IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 29, 2019, accepted June 19, 2019, date of publication June 24, 2019, date of current version August 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924491

Sequential Minimax Search for Multi-Layer Gene Grouping

WENTING WANG[1,2], XINGXING ZHOU[3], FUZHONG CHEN[4], AND BEISHAO CAO[5]

[1]College of Computer Science and Software Engineering, Big Data Institute, Shenzhen University, Shenzhen 518060, China
[2]National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen 518060, China
[3]Guangdong Academy of Agricultural Sciences, Guangzhou 510640, China
[4]School of International Trade and Economics, University of International Business and Economics, Beijing 100029, China
[5]Department of mathematics, Sun Yat-sen University, Guangzhou 510275, China

Corresponding author: Xingxing Zhou (zhouxingxing0808@163.com)

This work was supported in part by the Guangzhou Science and Technology Plan Project ‘‘Guangdong Provincial Modern Agri-tech Transfer and Demonstration Center Construction’’ under Grant 201806040001, and in part by the Guangdong Academy of Agricultural Sciences Dean Fund Project ‘‘Research on the Integration of Key Elements in the Construction of Agri-tech Transformation Service Platform’’ under Grant 201829.

ABSTRACT Many areas of exploratory data analysis need to deal with high-dimensional data sets. Some real life data like human gene have an inherent structure of hierarchy, which embeds multi-layer feature groups. In this paper, we propose an algorithm to search for the number of feature groups in high-dimensional data by sequential minimax method and detect the hierarchical structure of high-dimensional data. Several proper numbers of feature grouping can be discovered. The feature grouping and group weights are investigated for each group number. After the comparison of feature groupings, the multi-layer structure of feature groups is detected. The latent feature group learning (LFGL) algorithm is proposed to evaluate the effectiveness of the number of feature groups and provide a method of subspace clustering. In the experiments on several gene data sets, the proposed algorithm outstands several representative algorithms.

INDEX TERMS Machine learning, evolutionary computing, feature grouping, high-dimensional data analysis, gene grouping, knowledge transfer.

I. INTRODUCTION

Analysis of high-dimensional data is a challenging problem in machine learning and artificial intelligence. Thousands of features in the objects cause a great complexity when using the classic tools to cluster and analyze the data [1]. High-dimensional data often contain many redundant, irrelevant and noise features, which affects the learning of the data. Gene data, as one typical kind of high-dimensional data, has drawn attention from different disciplines [2]. The research on how the gene influence different kinds of diseases is expected to start from a proper division of gene groups [3]–[5]. However, the underlying distribution and structure of the features is invisible, which causes dilemma for the further understanding of gene. In the area of machine learning and bioinformatics, researchers attempt to transfer the problem of feature grouping to the clustering of the objects. By observing the objects in the same clusters, the important feature groups they share are identified.

In the past decades, subspace clustering algorithms are discovered to be one of the most effective method to handle high-dimensional data like the gene data. They do not cluster data in the original space, instead, they map the data into subspace where the clustering is easier [6]. Among the various subspace clustering methods, soft subspace clustering is an important technique. It assigns weights to individual features and uses the weights to identify important features from which the subspace structures of clusters can be discovered [7], [8]. For instance, the feature grouping weighting k-means algorithm FG-k-means was proposed for high-dimensional data [9]. In this algorithm, the features are divided into a small set of feature groups, each being treated as a grouping feature in the low dimensional space of feature groups. The high-dimensional data is clustered on group features and the clusters in different subspaces of group features are discovered by assigning weights to group features [10]. Because the group features generalize the information of individual features in high-dimensional data, the FG-k-means algorithm often performs better than the clustering algorithms that cluster data on individual features.

The associate editor coordinating the review of this manuscript and approving it for publication was Quan Zou.

In the past decade, research in this area have been developed significantly [8]–[25].

However, the research in this area are mostly based on the prior knowledge of the number of feature groups and feature structures. Due to the high cost and technical difficulties to obtain the real information of feature structures, the number of features are usually set randomly in the clustering algorithms. The natural structure of high-dimensional data can easily get misunderstood and the clustering algorithm becomes inaccurate and instable.

