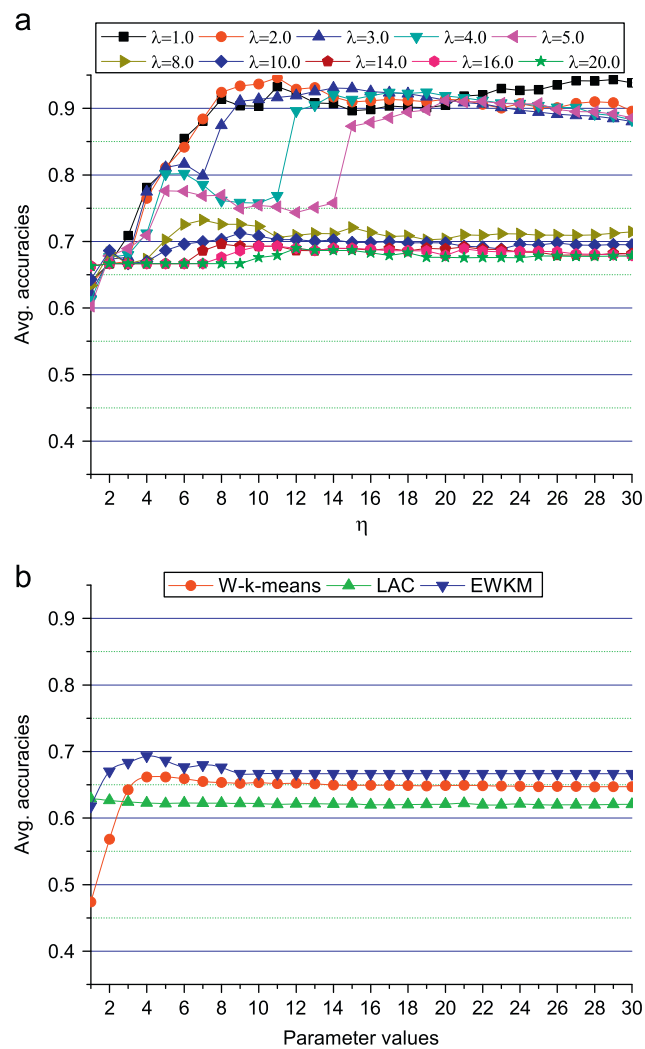


Fig. 9. The clustering results of four clustering algorithms versus their parameter values on D_3 . (a) Average accuracies of FG-k-means. (b) Average accuracies of other three algorithms.

8. Experiments on classification performance of FG-k-means

To investigate the performance of the FG-k-means algorithm in classifying real-life data, we selected two data sets from the UCI Machine Learning Repository [33]: one was the *Image Segmentation* data set and the other was the *Cardiotocography* data set. We compared FG-k-means with four clustering algorithms, i.e., k-means, W-k-means [19], LAC [20], EWKM [21].

8.1. Characteristics of real-life data sets

The *Image Segmentation* data set consists of 2310 objects drawn randomly from a database of seven outdoor images. The data set contains 19 features which can be naturally divided into two feature groups:

1. Shape group: contains the first nine features about the shape information of the seven images.
2. RGB group: contains the last 10 features about the RGB values of the seven images.

Here, we use G_1 and G_2 to represent the two feature groups. The *Cardiotocography* data set consists of 2126 fetal cardiotocograms (CTGs) represented by 21 features. Classification was

