Pattern Recognition 48 (2015) 3703–3713

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Subspace clustering with automatic feature grouping

Guojun Gan [a,*], Michael Kwok-Po Ng [b]

[a] Department of Mathematics, The Institute for Systems Genomics, and The Center for Health, Intervention, and Prevention (CHIP), University of Connecticut, 196 Auditorium Rd U-3009, Storrs, CT 06269, USA
[b] Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 30 December 2014
Received in revised form
12 April 2015
Accepted 16 May 2015
Available online 29 May 2015

Keywords:
Data clustering
Subspace clustering
k-means
Feature group

ABSTRACT

This paper proposes a subspace clustering algorithm with automatic feature grouping for clustering high-dimensional data. In this algorithm, a new component is introduced into the objective function to capture the feature groups and a new iterative process is defined to optimize the objective function so that the features of high-dimensional data are grouped automatically. Experiments on both synthetic data and real data show that the new algorithm outperforms the FG-*k*-means algorithm in terms of accuracy and choice of parameters.

1. Introduction

As one of the major tasks of data mining, data clustering is a process that aims to identify homogeneous groups or clusters of objects from a set of objects. Given a set of multi-dimensional data points, clustering algorithms can be used to find a partition of the points into clusters such that the points within a cluster are similar to each other and the points from different clusters are quite distinct [1,2]. Data clustering can be applied to a wide range of areas such as bioinformatics [3], pattern recognition [4], health care [5], insurance [6], to just name a few.

In the past six decades, many clustering algorithms have been developed. The *k*-means algorithm is one of the oldest and most widely used clustering algorithm [7]. In the *k*-means algorithm, the number of clusters is a required input. Given a dataset and a number *k* of clusters, the *k*-means algorithm starts from *k* initial cluster centers and then repeats updating the cluster memberships and the cluster centers until some stop criterion is met [8]. A key problem of the *k*-means algorithm and other conventional clustering algorithms is that they suffer from the curse of dimensionality. In high-dimensional data, clusters are usually embedded in subspaces of the original data space and different clusters might be embedded in different subspaces. As a result, these conventional clustering algorithms are not efficient to deal with high-dimensional data.

To address this problem, subspace clustering algorithms have been developed to identify clusters embedded in subspaces of the original data space. Agrawal et al. proposed a clustering algorithm called CLIQUE to find dense subspace clusters [9]. Parsons et al. presented a review of subspace clustering algorithms developed up to that time [10]. In [11], Huang et al. proposed a subspace clustering algorithm called W-*k*-means by introducing feature weighting to the *k*-means algorithm. Gan and Wu proposed the FSC algorithm and proved its convergence [12]. In [13], Jing et al. proposed a subspace clustering algorithm named EWKM by extending the *k*-means algorithm to include weight entropy in the objective function. In [14], Domeniconi et al. proposed the LAC algorithm, which is similar to EWKM. Kriegel et al. presented a comprehensive survey of high-dimensional data clustering, including subspace clustering [15]. In [16], Deng et al. extended the EWKM algorithm to a new subspace clustering algorithm named EEW-SC by considering between-cluster separation. In [17], Favaro et al. treated the subspace clustering problem as a rank minimization problem and proposed a closed-form solution. Müller et al. studied the scalability issue of clustering high-dimensional data and proposed a density-based subspace clustering algorithm [18]. Elhamifar and Vidal presented a sparse subspace clustering (SSC) algorithm using the idea of sparse representation [19]. The correctness of the SSC algorithm was proved by Soltanolkotabi et al. [20]. Timmerman et al. proposed a subspace *k*-means algorithm by modeling the centers and cluster residuals in reduced spaces [21]. In [22], Mcwilliams and Montana proposed a predictive subspace clustering (PSC) algorithm by assuming that each cluster can be approximated well by a linear subspace estimated by a principal component analysis. Zhu et al.

* Corresponding author. Tel.: +1 860 486 4238; fax: +1 860 486 3919.
E-mail addresses: Guojun.Gan@uconn.edu (G. Gan), mng@math.hkbu.edu.hk (M.-P. Ng).

http://dx.doi.org/10.1016/j.patcog.2015.05.016
0031-3203/

proposed online subspace clustering algorithm to clustering data streams [23]. In [24], the authors proposed a subspace clustering algorithm based on affinity propagation.

The aforementioned subspace clustering algorithms can be divided into two categories: hard subspace clustering and soft subspace clustering. A hard subspace clustering algorithm determines the exact subspaces in which clusters are embedded. A soft subspace clustering algorithm assigns weights to features and identify subspaces with large weights. One major challenge of the soft subspace clustering algorithms mentioned above is that the individual feature weights are sensitive to noise and missing values. To address this problem, Chen et al. introduced the idea of assigning weights to feature groups and proposed a new subspace clustering algorithm called FG-*k*-mean [25]. The FG-*k*-means algorithm is shown to outperform the *k*-means algorithm and several other subspace clustering algorithms such as W-*k*-means [11], LAC [14], and EWKM [13].

However, the FG-*k*-means algorithm requires that the feature groups are determined before the data is clusterized. In many cases, we do not know the group information of the features that describe a high-dimensional dataset. In this paper, we propose a subspace clustering algorithm, referred to as AFG-*k*-means, that is able to determine the feature groups automatically during the clustering process. The AFG-*k*-means algorithm extends the *k*-means algorithm by incorporating automatic feature group selection.

The remaining of the paper is organized as follows. In Section 2, we review the FG-*k*-means algorithm. In Section 3, we present the new subspace clustering algorithm, i.e., the AFG-*k*-means algorithm. In Section 4, we demonstrate the performance of the AFG-*k*-means algorithm using both synthetic data and real data. Section 5 concludes the paper with some remarks.

2. Related work

In this section, we give a brief introduction to the FG-*k*-means algorithm [25]. To describe these algorithms, we let $X=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$ be a dataset of n points, each of which is described by a set of m features: $A=\{A_1,A_2,\ldots,A_m\}$.

In the FG-*k*-means algorithm, the features that describe the high dimensional data are divided into feature groups, each of which is associated with a group weight. Within a feature group, each feature is also associated with a feature weight. The two types of weights are updated in the clustering process to identify important feature groups and individual features in each cluster.

Suppose that the set of features is divided into T groups $G=\{G_1,G_2,\ldots,G_T\}$ such that $G_t\neq\varnothing$, $G_t\cap G_s=\varnothing$ for $1\le t,s\le T$, $t\neq s$, and $\bigcup_{t=1}^{T}G_t=A$. To cluster X into k clusters, the FG-*k*-means algorithm minimizes the following objective function:

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

subject to the following conditions:

$$\sum_{l=1}^{k}u_{il}=1,\quad i=1,2,\ldots,n,\ \ u_{il}\in\{0,1\} \tag{2a}$$

$$\sum_{t=1}^{T}w_{lt}=1,\quad l=1,2,\ldots,k,\ \ w_{lt}>0 \tag{2b}$$

$$\sum_{j\in G_t}v_{lj}=1,\quad l=1,2,\ldots,k,\ \ t=1,2,\ldots,T,\ \ v_{lj}>0, \tag{2c}$$

where $U=(u_{il})_{n\times k}$ is a hard partition matrix, $Z=\{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_k\}$ is a set of k cluster centers, $V=(v_{lj})_{k\times m}$ and $W=(w_{lt})_{k\times T}$ are the two weight matrices mentioned before, λ and η are two positive parameters, and $d(x_{ij},z_{lj})$ is a distance measure between the i-th object and the center of the l-th cluster in the j-th feature. If the j-th feature is numeric, the distance measure is the square Euclidean distance. If the j-th feature is categorical, the distance measure is just the simple matching distance.

In the FG-*k*-means algorithm, the objective function given in Eq. (1) is optimized as follows. Given $Z=\hat{Z}$, $V=\hat{V}$, and $W=\hat{W}$, the hard partition matrix U that minimizes the objective function is given by

$$u_{il}=\begin{cases}1 & \text{if } D_{il}\le D_{is} \text{ for } 1\le s\le k;\\ 0 & \text{if otherwise,}\end{cases} \tag{3}$$

where $D_{is}=\sum_{t=1}^{T}\hat{w}_{st}\sum_{j\in G_t}\hat{v}_{sj}d(x_{ij},\hat{z}_{sj})$. Given $U=\hat{U}$, $V=\hat{V}$, and $W=\hat{W}$, the set Z of cluster centers that minimizes the objective function is given by

$$z_{lj}=\frac{\sum_{i=1}^{n}\hat{u}_{il}x_{ij}}{\sum_{i=1}^{n}\hat{u}_{il}}. \tag{4}$$

Given $U=\hat{U}$, $Z=\hat{Z}$, and $W=\hat{W}$, the weight matrix V that minimizes the objective function is given by

$$v_{lj}=\frac{\exp\left(-\frac{E_{lj}}{\eta}\right)}{\sum_{h\in G_t}\exp\left(-\frac{E_{lh}}{\eta}\right)}, \tag{5}$$

where $E_{lj}=\sum_{i=1}^{n}\hat{u}_{il}\hat{w}_{lt}d(x_{ij},\hat{z}_{lj})$ with t being the index of the feature group to which the j-th feature is assigned, i.e., $A_j\in G_t$. Given $U=\hat{U}$, $Z=\hat{Z}$, and $V=\hat{V}$, the weight matrix W that minimizes the objective function is given by

$$w_{lt}=\frac{\exp\left(-\frac{F_{lt}}{\lambda}\right)}{\sum_{s=1}^{T}\exp\left(-\frac{F_{ls}}{\lambda}\right)}, \tag{6}$$

where $F_{lt}=\sum_{i=1}^{n}\hat{u}_{il}\sum_{j\in G_t}\hat{v}_{lj}d(x_{ij},\hat{z}_{lj})$.