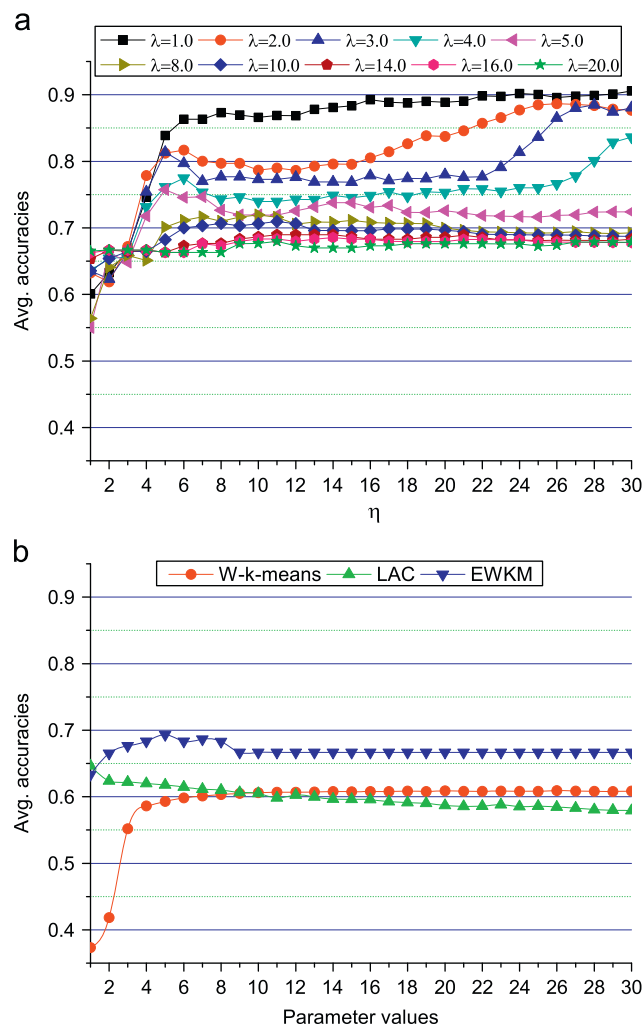


Fig. 10. The clustering results of four clustering algorithms versus their parameter values on D_4 . (a) Average accuracies of FG-k-means. (b) Average accuracies of other three algorithms.

Table 2
Parameter values of four clustering algorithms in the second experiment on the four synthetic data sets in Table 1.

Algorithms	D_1	D_2	D_3	D_4
W-k-means	12	6	5	14
LAC	1	1	1	1
EWKM	3	3	4	7
FG-k-means	(1,15)	(1,12)	(1,20)	(1,20)

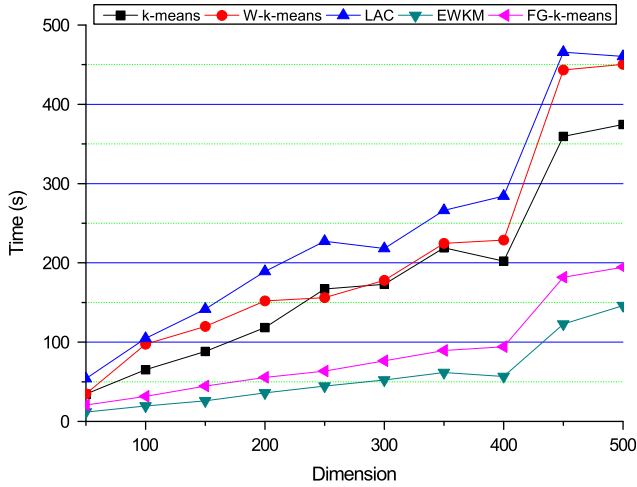
both with respect to a morphologic pattern (A, B, C, ...) and to a fetal state (N, S, P). Therefore, the data set can be used either for 10-class or 3-class experiments. In our experiments, we named this data set as *Cardiotocography1* for the 10-class experiment and *Cardiotocography2* for the 3-class experiment. The 23 features in this data set can be naturally divided into three feature groups:

1. Frequency group: contains the first seven features about the frequency information of the fetal heart rate (FHR) and uterine contraction (UC).
2. Variability group: contains four features about the variability information of these fetal cardiotocograms.
3. Histogram group: contains the last 10 features about the histogram information values of these fetal cardiotocograms.

Table 3

Summary of clustering results on four synthetic data sets listed in Table 1 by five clustering algorithms. The value of the FG-*k*-means algorithm is the mean value of 100 results and the other values are the differences of the mean values between the corresponding algorithms and the FG-*k*-means algorithm. The value in parenthesis is the standard deviation of 100 results. “*” indicates that the difference is significant.