Note that in the FG-*k*-means algorithm, the feature group G is given as an input. The feature group weights are automatically calculated by the algorithm.

3. The AFG-*k*-means algorithm

In this section, we present the AFG-*k*-means algorithm that incorporates automatic feature grouping in the clustering process. To describe the algorithm, we let $X=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$ be a dataset of n points, each of which is described by a set of m numerical features: $A=\{A_1,A_2,\ldots,A_m\}$. Let k be the desired number of clusters and let T be the desired number of feature groups.

The objective function of the AFG-*k*-means algorithm is defined as

$$Q(U,Z,W,G,V,\Gamma)=\sum_{l=1}^{k}\sum_{i=1}^{n}u_{il}\sum_{j=1}^{m}w_{lj}^2(x_{ij}-z_{lj})^2+\epsilon_1\sum_{l=1}^{k}\sum_{j=1}^{m}w_{lj}^2+\beta\left(\sum_{t=1}^{T}\sum_{j=1}^{m}g_{jt}\sum_{l=1}^{k}\gamma_{lt}^2(w_{lj}-v_{lt})^2+\epsilon_2\sum_{l=1}^{k}\sum_{t=1}^{T}\gamma_{lt}^2\right), \tag{7}$$

where $U=(u_{il})_{n\times k}$ is an $n\times k$ matrix of binary numbers. If point $\mathbf{x}_i$ belongs to the l-th cluster, then $u_{il}=1$; otherwise, $u_{il}=0$. $Z=\{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_k\}$ is a set of k cluster centers. $W=(w_{lj})_{k\times m}$ is a $k\times$

m matrix of positive real numbers that satisfies the following conditions:

$$\sum_{j=1}^{m} w_{lj} = m, \quad l = 1, 2, \ldots, k. \tag{8}$$

$G = (g_{jt})_{m \times T}$ is an $m \times T$ matrix of binary numbers. If the j-th feature belongs to the t-th group, then $g_{jt} = 1$; otherwise, $g_{jt} = 0$. $V = (v_{lt})_{k \times T}$ is a $k \times T$ matrix of real numbers. $\Gamma = (\gamma_{lt})_{k \times T}$ is a $k \times T$ matrix of positive real numbers that satisfies the following conditions:

$$\sum_{l=1}^{k} \gamma_{lt} = k, \quad t = 1, 2, \ldots, T. \tag{9}$$

ϵ_1 is a nonnegative regularization constant. ϵ_2 is a nonnegative regularization constant. β is a nonnegative constant. x_{ij} denotes the value of $\mathbf{x}_i$ in the j-th feature. z_{lj} denotes the value of $\mathbf{z}_l$ in the j-th feature. The AFG-k-means algorithm minimizes this objective function to find optimal values for U, Z, W, G, V, and Γ. Comparing the objective function of the FG-k-means algorithm given in Eq. (1) and that of the AFG-k-means algorithm given in Eq. (7), we see that the AFG-k-means algorithm includes a component in its objective function to group features by the individual feature weights. In the AFG-k-means algorithm, two features are assigned to the same group if the individual feature weight patterns are similar.

Here the second component of the objective function is the objective function of the FSC algorithm for the dataset consisting of feature weights:

$$\begin{pmatrix} w_{11} & w_{21} & \cdots & w_{k1} \\ w_{12} & w_{22} & \cdots & w_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1m} & w_{2m} & \cdots & w_{km} \end{pmatrix}. \tag{10}$$

which has m points described by k features.

The objective function given in Eq. (7) can be minimized iteratively as follows. At the beginning of the algorithm, we initialize Z by selecting randomly k points from the dataset X and initialize W by equal values, i.e., $w_{lj} = 1$ for $l = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, m$. Since the weight matrix W has equal values, the objective function value is independent of G. After initialization, the algorithm proceeds to minimize the objective function according to the following theorems.

Theorem 1 (*Update U*). *Given $Z = \hat{Z}$, $W = \hat{W}$, $G = \hat{G}$, $V = \hat{V}$, and $\Gamma = \hat{\Gamma}$, the partition matrix U that minimizes the objective function given in Eq. (7) is given by*

$$u_{il} = \begin{cases} 1 & \text{if } D_{il} \le D_{ih} \quad \text{for all } h = 1, 2, \ldots, k, \\ 0 & \text{if otherwise}, \end{cases} \tag{11}$$

where $D_{il} = \sum_{j=1}^{m} \hat{w}_{lj}^2 (x_{ij} - \hat{z}_{lj})^2$, $i = 1, 2, \ldots, n$, $l = 1, 2, \ldots, k$.

Theorem 1 says that point $\mathbf{x}_i$ should be assigned to a cluster such that $\mathbf{x}_i$ is closest to the cluster's center. The proof of Theorem 1 is straightforward.

Theorem 2 (*Update Z*). *Given $U = \hat{U}$, $W = \hat{W}$, $G = \hat{G}$, $V = \hat{V}$, and $\Gamma = \hat{\Gamma}$, the set Z of cluster centers that minimizes the objective function given in Eq. (7) is given by*

$$z_{lj} = \frac{\sum_{i=1}^{n} \hat{u}_{il} x_{ij}}{\sum_{i=1}^{n} \hat{u}_{il}}, \quad i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, m. \tag{12}$$

From Theorem 2 we see that updating the cluster centers is similar to that of the k-means algorithm. Updating the cluster centers is independent of the weight matrix W, feature membership matrix G, and feature center V.

Theorem 3 (*Update W*). *Given $U = \hat{U}$, $Z = \hat{Z}$, $G = \hat{G}$, $V = \hat{V}$, and $\Gamma = \hat{\Gamma}$, the weight matrix W that minimizes the objective function given in Eq. (7) is given by*

$$\begin{aligned} w_{lj} &= \frac{\beta \sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 \hat{v}_{lt} - \frac{1}{2}\lambda_l}{\beta \sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 + E_{lj}} \\ &= \frac{\beta \sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 \hat{v}_{lt}}{\beta \sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 + E_{lj}} \\ &\quad - \frac{-m + \sum_{h=1}^{m} \left(\beta \sum_{t=1}^{T} \hat{g}_{ht} \gamma_{lt}^2 + E_{lh}\right)^{-1} \beta \sum_{t=1}^{T} \hat{g}_{ht} \gamma_{lt}^2 \hat{v}_{lt}}{\left(\beta \sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 + E_{lj}\right) \sum_{h=1}^{m} \left(\beta \sum_{t=1}^{T} \hat{g}_{ht} \gamma_{lt}^2 + E_{lh}\right)^{-1}} \end{aligned} \tag{13}$$

where

$$E_{lj} = \epsilon_1 + \sum_{i=1}^{n} \hat{u}_{il} (x_{ij} - \hat{z}_{lj})^2, \quad l = 1, 2, \ldots, k, \ j = 1, 2, \ldots, m, \tag{14}$$

and

$$\lambda_l = \frac{-2m + 2 \sum_{h=1}^{m} \left(\beta \sum_{t=1}^{T} \hat{g}_{ht} \gamma_{lt}^2 + E_{lh}\right)^{-1} \beta \sum_{t=1}^{T} \hat{g}_{ht} \gamma_{lt}^2 \hat{v}_{lt}}{\sum_{h=1}^{m} \left(\beta \sum_{t=1}^{T} \hat{g}_{ht} \gamma_{lt}^2 + E_{lh}\right)^{-1}} \tag{15}$$

for $l = 1, 2, \ldots, k$.