Data	Evaluation indices	<i>k</i> -Means	W- <i>k</i> -means	LAC	EWKM	FG- <i>k</i> -means
D1	Precision	−0.21 (0.12)*	−0.11 (0.22)*	−0.21 (0.15)*	−0.11 (0.18)*	0.84 (0.17)
	Recall	−0.17 (0.09)*	−0.05 (0.14)*	−0.15 (0.08)*	−0.13 (0.10)*	0.82 (0.16)
	F-measure	−0.12 (0.13)*	−0.02 (0.19)	−0.12 (0.13)*	−0.16 (0.13)*	0.75 (0.23)
	Accuracy	−0.17 (0.09)*	−0.05 (0.14)*	−0.15 (0.08)*	−0.13 (0.10)*	0.82 (0.16)
D2	Precision	−0.16 (0.05)*	−0.04 (0.10)	−0.14 (0.07)*	−0.09 (0.20)*	0.82 (0.25)
	Recall	−0.24 (0.04)*	−0.11 (0.10)*	−0.21 (0.06)*	−0.15 (0.13)*	0.87 (0.16)
	F-measure	−0.18 (0.05)*	−0.07 (0.12)*	−0.16 (0.07)*	−0.19 (0.17)*	0.82 (0.22)
	Accuracy	−0.24 (0.04)*	−0.11 (0.10)*	−0.21 (0.06)*	−0.15 (0.13)*	0.87 (0.16)
D3	Precision	−0.26 (0.05)*	−0.25 (0.14)*	−0.26 (0.06)*	−0.33 (0.16)*	0.90 (0.20)
	Recall	−0.32 (0.04)*	−0.27 (0.07)*	−0.31 (0.06)*	−0.24 (0.09)*	0.94 (0.13)
	F-measure	−0.29 (0.06)*	−0.27 (0.11)*	−0.29 (0.08)*	−0.32 (0.12)*	0.91 (0.18)
	Accuracy	−0.32 (0.04)*	−0.27 (0.07)*	−0.31 (0.06)*	−0.24 (0.09)*	0.94 (0.13)
D4	Precision	−0.29 (0.05)*	−0.26 (0.07)*	−0.23 (0.05)*	−0.32 (0.16)*	0.89 (0.17)
	Recall	−0.31 (0.04)*	−0.30 (0.06)*	−0.26 (0.04)*	−0.22 (0.08)*	0.91 (0.13)
	F-measure	−0.29 (0.05)*	−0.28 (0.07)*	−0.23 (0.05)*	−0.30 (0.11)*	0.88 (0.18)
	Accuracy	−0.31 (0.04)*	−0.30 (0.06)*	−0.26 (0.04)*	−0.22 (0.08)*	0.91 (0.13)

**Fig. 11.** Average time costs of five clustering algorithms on 10 synthetic data sets.

We can see that different feature groups represent different measurements of the data from different perspectives. In the following, we use the three real-life data sets to investigate the classification performance of the FG-*k*-means clustering algorithm.

8.2. Experiment setup

We conducted two experiments, as with the synthetic data in Section 7.2, and only report the experimental results in the second experiment. In the second experiment, we set the parameters of four clustering algorithms as shown in Table 4.

8.3. Classification results

Table 5 summarizes the total 1500 results produced by the five clustering algorithms on the three real-life data sets. From these results, we can see that FG-*k*-means significantly outperformed the other four algorithms in most results. On the *Image Segmentation* data set, FG-*k*-means significantly outperformed all other four clustering algorithms in recall and accuracy. On the *Cardiotocography1* data set, FG-*k*-means also significantly outperformed all other four clustering algorithms in recall and accuracy. On the

Table 4

Parameter values of four clustering algorithms in the experiment on the three real-life data sets. IS: Image Segmentation data set, Ca1: Cardiotocography1 data set, Ca2: Cardiotocography2 data set.