Proof. To prove Theorem 3, we use the method of Lagrange multiplier. To minimize the objective function given in Eq. (7) subject to the constraint given in Eq. (8), we minimize the following unconstrained objective function:

$$\begin{aligned} &Q_1(\hat{U}, \hat{Z}, W, \hat{G}, \hat{V}, \hat{\Gamma}, \lambda) \\ &\quad = \sum_{l=1}^{k} \sum_{i=1}^{n} \hat{u}_{il} \sum_{j=1}^{m} w_{lj}^2 (x_{ij} - \hat{z}_{lj})^2 + \epsilon_1 \sum_{l=1}^{k} \sum_{j=1}^{m} w_{lj}^2 \\ &\qquad + \beta \sum_{t=1}^{T} \sum_{j=1}^{m} \hat{g}_{jt} \sum_{l=1}^{k} \hat{\gamma}_{lt}^2 (w_{lj} - \hat{v}_{lt})^2 + \beta \epsilon_2 \sum_{l=1}^{k} \sum_{t=1}^{T} \hat{\gamma}_{lt}^2 \\ &\qquad + \sum_{l=1}^{k} \lambda_l \left(\sum_{j=1}^{m} w_{lj} - m \right), \end{aligned} \tag{16}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$. By fixing l, taking derivative of Q_1 with respect to w_{l1}, w_{l2}, …, w_{lm}, and λ_l, and equating the derivatives to zero, we get

$$\begin{aligned} \frac{\partial Q_1}{\partial w_{lj}} &= \sum_{i=1}^{n} \hat{u}_{il} 2 w_{lj} (x_{ij} - \hat{z}_{lj})^2 + 2\epsilon_1 w_{lj} \\ &\quad + 2\beta \sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 (w_{lj} - \hat{v}_{lt}) + \lambda_l = 0, \quad j = 1, 2, \ldots, m \end{aligned} \tag{17}$$

and

$$\frac{\partial Q_1}{\partial \lambda_l} = \sum_{j=1}^{m} w_{lj} - m = 0. \tag{18}$$

Solving the above linear equation system, we obtain

$$w_{lj} = \frac{2\beta \sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 \hat{v}_{lt} - \lambda_l}{2\beta \sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 + 2E_{lj}}, \tag{19}$$

and

$$\sum_{j=1}^{m} \frac{2\beta \sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 \hat{v}_{lt} - \lambda_l}{2\beta \sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 + 2E_{lj}} = m. \tag{20}$$

The results follow by rearranging the terms in the above equation. □

From Theorem 3 we see that the AFG-k-means algorithm reduces to the W-k-means algorithm [11] when $\beta \to 0$. When $\beta \to \infty$, we have

$$w_{lj} = \frac{\sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 \hat{v}_{lt}}{\sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2} - \frac{-m + \sum_{h=1}^{m} \left(\sum_{t=1}^{T} \hat{g}_{ht} \gamma_{lt}^2 \right)^{-1} \sum_{t=1}^{T} \hat{g}_{ht} \gamma_{lt}^2 \hat{v}_{lt}}{\left(\sum_{t=1}^{T} \hat{g}_{jt} \hat{\gamma}_{lt}^2 \right) \sum_{h=1}^{m} \left(\sum_{t=1}^{T} \hat{g}_{ht} \gamma_{lt}^2 \right)^{-1}} \tag{21}$$

which shows that features in the same group have the same weight.

Theorem 4 (*Update G*). *Given $U = \hat{U}$, $Z = \hat{Z}$, $W = \hat{W}$, $V = \hat{V}$, $\Gamma = \hat{\Gamma}$, and $\beta > 0$, the feature membership matrix G that minimizes the objective function given in Eq. (7) is given by*

$$g_{jt} = \begin{cases} 1 & \text{if } F_{jt} \le F_{js} \text{ for all } s = 1, 2, \ldots, T, \\ 0 & \text{if otherwise}, \end{cases} \tag{22}$$

where

$$F_{jt} = \sum_{l=1}^{k} \hat{\gamma}_{lt}^2 (\hat{w}_{lj} - \hat{v}_{lt})^2, \quad j = 1, 2, \ldots, m, \quad t = 1, 2, \ldots, T. \tag{23}$$

If $\beta = 0$, then G has no impact on the objective function. In this case, we just set G by assigning all features into one group.

Similar to Theorem 1, Theorem 4 can be proved straightforwardly. Theorem 4 shows that updating G depends only on W and V.

Theorem 5 (*Update V*). *Given $U = \hat{U}$, $Z = \hat{Z}$, $W = \hat{W}$, $\Gamma = \hat{\Gamma}$, and $\beta > 0$, the matrix V that minimizes the objective function given in Eq. (7) is given by*

$$v_{lt} = \frac{\sum_{j=1}^{m} \hat{g}_{jt} \hat{w}_{lj}}{\sum_{j=1}^{m} \hat{g}_{jt}}, \quad l = 1, 2, \ldots, k, \quad t = 1, 2, \ldots, T. \tag{24}$$

If $\sum_{j=1}^{m} \hat{g}_{jt} = 0$ (i.e., the t-th feature group is empty*), then we set $V_{lt} = 0$. If $\beta = 0$, then V has no impact on the objective function. In this case, we also set all elements of V to be zero.*

Table 1
Default values for some parameters required by the AFG-k-means algorithm.

Parameter	Default Value
β	1
ϵ_1	0.0001
ϵ_2	0.0001
N_{max}	100
δ	10^{-6}

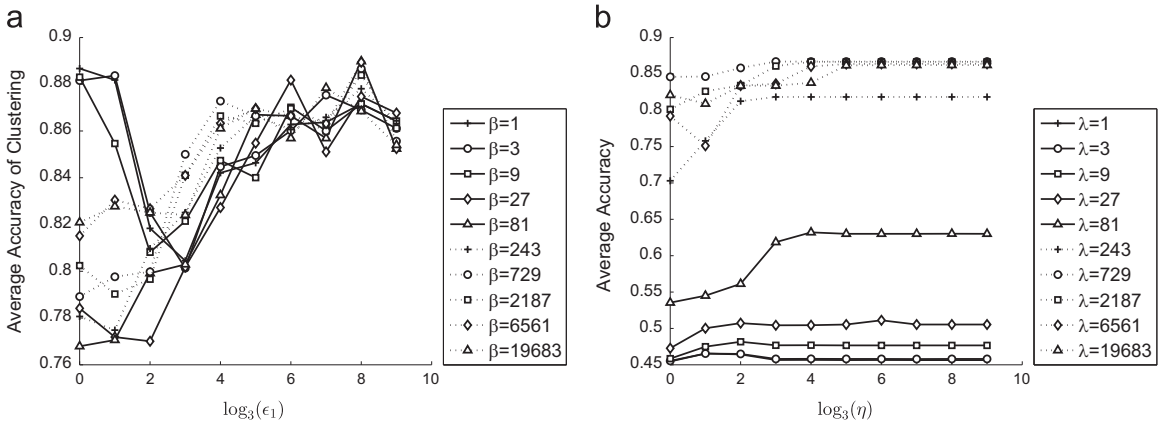

Fig. 1. The average accuracy of 100 runs of the AFG-k-means algorithm and the FG-k-means algorithm on the first synthetic dataset. (a) The average corrected Rand indices of the AFG-k-means algorithm with $k=3$, $T=3$, and default values for other parameters. (b) The average corrected Rand indices of the FG-k-means algorithm with the corrected feature groups as input.

Table 2
The results of a single run of the AFG-k-means algorithm on the first synthetic dataset. (a) The confusion matrix of the given clusters and the found clusters. (b) The confusion matrix of the given feature groups and the found feature groups. (c) The feature group centers V. (d) The feature group weights Γ. In this run, we used $k=3$, $T=3$, $\beta=3$, and default values for other parameters.

	A	B	C
1	0	2000	0
2	2000	0	0
3	0	0	1000

(a)

	G_1	G_2	G_3
1	0	40	0
2	40	0	0
3	0	0	120

(b)

	Feature	Group	Centers, V
1	0.7277	4.1622	0.0367
2	0.2718	4.4024	0.1086
3	4.6651	0.1182	0.0722

(c)

	Feature	Group	Centers, Γ
1	0.3441	0.006	2.5231
2	2.6521	0.0035	0.20831
3	0.0038	2.9906	0.2685

(d)

Theorem 6 (*Update Γ*). *Given $U=\hat{U}$, $Z=\hat{Z}$, $W=\hat{W}$, $G=\hat{G}$, $V=\hat{V}$, and $\beta>0$, the weight matrix Γ that minimizes the objective function given in Eq. (7) is given by*

$$\gamma_{lt}=\frac{k}{\sum_{s=1}^{k}\frac{H_{lt}}{H_{st}}}, \tag{25}$$

where

$$H_{lt}=\epsilon_2+\sum_{j=1}^{m}\hat{g}_{jt}(\hat{w}_{lj}-\hat{v}_{lt})^2,\quad l=1,2,\ldots,k,\ \ t=1,2,\ldots,T. \tag{26}$$

If $\beta=0$, then Γ has no impact on the objective function. In this case, we set all elements of Γ to be one.

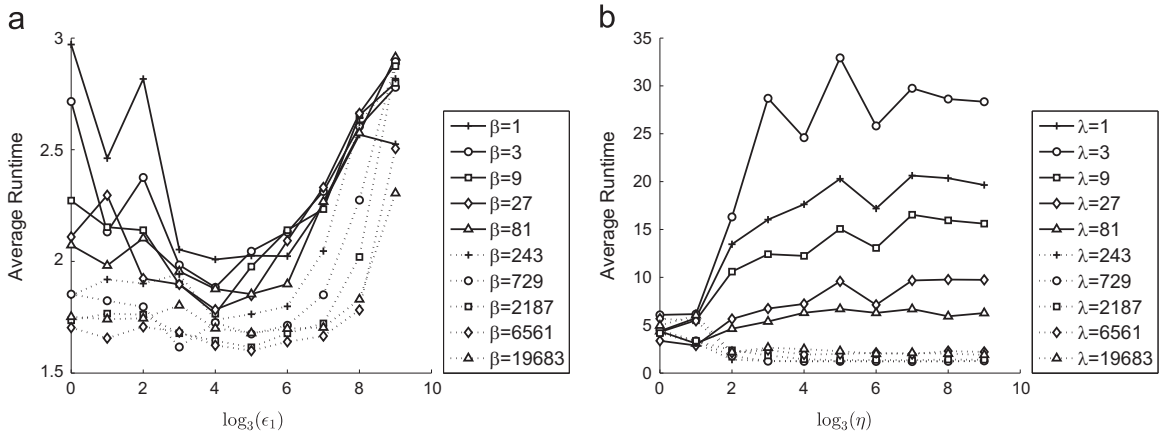

Fig. 2. The average speed of 100 runs of the AFG-*k*-means algorithm and the FG-*k*-means algorithm on the first synthetic dataset with various parameter values. (a) The average runtime (in seconds) of the AFG-*k*-means algorithm with $k=3$, $T=3$, and default values for other parameters. (b) The average runtime (in seconds) of the FG-*k*-means algorithm with the corrected feature groups as input.

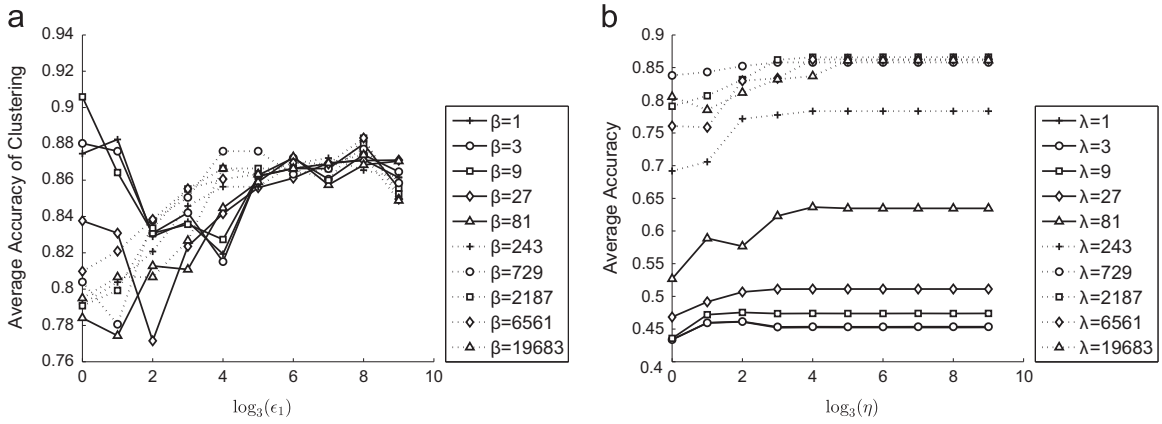

Fig. 3. The average accuracy of 100 runs of the AFG-*k*-means algorithm and the FG-*k*-means algorithm on the second synthetic dataset with various parameter values. (a) The average corrected Rand indices of the AFG-*k*-means algorithm with $k=3$, $T=3$, and default values for other parameters. (b) The average corrected Rand indices of the FG-*k*-means algorithm with the correct feature groups as input.

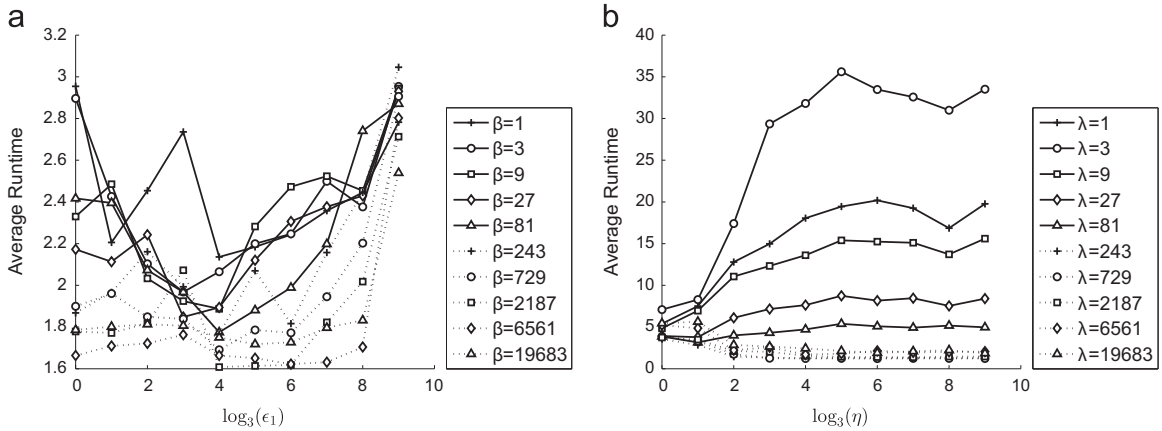

Fig. 4. The average speed of 100 runs of the AFG-*k*-means algorithm and the FG-*k*-means algorithm on the second synthetic dataset with various parameter values. (a) The average runtime (in seconds) of the AFG-*k*-means algorithm with $k=3$, $T=3$, and default values for other parameters. (b) The average runtime (in seconds) of the FG-*k*-means algorithm with the corrected feature groups as input.

The pseudo-code of the AFG-k-means algorithm is shown in Algorithm 1. The inputs to the algorithm include a dataset and several parameters, which are the number of clusters, the number of feature groups, β, ϵ_1, ϵ_2, N_{max}, and δ. The last two parameters are used to terminate the algorithm. The parameter N_{max} is the maximum number of iterations. The parameter δ is a small positive constant. If the absolute change of the values of the first objective function is less than δ, the algorithm is terminated. We can choose the desired number of clusters and the number of feature groups for the underlying dataset. For the other parameters (i.e., β, ϵ_1, ϵ_2, N_{max}, and δ), we can use the default values given in Table 1.

Algorithm 1. The AFG-k-means algorithm.

Input: X, k, T, β, ϵ_1, ϵ_2, N_{max}, δ
Output: Optimal values of U, Z, W, G, and V

Initialize $Z^{(0)}$ by selecting k points from X randomly;
Set all initial elements in $W^{(0)}$, $\Gamma^{(0)}$, and $V^{(0)}$ to one;
Update $U^{(0)}$ according to Theorem 1;
Initialize $G^{(0)}$ by assigning all features into one group;
$s \leftarrow 0$;
$Q^{(0)} \leftarrow 0$;
While *True* **do**
  Update $Z^{(s+1)}$ according to Theorem 2;
  Update $U^{(s+1)}$ according to Theorem 1;
  Update $W^{(s+1)}$ according to Theorem 3;
  if $s = 0$ **then**
    Initialize$V^{(1)}$ by selecting T columns from$W$ randomly;
  else
    Update $V^{(s+1)}$ according to Theorem 5;
  end
  Update $G^{(s+1)}$ according to Theorem 4;
  Update $\Gamma^{(s+1)}$ according to Theorem 6;
  $s \leftarrow s+1$;
  $Q^{(s+1)} \leftarrow Q\left(U^{(s+1)}, Z^{(s+1)}, W^{(s+1)}, G^{(s+1)}, V^{(s+1)}\right)$;
  if $|Q^{(s+1)} - Q^{(s)}| < \delta$ or $s \geq N_{max}$ **then**
    Break;
  end
end

4. Numerical experiments

In this section, we present some numerical experiments to demonstrate the performance of the AFG-k-means algorithm in terms of discovering subspace clusters and identifying feature groups associated with them. We use both synthetic data and real data in these experiments.

4.1. Experiments on synthetic data

In this subsection, we use synthetic data to demonstrate the performance of the AFG-k-means algorithm.

4.1.1. Synthetic data generation

We follow the idea given in [25] to generate subspace clusters in feature groups. Suppose that we want to generate a dataset that contains n points in an m-dimensional space and has k subspace clusters in T feature groups. Let $A = (a_{lt})_{k\times T}$ a matrix of real numbers and let $B = (b_{lt})_{k\times T}$ be a matrix of positive real numbers. Let $G = (g_{jt})_{m\times T}$ is a binary matrix indicating the feature groups. Let $U = (u_{il})_{n\times k}$ is a binary matrix indicating the clusters. Then we can generate n data points as follows:

$$x_{ij} = \sum_{l=1}^{k} u_{il} \sum_{t=1}^{T} g_{jt}(a_{lt} + R_{ij} b_{lt}), \tag{27}$$

where R_{ij} is a random number generated from the standard normal distribution. Once we have the dataset, then we normalize dataset such that each dimension has a standard deviation of one.

To generate a synthetic dataset using this method, we only need to specify n, m, G, U, A, and B. From Eq. (27) we see that each cluster may have different centers and that the points in each cluster can have different dispersions in different feature groups. We can change the input standard deviation matrix B to generate subspace clusters in feature groups.