Algorithms	IS	Ca1	Ca2
W- <i>k</i> -means	30	35	5
LAC	30	30	15
EWKM	30	40	15
FG- <i>k</i> -means	(10,30)	(1,5)	(20,5)

Cardiotocography2 data set, FG-*k*-means significantly outperformed all other four clustering algorithms in the four evaluation indices. From the above results, we can see that the introduction of weights to subspaces of both feature groups and individual features improves the clustering results.

9. Experiments on feature selection

In FG-*k*-means, the weights of feature groups and individual features indicate the importance of the subspaces where the clusters are found. Small weights indicate that the feature groups or individual features are not relevant to the clustering. Therefore, we can do feature selection with these weights. In this section, we show an experiment on a real-life data set for feature selection with FG-*k*-means.

9.1. Characteristics of the Multiple Features data set

The *Multiple Features* data set contains 2000 patterns of hand-written numerals that were extracted from a collection of Dutch utility maps. These patterns were classified into 10 classes (“0”–“9”), each having 200 patterns. Each pattern was described by 649 features that were divided into the following six feature groups:

1. mfeat-fou group: contains 76 Fourier coefficients of the character shapes;
2. mfeat-fac group: contains 216 profile correlations;
3. mfeat-kar group: contains 64 Karhunen-Loève coefficients;
4. mfeat-pix group: contains 240 pixel averages in 2×3 windows;

Table 5

Summary of clustering results on three real-life data sets by five clustering algorithms. The value of the FG-k-means algorithm is the mean value of 100 results and the other values are the differences of the mean values between the corresponding algorithms and the FG-k-means algorithm. The value in parenthesis is the standard deviation of 100 results. “*” indicates that the difference is significant.

Data	Evaluation indices	k-Means	W-k-means	LAC	EWKM	FG-k-means
IS	Precision	−0.00 (0.07)	−0.01 (0.08)	−0.00 (0.07)	−0.00 (0.09)	0.59 (0.09)
	Recall	−0.02 (0.05)*	−0.02 (0.03)*	−0.02 (0.05)*	−0.02 (0.05)*	0.63 (0.05)
	F-measure	−0.00 (0.07)	−0.01 (0.05)	−0.00 (0.07)	−0.01 (0.07)	0.59 (0.07)
	Accuracy	−0.02 (0.05)*	−0.02 (0.03)*	−0.02 (0.05)*	−0.02 (0.05)*	0.63 (0.05)
Ca1	Precision	0.07 (0.03)*	0.05 (0.03)*	0.07 (0.03)*	0.07 (0.03)*	0.40 (0.06)
	Recall	−0.01 (0.02)*	−0.01 (0.02)*	−0.01 (0.02)*	−0.01 (0.02)*	0.38 (0.03)
	F-measure	0.12 (0.02)*	0.12 (0.02)*	0.12 (0.02)*	0.12 (0.02)*	0.27 (0.03)
	Accuracy	−0.01 (0.02)*	−0.01 (0.02)*	−0.01 (0.02)*	−0.01 (0.02)*	0.38 (0.03)
Ca2	Precision	−0.03 (0.01)*	0.03 (0.04)*	−0.03 (0.01)*	−0.02 (0.02)*	0.76 (0.05)
	Recall	−0.36 (0.03)*	−0.29 (0.06)*	−0.36 (0.03)*	−0.02 (0.02)*	0.81 (0.02)
	F-measure	−0.27 (0.04)*	−0.20 (0.07)*	−0.27 (0.04)*	−0.02 (0.01)*	0.77 (0.04)
	Accuracy	−0.36 (0.03)*	−0.29 (0.06)*	−0.36 (0.03)*	−0.02 (0.02)*	0.81 (0.02)

Table 6

Summary of clustering results from the *Multiple Features* data set by five clustering algorithms. The value in the cell is the mean value and the range of 100 results, and the value in parenthesis is the standard deviation of 100 results. “*” indicates that the difference is significant. The bold in each row represents the best result in the corresponding evaluation index.

Evaluation indices	k-Means	W-k-means	LAC	EWKM	FG-k-means
Precision	0.72 ± 0.20 (0.09)	0.74 ± 0.20 (0.10)*	0.72 ± 0.20 (0.09)	0.55 ± 0.17 (0.09)*	0.70 ± 0.25 (0.11)
Recall	0.73 ± 0.18 (0.08)*	0.74 ± 0.19 (0.08)*	0.73 ± 0.19 (0.08)	0.50 ± 0.18 (0.10)*	0.71 ± 0.23 (0.10)
F-measure	0.72 ± 0.20 (0.09)*	0.73 ± 0.20 (0.10)*	0.71 ± 0.21 (0.09)*	0.42 ± 0.20 (0.10)*	0.65 ± 0.29 (0.12)
Accuracy	0.73 ± 0.18 (0.08)*	0.74 ± 0.19 (0.08)*	0.73 ± 0.19 (0.08)	0.50 ± 0.18 (0.10)*	0.71 ± 0.23 (0.10)

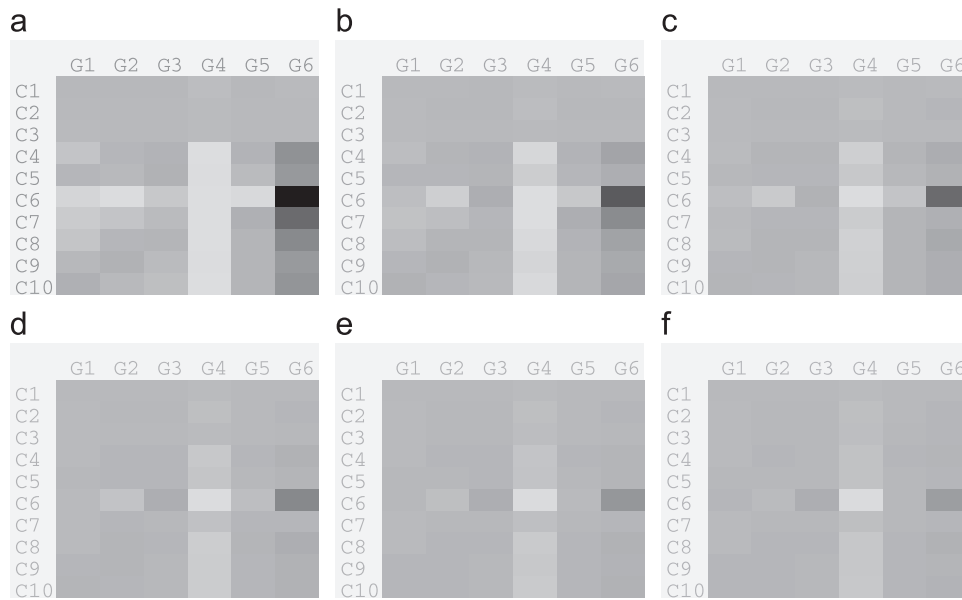


Fig. 12. Subspace structure recovered by the FG-k-means from the *Multiple Features* data set. (a) Subspace structure with $\lambda = 5$. (b) Subspace structure with $\lambda = 10$. (c) Subspace structure with $\lambda = 15$. (d) Subspace structure with $\lambda = 20$. (e) Subspace structure with $\lambda = 25$. (f) Subspace structure with $\lambda = 30$.

5. mfeat-zer group: contains 47 Zernike moments;
6. mfeat-mor group: contains six morphological features.

Here, we use G_1 , G_2 , G_3 , G_4 , G_5 and G_6 to represent the six feature groups.

9.2. Classification results on the Multiple Features data set

In the experiment, we set $\beta = 8$ for W-k-means, $h = 30$ for LAC, $\lambda = 5$ for EWKM and $\lambda = 6$, $\eta = 30$ for FG-k-means. Table 6

summarizes the total 500 results produced by the five clustering algorithms. W-k-means produced the highest average values in all four indices. EWKM produced the worst results. Although FG-k-means is an extension to EWKM, it produced much better results than EWKM. FG-k-means did not produce the highest average values, but it produced the highest maximal results in all four indices. This indicates that the results were unstable in this data set, which may be caused by noise. To find the reason, we investigated the subspace structure of this data set.