4.1.2. Results

We use the aforementioned method to generate two synthetic datasets. The first dataset is generated with the following parameters: $n = 5000$, $m = 200$,

$$G = \{\{1,\ldots,40\},\{41,\ldots,80\},\{81,\ldots,200\}\},$$
$$U = \{\{1,2,\ldots,2000\},\{2001,\ldots,4000\},\{4001,\ldots,5000\}\},$$
$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 20 & 0 \\ 20 & 0 & 0 \end{pmatrix}, \quad \text{and} \quad B = \begin{pmatrix} 1 & 5 & 3 \\ 1 & 3 & 5 \\ 5 & 1 & 3 \end{pmatrix}. \tag{28}$$

Here we use sets to represent the binary matrices G and U to save spaces. The first dataset contains 5000 points, each of which is described by 200 features. The features are divided into 3 groups. The dataset contains three clusters, which contain 2000, 2000, and 1000 points. The first cluster has relatively small dispersions in the first and the third feature groups. The third cluster has relatively small dispersions in the second and the third feature groups.

To test the impact of the parameters β and ϵ_1 on the performance of the AFG-k-means algorithm, we applied the AFG-k-means algorithm to the first synthetic data with $\beta = 3^i$ and $\epsilon_1 = 3^j$ for $i,j = 0,1,\ldots,9$. For each combination of these values, we run the algorithm 100 times with different seeds used to initialize cluster centers and feature groups randomly. For comparison purpose, we also applied the FG-k-means to the first synthetic dataset with $\lambda = 3^i$ and $\eta = 3^j$ for $i,j = 0,1,\ldots,9$. The average accuracy of the two algorithms on the first synthetic dataset is shown in Fig. 1. From Fig. 1, we see that the AFG-k-means algorithm produces more accurate results than the FG-k-means algorithm does. In addition, the AFG-k-means algorithm is less sensitive to its parameters than the FG-k-means algorithm does. As we can see from Fig. 1, the FG-k-means algorithm produced more accurate results when the parameter λ was larger. If λ is small, then the feature group weight is dominated by the feature group with the smallest dispersion due to the property of exponential normalization. If λ is large, the feature group weights become approximately the same.

Table 2 shows the results of a single run of the AFG-k-means algorithm on the first synthetic data with $\beta = 3$. From the first two tables we see that the AFG-k-means algorithm recovered the clusters and the feature groups correctly. From the third table, we see that the relative magnitudes of the feature group centers in each column match inversely with the relative magnitudes of the corresponding standard deviations in B (see Eq. (28)). For example, the first column of V is $(0.7277, 0.2718, 4.6651)^T$. From the first two tables, we know that the first number 0.7277 corresponds to $b_{22} = 3$, the second number 0.2718 corresponds to $b_{12} = 5$, and the third number 4.6651 corresponds to $b_{32} = 1$. From the fourth table, we see that the relative magnitudes of the feature group weights

in each column match the relative magnitudes of the corresponding standard deviations in B.

The average speed of the two algorithms on the first synthetic data is shown in Fig. 2, from which we see that the AFG-k-means algorithm converged much faster than the FG-k-means algorithm did. In addition, the average runtime of the FG-k-means algorithm is also sensitive to the parameters λ and η. In the two algorithms, we used the same criteria to terminate the iterative process.

To test the performance of the AFG-k-means algorithm on noise data, we created the second synthetic data by adding normal noise to 20% of the components of the first synthetic dataset. The noise was generated from the standard normal distribution. The average accuracy of the AFG-k-means algorithm and the FG-k-means algorithm on the second synthetic dataset is shown in Fig. 3. Comparing Figs. 1 and 3, we see that the average accuracy of the two algorithm is not affected by the additional noises.

The average runtime of the AFG-k-means algorithm and the FG-k-means algorithm on the second dataset are summarized in Fig. 4. Comparing Figs. 2 and 4, we see that the average runtime of the AFG-k-means algorithm is not affected by the additional noises. However, the average runtime of the FG-k-means algorithm increased a little bit.

We also tested the FG-k-means algorithm on the synthetic datasets with incorrect input of feature groups. In particular, we divided the features into three groups randomly in each run of the FG-k-means algorithm. The average accuracy and speed are shown in Figs. 5 and 6. From these figures we see that when incorrect feature groups are input to the FG-k-means algorithm, the accuracy of the clustering results depends on the parameter η. In general, the larger the parameter η, the more accurate the results. To reduce the effect of wrong feature groups, the FG-k-means algorithm requires larger values for η to make the individual feature weights more uniform.

In summary, the experiments on synthetic data show that the AFG-k-means algorithm produces more accurate results than the FG-k-means algorithm does and the AFG-k-means algorithm is less sensitive to its parameters than the FG-k-means is.

4.2. Experiments on real data

To compare the AFG-k-means algorithm and the FG-k-means algorithm on real data, we obtained two gene expression datasets

Table 3
Two real gene expression datasets. The first real dataset has three known clusters and the second real dataset contains two clusters.

Dataset	Samples	Attributes	Cluster sizes
Alizadeh-2000-v2	62	2093	42, 9, 11
Gordon-2002	181	1626	31, 150

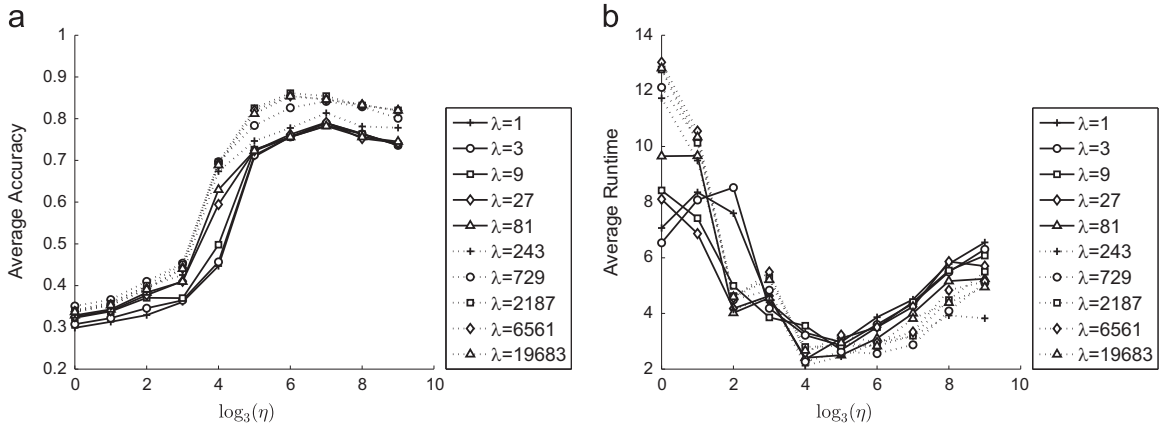

Fig. 5. The average accuracy and speed of 100 runs of the FG-k-means algorithm on the first synthetic dataset with various parameter values. In these runs, three feature groups were randomly created. (a) The average corrected Rand indices. (b) The average runtime (in seconds).

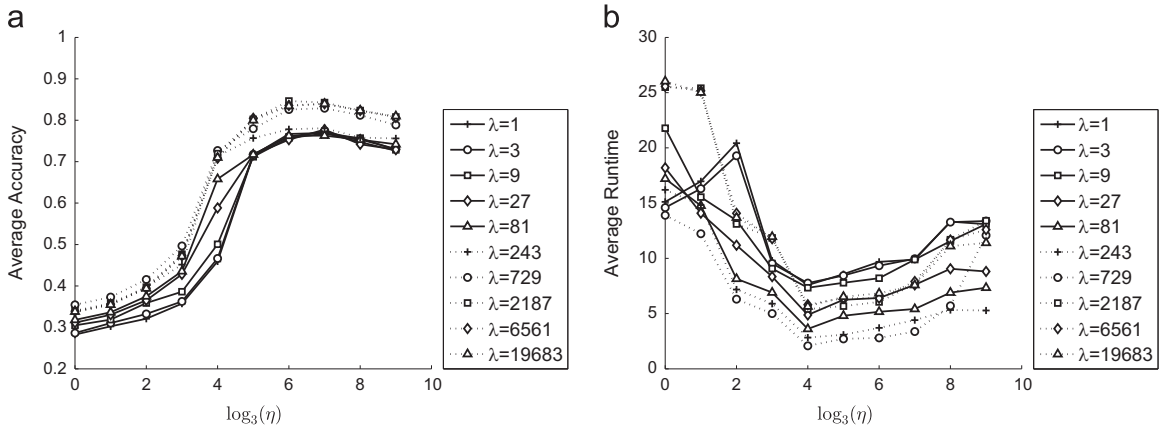

Fig. 6. The average accuracy and speed of 100 runs of the FG-k-means algorithm on the second synthetic dataset with various parameter values. In these runs, three feature groups were randomly created. (a) The average corrected Rand indices. (b) The average runtime (in seconds).

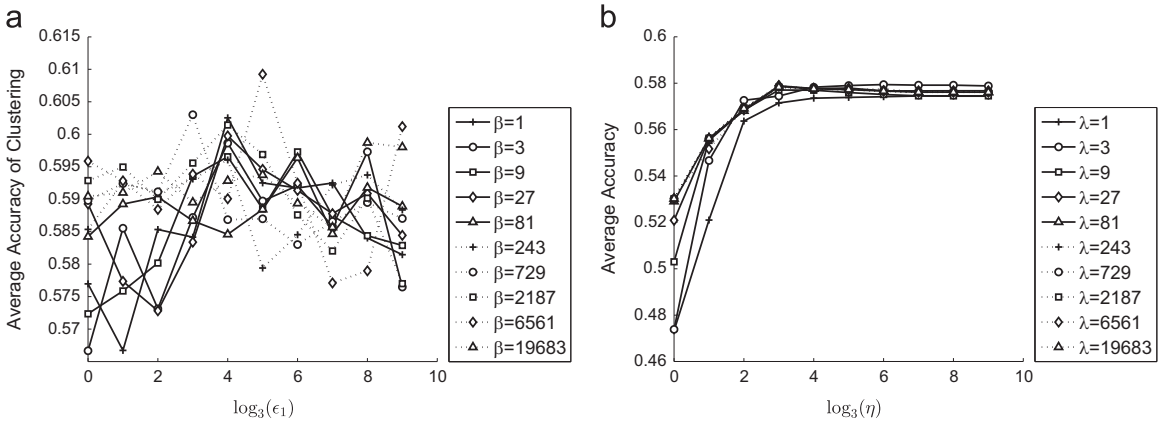

Fig. 7. The average accuracy of 100 runs of the AFG-*k*-means algorithm and the FG-*k*-means algorithm on the Alizadeh-2000-v2 dataset with various parameter values. (a) The average corrected Rand indices of the AFG-*k*-means algorithm with $k=3$, $T=3$, and default values for other parameters. (b) The average corrected Rand indices of the FG-*k*-means algorithm with random feature groups as input.

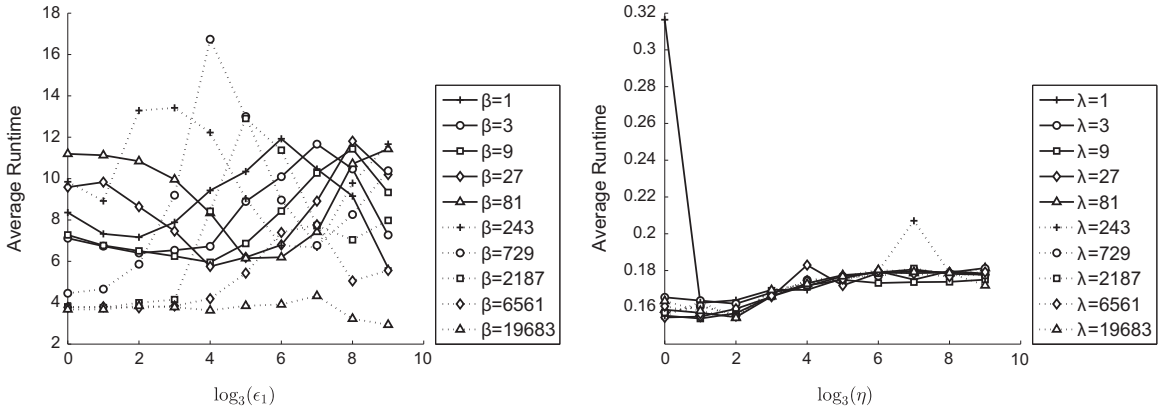

Fig. 8. The average speed of 100 runs of the AFG-*k*-means algorithm and the FG-*k*-means algorithm on the Alizadeh-2000-v2 dataset with various parameter values. (a) The average runtime (in seconds) of the AFG-*k*-means algorithm with $k=3$, $T=3$, and default values for other parameters. (b) The average runtime (in seconds) of the FG-*k*-means algorithm with random feature groups as input.

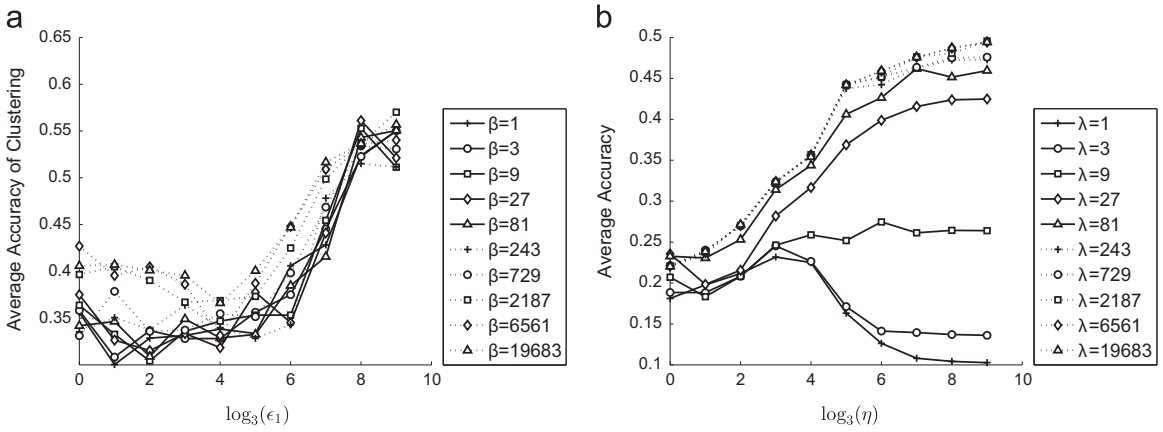

Fig. 9. The average accuracy of 100 runs of the AFG-*k*-means algorithm and the FG-*k*-means algorithm on the Gordon-2002 dataset with various parameter values. (a) The average corrected Rand indices of the AFG-*k*-means algorithm with $k=2$, $T=3$, and default values for other parameters. (b) The average corrected Rand indices of the FG-*k*-means algorithm with the random feature groups as input.

from [3][1]: the gene expression data from adult lymphoid malignancies and the gene expression data from the lung cancer. Both datasets have known labels. Table 3 shows the information of the two real datasets. Since different attributes of the real datasets have different ranges, we use the *z*-score method to normalize all the attributes before applying the two algorithms to the datasets.

We applied both the AFG-*k*-means algorithm and the FG-*k*-means algorithm to the Alizadeh-2000-v2 dataset with various parameter values. Unlike the synthetic datasets, the real datasets

[1] The two datasets are available at http://bioinformatics.rutgers.edu/Static/Supplements/CompCancer/datasets.htm.

do not have the feature group information. As a result, we use random feature groups in the FG-*k*-means algorithm. In particular, we randomly divide the attributes into three groups. The average accuracy of the two algorithm on the Alizadeh-2000-v2 dataset is summarized in Fig. 7. From the figure we see that the AFG-*k*-means algorithm produces slightly more accurate results than the

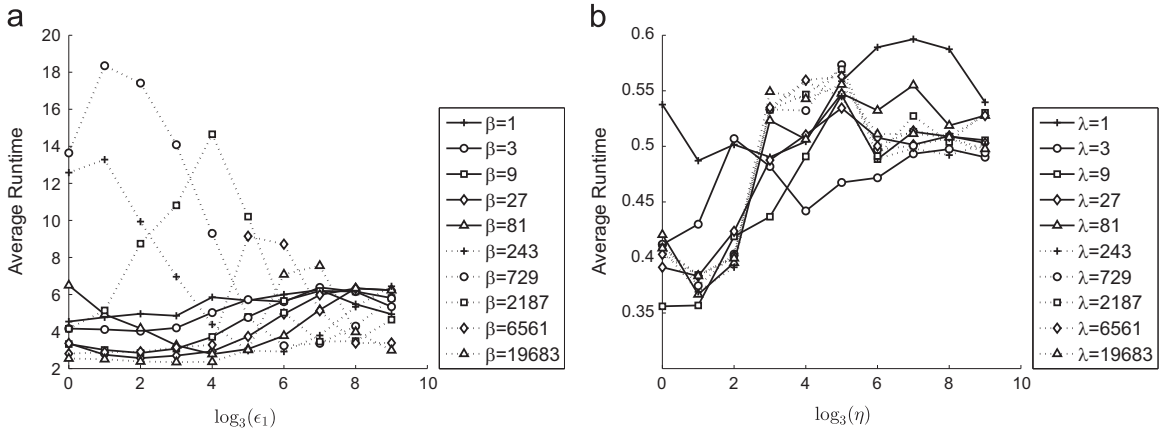

Fig. 10. The average speed of 100 runs of the AFG-*k*-means algorithm and the FG-*k*-means algorithm on the Gordon-2002 dataset with various parameter values. (a) The average runtime (in seconds) of the AFG-*k*-means algorithm with $k=2$, $T=3$, and default values for other parameters. (b) The average runtime (in seconds) of the FG-*k*-means algorithm with random feature groups as input.