Table 7

Summary of clustering results from the *Filtered Multiple Features* data set by five clustering algorithms. The value of the FG-*k*-means algorithm is the mean value of 100 results and the other values are the differences of the mean values between the corresponding algorithms and the FG-*k*-means algorithm. The value in parenthesis is the standard deviation of 100 results. “*” indicates that the difference is significant.

Evaluation indices	<i>k</i> -Means	W- <i>k</i> -means	LAC	EWKM	FG- <i>k</i> -means
Precision	−0.01 (0.10)	−0.01 (0.09)	−0.01 (0.10)	−0.01 (0.09)	0.75 (0.11)
Recall	−0.03 (0.08)*	−0.03 (0.08)*	−0.03 (0.08)*	−0.03 (0.08)*	0.79 (0.10)
F-measure	−0.02 (0.10)	−0.02 (0.09)	−0.02 (0.09)	−0.02 (0.09)	0.75 (0.11)
Accuracy	−0.03 (0.08)*	−0.03 (0.08)*	−0.03 (0.08)*	−0.03 (0.08)*	0.79 (0.10)

We set λ as {5,10,15,20,25,30} and η as 30 positive integers from 1 to 30, and then ran FG-*k*-mean with 100 randomly generated cluster centers to produce 18,000 clustering results. For each value of λ , we computed the average weight of each feature group in each cluster from 3000 clustering results. Fig. 12 draws the six sets of average weights, where the dark color indicates high weight and the light color represents low weight. We can see that subspace structures recovered are similar with different values of λ . We noticed that most weights in G_4 were very small, which indicates that G_4 was not important and could be considered as a noise feature group. This feature group could be the cause that made the cluster structure of this data insignificant and these clustering algorithms sensitive to the initial cluster centers.

9.3. Feature selection

To further investigate the assumption that G_4 was a noise feature group, we conducted a new experiment. In the new experiment, we deleted the features in G_4 and produced the *Filtered Multiple Features* data set which only contained 409 features. We set $\beta = 30$ for W-*k*-means, $h = 30$ for LAC, $\lambda = 30$ for EWKM and $\lambda = 20$, $\eta = 11$ for FG-*k*-means and ran each of the five algorithms 100 times with 100 randomly generated cluster centers. Table 7 summarizes the total 500 results produced by the five clustering algorithms. Compared with the results in Table 6, we can see that all algorithms improved their results, especially EWKM and FG-*k*-means. EWKM resulted in significant increases in performance and the new results were comparable to W-*k*-means and LAC. FG-*k*-means significantly outperformed the other four clustering algorithms in recall and accuracy. In precision and F-measure, FG-*k*-means produced similar results as the other four clustering algorithms. These results indicate that the cluster structure of this data set was made more obvious and easier to recover for soft subspace clustering algorithms after removing the features in G_4 . In this way FG-*k*-means can be used for feature selection.

10. Conclusions

In this paper, we have presented a new clustering algorithm FG-*k*-means to cluster high-dimensional data from subspaces of feature groups and individual features. Given a high-dimensional data set with features divided into groups, FG-*k*-means can discover clusters in subspaces by automatically computing feature group weights and individual feature weights. From the two types of weights, the subspaces of clusters can be revealed. The experimental results on both synthetic and real-life data sets have shown that the FG-*k*-means algorithm outperformed the other four clustering algorithms, i.e., *k*-means, W-*k*-means, LAC and EWKM. The results on synthetic data also show that FG-*k*-means was more robust to noise and missing values. Finally, the experimental results on a real-life data set demonstrated that FG-*k*-means can be used in feature selection.

Our future work will develop a method that can automatically divide features into groups in the weighted clustering process. Moreover, the weighting method used in FG-*k*-means can also be considered for other clustering and classification methods. Finally, we will test and improve our method on further real applications.

Acknowledgment

This research is supported in part by NSFC under Grant no. 61073195, and Shenzhen New Industry Development Fund under Grant nos. CXB201005250024A and CXB201005250021A.

References

- [1] D. Donoho, High-dimensional data analysis: the curses and blessings of dimensionality, American Mathematical Society-Mathematical Challenges of the 21st Century, Los Angeles, CA, USA, 2000.
- [2] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, ACM SIGKDD Explorations Newsletter 6 (1) (2004) 90–105.
- [3] H. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: a survey on subspace clustering, pattern based clustering, and correlation clustering, ACM Transactions on Knowledge Discovery from Data 3 (1) (2009) 1–58.
- [4] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, in: Proceedings of ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA, 1998, pp. 94–105.
- [5] C.C. Aggarwal, J.L. Wolf, P.S. Yu, C. Procopiuc, J.S. Park, Fast algorithms for projected clustering, in: Proceedings of ACM SIGMOD International Conference on Management of Data, Philadelphia, Pennsylvania, USA, 1999, pp. 61–72.
- [6] C.C. Aggarwal, P.S. Yu, finding generalized projected clusters in high dimensional spaces, in: Proceedings of ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, 2000, pp. 70–81.
- [7] K. Chakrabarti, S. Mehrotra, Local dimensionality reduction: a new approach to indexing high dimensional spaces, in: Proceedings of the 26th International Conference on Very Large Data Bases, Cairo, Egypt, 2000, pp. 89–100.
- [8] C. Procopiuc, M. Jones, P. Agarwal, T. Murali, A Monte Carlo algorithm for fast projective clustering, in: Proceedings of ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, USA, 2002, pp. 418–427.
- [9] K. Yip, D. Cheung, M. Ng, HARP: a practical projected clustering algorithm, IEEE Transactions on Knowledge and Data Engineering 16 (11) (2004) 1387–1397.
- [10] K. Yip, D. Cheung, M. Ng, On discovery of extremely low-dimensional clusters using semi-supervised projected clustering, in: Proceedings of the 21st International Conference on Data Engineering, Tokyo, Japan, 2005, pp. 329–340.
- [11] W. DeSarbo, J. Carroll, L. Clark, P. Green, Synthesized clustering: a method for amalgamating alternative clustering bases with differential weighting of variables, Psychometrika 49 (1) (1984) 57–78.
- [12] G. Milligan, A validation study of a variable weighting algorithm for cluster analysis, Journal of Classification 6 (1) (1989) 53–71.
- [13] D. Modha, W. Spangler, Feature weighting in *k*-means clustering, Machine Learning 52 (3) (2003) 217–237.
- [14] E.Y. Chan, W.-K. Ching, M.K. Ng, J.Z. Huang, An optimization algorithm for clustering using weighted dissimilarity measures, Pattern Recognition 37 (5) (2004) 943–952.
- [15] H. Frigui, O. Nasraoui, Simultaneous clustering and dynamic keyword weighting for text documents, in: M.W. Berry (Ed.), Survey of Text Mining: Clustering, Classification, and Retrieval, Springer, New York, 2004, pp. 45–72.
- [16] H. Frigui, O. Nasraoui, Unsupervised learning of prototypes and attribute weights, Pattern Recognition 37 (3) (2004) 567–581.
- [17] C. Domeniconi, D. Papadopoulos, D. Gunopulos, S. Ma, Subspace clustering of high dimensional data, in: Proceedings of the Fourth SIAM International