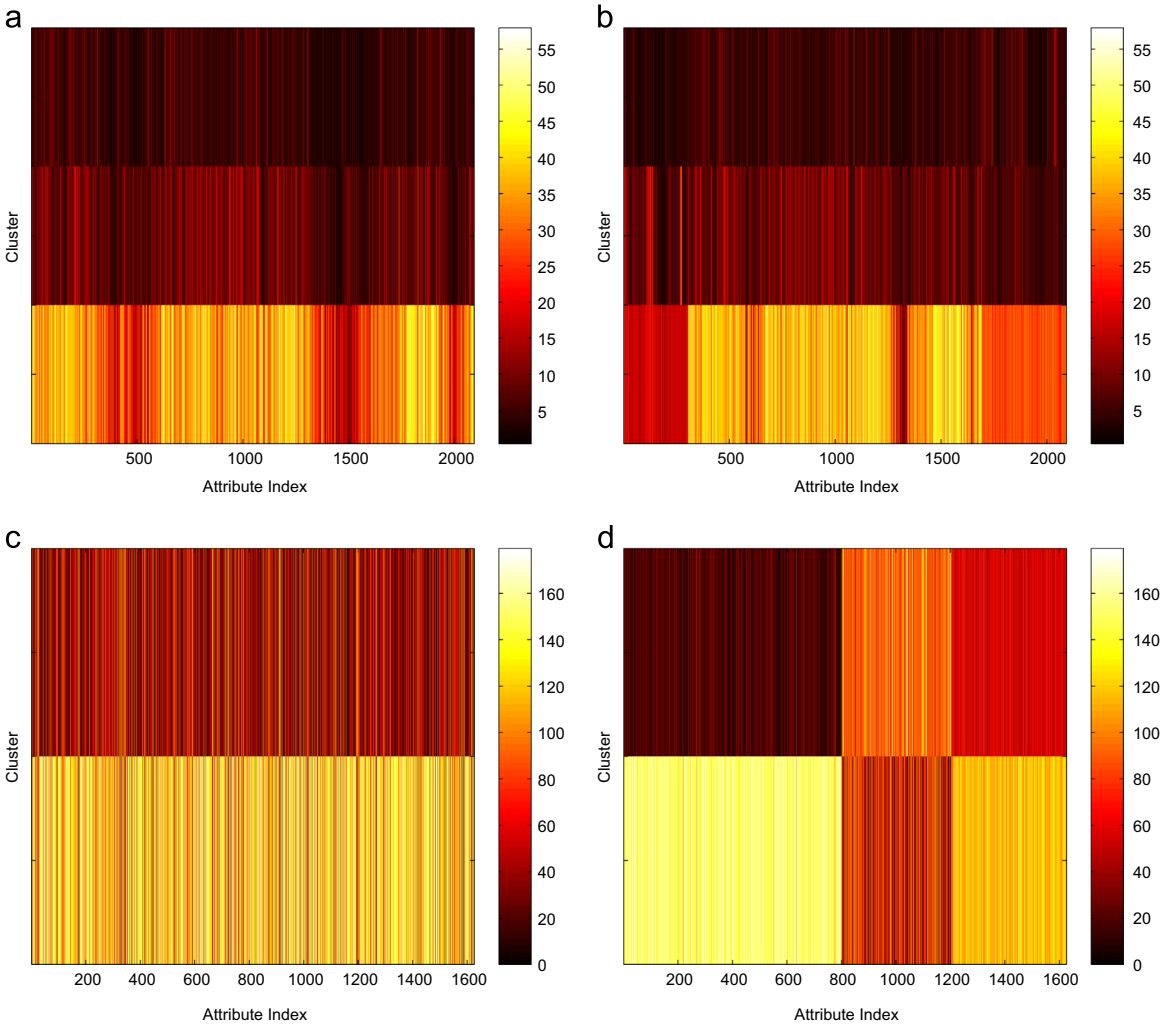

Fig. 11. The color maps of the data dispersions by cluster and attribute. (a) Alizadeh-2000-v2 with attributes in their original order. (b) Alizadeh-2000-v2 with attributes grouped by AFG-*k*-means. (c) Gordon-2002 with attributes in their original order. (d) Gordon-2002 with attributes grouped by AFG-*k*-means. All the four color maps are based on results produced by AFG-*k*-means with $T=3$, $\beta=729$, $\epsilon_1=27$, and $\epsilon_2=0.0001$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

FG-*k*-means algorithm does. In addition, the AFG-*k*-means algorithm is less sensitive to the parameters than the Fg-*k*-means algorithm is.

The average runtime of the AFG-*k*-means algorithm and the FG-*k*-means algorithm on the Alizadeh-2000-v2 dataset is shown in Fig. 8. From the figure we see that the AFG-*k*-means algorithm converged much slower than the FG-*k*-means algorithm did. The reason is the Alizadeh-2000-v2 dataset has 2093 features and only 62 records and it took time for the AFG-*k*-means algorithm to group the large number of features. For the FG-*k*-means algorithm, the features were randomly divided into 3 groups and the feature groups were the same during iterations. As a result, the FG-*k*-means algorithm runs much faster for small-sample, high-dimensional data.

For the Gordon-2002 dataset, the average accuracy of the two algorithms is shown in Fig. 9. From the figure we see that the AFG-*k*-means algorithm also outperformed the FG-*k*-means algorithm. The AFG-*k*-means algorithm produced more accurate results with large values of ϵ_1. Larger value of ϵ_1 lead to more uniform individual feature weights. For small λ, the FG-*k*-means algorithm produced much less accurate results. For this dataset, the FG-*k*-means algorithm is sensitive to both λ and η. Fig. 10 shows the average runtime of the two algorithms on the Gordon-2002 dataset. Since this dataset has 1626 attributes, the AFG-*k*-means algorithm also converged slower than the FG-*k*-means algorithm did.

Fig. 11 shows the color maps produced from the dispersions of the two real datasets. We calculate the dispersion of each attribute within each cluster and visualize these dispersions based on the original attribute order and the attribute groups. From the figures we can see that the AFG-*k*-means algorithm is able to cluster data and group features simultaneously.

The numerical results on real datasets show that the AFG-*k*-means algorithm outperforms the FG-*k*-means algorithm in terms of accuracy. However, the AFG-*k*-means algorithm converges slower than the FG-*k*-means algorithm because the real datasets have a large number of features and only a few records. Since the AFG-*k*-means algorithm groups both records as well as features during the iterative process, the runtime is dominated by the feature grouping where the number of features is much larger than the number of records.

Table 4
Eight datasets. The n, d, k refer to the number of data points, the number of attributes, and the number of clusters, respectively.

Dataset	n	d	k	Reference
S1	5000	200	3	The first synthetic data
S2	5000	200	3	The second synthetic data
R1	62	2093	3	Alizadeh-2000-v2 [3]
R2	181	1626	2	Gordon-2002 [3]
D2	6000	200	3	D2 [25]
MF	2000	649	10	Multiple features [25]
R3	72	1081	2	Armstrong-2002-v1 [3]
R4	104	182	2	Chowdary-2006 [3]

Table 5
Some parameter values of the five clustering algorithms used in the comparison.

Algorithm	Parameters
FSC	$\alpha=2$, $\epsilon=0.0001$
LAC	$h=729$
EWKM	$\gamma=729$
FG-*k*-means	$T=3$, $\lambda=729$, $\eta=729$
AFG-*k*-means	$T=3$, $\beta=1$, $\epsilon_1=0.0001$, $\epsilon_2=0.0001$

Table 6
Summary of clustering results on eight datasets by five algorithms. The accuracy is measured by the corrected Rand index and the runtime is measured in seconds. Numbers outside parenthesis are the mean values of 100 runs. Numbers in parenthesis are the standard deviations of 100 runs. Bold numbers indicate the best results.