- Conference on Data Mining, Lake Buena Vista, Florida, USA, 2004, pp. 517–521.
- [18] J. Friedman, J. Meulman, Clustering objects on subsets of attributes, *Journal of the Royal Statistical Society Series B (Statistical Methodology)* 66 (4) (2004) 815–849.
 - [19] Z. Huang, M. Ng, H. Rong, Z. Li, Automated variable weighting in *k*-means type clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (5) (2005) 657–668.
 - [20] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, D. Papadopoulos, Locally adaptive metrics for clustering high dimensional data, *Data Mining and Knowledge Discovery* 14 (1) (2007) 63–97.
 - [21] L. Jing, M. Ng, Z. Huang, An entropy weighting *k*-means algorithm for subspace clustering of high-dimensional sparse data, *IEEE Transactions on Knowledge and Data Engineering* 19 (8) (2007) 1026–1041.
 - [22] C. Bouveyron, S. Girard, C. Schmid, High dimensional data clustering, *Computational Statistics & Data Analysis* 52 (1) (2007) 502–519.
 - [23] C.-Y. Tsai, C.-C. Chiu, Developing a feature weight self-adjustment mechanism for a *k*-means clustering algorithm, *Computational Statistics & Data Analysis* 52 (10) (2008) 4658–4672.
 - [24] H. Cheng, K.A. Hua, K. Vu, Constrained locally weighted clustering, in: *Proceedings of the VLDB Endowment*, vol. 1, Auckland, New Zealand, 2008, pp. 90–101.
 - [25] J. Mui, K. Fu, Automated classification of nucleated blood cells using a binary tree classifier, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2 (5) (1980) 429–443.
 - [26] Z. Huang, Extensions to the *k*-means algorithms for clustering large data sets with categorical values, *Data Mining and Knowledge Discovery* 2 (3) (1998) 283–304.
 - [27] P. Green, J. Kim, F. Carmone, A preliminary study of optimal variable weighting in *k*-means clustering, *Journal of Classification* 7 (2) (1990) 271–285.
 - [28] P. Hoff, Model-based subspace clustering, *Bayesian Analysis* 1 (2) (2006) 321–344.
 - [29] Z. Deng, K. Choi, F. Chung, S. Wang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, *Pattern Recognition* 43 (3) (2010) 767–781.
 - [30] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, B. Futcher, Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization, *Molecular Biology of the Cell* 9 (12) (1998) 3273–3297.
 - [31] G. Milligan, P. Isaac, The validation of four ultrametric clustering algorithms, *Pattern Recognition* 12 (2) (1980) 41–50.
 - [32] M. Zait, H. Messatfa, A comparative study of clustering methods, *Future Generation Computer Systems* 13 (2–3) (1997) 149–159.
 - [33] A. Frank, A. Asuncion, UCI Machine Learning Repository <<http://archive.ics.uci.edu/ml>>, 2010.

Xiaojun Chen is a Ph.D. student in the Shenzhen Graduate School, Harbin Institute of Technology, China. His research interests are in the areas of data mining, subspace clustering algorithm, topic model and business intelligence.

Yunming Ye received the Ph.D. degree in Computer Science from Shanghai Jiao Tong University. He is now a Professor in the Shenzhen Graduate School, Harbin Institute of Technology, China. His research interests include data mining, text mining, and clustering algorithm.

Xiaofei Xu received B.S. Degree, M.S. Degree and Ph.D. Degree in the Department of Computer Science and Engineering in Harbin Institute of Technology (HIT) in 1982, 1985 and 1988, respectively. He is now a Professor in the Department of Computer Science and Engineering, Harbin Institute of Technology. His research interests include enterprise computing, service computing and service engineering, enterprise interoperability, enterprise modeling, ERP and supply chain management systems, databases and data mining, knowledge management software engineering.

Joshua Zhexue Huang is a professor and Chief Scientist at Shenzhen Institutes of Advanced Technology Chinese Academy of Sciences, and Honorary Professor at Department of Mathematics, The University of Hong Kong. He is known for his contribution to a series of *k*-means type clustering algorithms in data mining that is widely cited and used, and some have been included in commercial software. He has led the development of the open source data mining system AlphaMiner (www.alphaminer.org) that is widely used in education, research and industry. He has extensive industry expertise in business intelligence and data mining and has been involved in numerous consulting projects in Australia, Hong Kong, Taiwan and mainland China. Dr. Huang received his Ph.D. degree from the Royal Institute of Technology in Sweden. He has published over 100 research papers in conferences and journals.