Data	FSC	LAC	EWKM	FG-*k*-means	AFG-*k*-means
	Accuracy				
S1	0.67 (0.3)	0.86 (0.23)	0.75 (0.26)	0.88 (0.21)	**0.89** (0.23)
S2	0.75 (0.3)	0.84 (0.23)	0.73 (0.25)	0.89 (0.19)	**0.9** (0.22)
R1	0.5 (0.27)	0.59 (0.19)	**0.59** (0.2)	0.56 (0.19)	0.54 (0.22)
R2	0.5 (0.42)	0.57 (0.48)	0.53 (0.46)	0.54 (0.46)	**0.69** (0.39)
D2	**0.8** (0.25)	0.44 (0.09)	0.59 (0.09)	0.5 (0.01)	0.75 (0.25)
MF	0.13 (0.05)	**0.68** (0.07)	0.67 (0.07)	0.57 (0.06)	0.14 (0.05)
R3	0.18 (0.23)	0.32 (0.28)	0.32 (0.28)	0.28 (0.25)	**0.33** (0.32)
R4	0.54 (0.32)	0.06 (0.02)	0.06 (0.02)	0.06 (0.02)	**0.76** (0.24)
	Runtime				
S1	3.55 (4.86)	2.1 (2.85)	1.06 (0.59)	1.19 (1)	1.89 (2.54)
S2	2.96 (4.68)	2.47 (3.34)	1.03 (0.66)	1.12 (0.83)	1.71 (2.27)
R1	0.13 (0.07)	0.09 (0.03)	0.1 (0.03)	0.1 (0.04)	4.3 (1.95)
R2	0.31 (0.15)	0.27 (0.18)	0.31 (0.18)	0.3 (0.23)	2.31 (0.9)
D2	5.48 (5.28)	7.86 (2.99)	1.32 (0.19)	5.58 (0.6)	5.87 (4.87)
MF	2.72 (0.51)	8.65 (2.79)	8.83 (2.74)	12.73 (4.27)	12.7 (4.82)
R3	0.07 (0.02)	0.06 (0.02)	0.07 (0.03)	0.06 (0.03)	1.28 (0.74)
R4	0.02 (0.01)	0.01 (0.01)	0.01 (0.01)	0.02 (0.01)	0.04 (0.02)

4.3. Comparison with other algorithms

In this subsection, we compare AFG-*k*-means with FG-*k*-means, W-*k*-means, FSC, EWKM, and LAC using eight datasets summarized in Table 4. Since W-*k*-means and FSC are similar except for the regularization term, we only include FSC in our comparison. We already tested AFG-*k*-means and FG-*k*-means on the first four datasets with various parameter values.

Some parameter values of the five algorithm are given in Table 5. Except for the desired number of clusters *k*, which was set to the true number of clusters of a dataset, other parameters of an algorithm were the same for all datasets. We selected these parameter values based on the experiments we conducted in the previous two subsections. Since LAC, EWKM, and FG-*k*-means use exponentially normalized weights, we used relatively large values for their parameters in order to prevent one attribute or one feature group dominates the weights. For the FG-*k*-means algorithm, attributes were divided into three groups randomly for each run. We also normalized all datasets so that each attribute has a standard deviation of 1.

The average accuracy and runtime of the five algorithms on the eight datasets are summarized in Table 6, from which we see that the AFG-*k*-means algorithm produced the most accurate results in most cases. As expected, for small-sample and high-dimensional datasets (e.g., R1, R2, and R3), the AFG-*k*-means algorithm converged slower than other algorithms did.

5. Concluding remarks

In this paper, we proposed a subspace clustering algorithm with automatic feature grouping, called the AFG-*k*-means algorithm, based on the feature grouping idea of the FG-*k*-means algorithm proposed by [25]. In the FG-*k*-means algorithm, the feature groups are given as input. In our algorithm, the feature groups are automatically determined during the iterative process of the algorithm. The automatic feature grouping is achieved by introducing an additional component to the objective function of the FSC algorithm [12] and dynamically updating the feature groups during the iteration.

The experiments on both synthetic data and real data have shown that the AFG-k-means algorithm outperformed the FG-k-means algorithm in terms of accuracy and choice of parameters. The experiments on synthetic data have shown that the AFG-k-means algorithm is able to recover the clusters embedded in feature groups as well as the feature groups. The experiments on real gene expression data have shown that the AFG-k-means algorithm produces more accurate clustering results than the FG-k-means algorithm. In addition, the experiments show that the AFG-k-means algorithm is less sensitive to parameters than the FG-k-means algorithm is.

One drawback of the AFG-k-means algorithm is that the clustering results depend on initial cluster centers. This drawback is common to k-mean type algorithms including the FG-k-means algorithm. Another limitation of the AFG-k-means algorithm is that its objective function is restricted to the form of sum of squares. For example, if we use w_{lj}^{α} or $w_{lj}\log w_{lj}$ in the objective function, then we have no closed-form formulas to update the weights.

It is also worth to point out that FG-k-means and AFG-k-means are related to multi-view clustering [26,27]. The feature groups are similar to the views in multi-view clustering. In FG-k-means and multi-view clustering, the feature groups and views are known beforehand. In the proposed AFG-k-means algorithm, the feature groups or views are not known and recovering these feature groups and views is part of the clustering task.

Conflict of interest

None declared.

Acknowledgments

The authors would like to thank two referees for their insightful comments that greatly improve the organization and quality of the paper. The visit of Dr. Guojun Gan to Hong Kong Baptist University in March 2015 is supported in part by Centre for Mathematical Imaging and Vision.

References

[1] G. Gan, Data Clustering in C++: An Object-oriented Approach, Chapman & Hall/CRC Press, Boca Raton, FL, USA, 2011.
[2] C.C. Aggarwal, C.K. Reddy (Eds.), Data Clustering: Algorithms and Applications, CRC Press, Boca Raton, FL, USA, 2013.
[3] M. de Souto, I. Costa, D. de Araujo, T. Ludermir, A. Schliep, Clustering cancer gene expression data: a comparative study, BMC Bioinform. 9 (1) (2008) 1–14.
[4] A. Jain, R. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.
[5] N. Yilmaz, O. Inan, M. Uzer, A new data preparation method based on clustering algorithms for diagnosis systems of heart and diabetes diseases, J. Med. Syst. 38 (5) (2014) 1–12.
[6] G. Gan, Application of data clustering and machine learning in variable annuity valuation, Insur.: Math. Econ. 53 (3) (2013) 795–801.
[7] A. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[8] J. Macqueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.
[9] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, in: SIGMOD Record ACM Special Interest Group on Management of Data, ACM Press, New York, NY, USA, 1998, pp. 94–105.
[10] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, SIGKDD, Newsl. ACM Spec. Interest Group Knowl. Discov. Data Min. 6 (1) (2004) 90–105.
[11] J. Huang, M. Ng, H. Rong, Z. Li, Automated variable weighting in k-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 657–668.
[12] G. Gan, J. Wu, A convergence theorem for the fuzzy subspace clustering (FSC) algorithm, Pattern Recognit. 41 (6) (2008) 1939–1947.
[13] L. Jing, M. Ng, J. Huang, An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, IEEE Trans. Knowl. Data Eng. 19 (8) (2007) 1026–1041.
[14] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, D. Papadopoulos, Locally adaptive metrics for clustering high dimensional data, Data Min. Knowl. Discov. 14 (1) (2007) 63–97.
[15] H.-P. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering, ACM Trans. Knowl. Discov. from Data 3 (1) (2009) 1:1–1:58.
[16] Z. Deng, K.-S. Choi, F.-L. Chung, S. Wang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, Pattern Recognit. 43 (3) (2010) 767–781.
[17] P. Favaro, R. Vidal, A. Ravichandran, A closed form solution to robust subspace estimation and clustering, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1801–1807.
[18] E. Müller, I. Assent, S. Günnemann, T. Seidl, Scalable density-based subspace clustering, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, 2011, pp. 1077–1086.
[19] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
[20] M. Soltanolkotabi, E. Elhamifar, E.J. Candes, Robust subspace clustering, Ann. Stat. 42 (2) (2014) 669–699.
[21] M.E. Timmerman, E. Ceulemans, K.D. Roover, K.V. Leeuwen, Subspace k-means clustering, Behav. Res. Methods 45 (4) (2013) 1011–1023.
[22] B. Mcwilliams, G. Montana, Subspace clustering of high-dimensional data: a predictive approach, Data Min. Knowl. Discov. 28 (3) (2014) 736–772.
[23] L. Zhu, L. Cao, J. Yang, J. Lei, Evolving soft subspace clustering, Appl. Soft Comput. 14 (2014) 210–228.
[24] G. Gan, M.K.-P. Ng, Subspace clustering using affinity propagation, Pattern Recognit. 48 (4) (2015) 1451–1460.
[25] X. Chen, Y. Ye, X. Xu, J.Z. Huang, A feature group weighting method for subspace clustering of high-dimensional data, Pattern Recognit. 45 (1) (2012) 434–446.
[26] X. Zhao, N. Evans, J.-L. Dugelay, A subspace co-training framework for multi-view clustering, Pattern Recognit. Lett. 41 (0) (2014) 73–82.
[27] Q. Yin, S. Wu, R. He, L. Wang, Multi-view clustering via pairwise sparse subspace representation, Neurocomputing 156 (0) (2015) 12–21.

Guojun Gan is an assistant professor in the Department of Mathematics at the University of Connecticut, where he has been since August 2014. He received a B.S. from Jilin University, China, in 2001 and M.S. and Ph.D. degrees from York University, Canada, in 2003 and 2007, respectively. His research is on variable annuity valuation and hedging, open source variable annuity valuation systems, and high dimensional data and large data clustering.

Michael Kwok-Po Ng is a Professor in the Department of Mathematics and Professor (Affiliate) of Department of Computer Science at the Hong Kong Baptist University. He obtained his B.Sc. degree in 1990 and M.Phil. degree in 1992 at the University of Hong Kong, and Ph.D. degree in 1995 at Chinese University of Hong Kong. He was a Research Fellow of Computer Sciences Laboratory at Australian National University (1995–1997), and an Assistant/Associate Professor (1997–2005) of the University of Hong Kong before joining Hong Kong Baptist University.