In this paper, we propose a method to determine the number of feature groups and feature structures by the sequential minimax search. An unsupervised clustering algorithm named latent feature group learning (LFGL) is established to evaluate the feature grouping and provide clustering results for gene data. The algorithm learns the latent feature groups in the process of subspace clustering of high-dimensional data. Since the best number of feature groups is not unique for some data sets. We compare the feature groupings based on several different groups numbers and investigate the hierarchical structure in the gene data.

Experiments were conducted on disease data sets, which contain high-dimensional gene features. In the experiments on several gene data sets, the proposed algorithm outstands several representative algorithms.

The remainder of this paper is organized as follows. In Section 2, we review some related work. In Section 3, we present the details of sequential minimax search. In Section 4, we propose the algorithm to determine the best numbers of feature groups. Section 5 presents the latent feature group learning(LFGL) model for projection of high-dimensional data to a low-dimensional space. The experimental results on several real world data sets are discussed in Section 6. The conclusions are drawn in Section 7.

II. RELATED WORK

The technology of clustering has many applications in gene data expression [26]–[28] and gene sequence [29]. The research on both proteomics and metabolomics are developed [30], as well as on the context of protein comparison and structure prediction [31], [32]. However, the studies in this kind are usually rely on the class symbol of the clusters. For some data, it is difficult to obtain the symbols, where the unsupervised are employed.

In the past decade, soft subspace clustering has been an important research topic in cluster analysis [8]–[15], [17]–[21], [24], [25].

Huang *et al.* [19] proposed the W-k-means clustering algorithm which not only provide the clustering but the weights of the clusters as well. Some similar algorithms were proposed [8], [20]

Chen's algorithm [9], [10] proposed a two-level weighting method, which enhance the ability to select the important features and feature groups among the high-dimensional features. Later, Cai *et al.* [33] used the FG-k-means for text clustering. They first used the topic model LDA to partition the words into several groups and then used FG-k-means to cluster text data. The experimental results showed that the word grouping method has improved the clustering performance on text data.

However, the research in this area are mostly based on the prior knowledge of the number of feature groups and feature structures. The number of feature groups t in a cluster is a hyperparameter whose setting is affected by the subspace dimension of the cluster. For most of the studies, an estimation of the upper bound of t is

$$t \leq m/f_{max} \tag{1}$$

where f_{max} denotes the maximum number of determinant features of a cluster. The estimation in (1) is based on the assumption that the determinant features of a cluster can be collected as one of the t groups automatically by the algorithms listed. The natural structure of high-dimensional data can easily get misunderstood and the clustering algorithm becomes inaccurate and instable. In this paper, we will learn the number of feature groups and the clustering of data sets at the same time.

III. SEQUENTIAL MINIMAX SEARCH

High-dimensional data exist in the structure of groups, or even in the form of hierarchy. More than one proper value of t may exist. It is almost impossible to learn the proper group numbers by violence search. Hence, we intend to find a method to search for more than one feature grouping in a data set analytically. In this section, we introduce the sequential minimal search, which provide a method to search for the number of feature groups.

The method was built to search for an interval of the parameter, or a strategy to optimize a problem when the function is not given [34], [35]. Take the number of feature group t for instance, the unknown function to solve is the best way to group the high-dimensional features and calculate any possible machine learning process $f(t)$ afterwards. The cost of computation is huge if we consider it as a discrete problem and search for the solution of any t given the feature number N. Instead, we look for an interval D with length $L(D)$ where the proper value of t is contained.

First of all, we build several nonrandomized strategy sets in the form of $S = \{t_1, \varphi_2, \cdots, \varphi_n, s, t\}$. In the strategy, the number t_1 is any random value $t \in [1, N]$, functions $\varphi_2, \cdots, \varphi_n$ are the strategies to looking for the value of $t_2, \cdots, t_n$. For instance, $t_2 = g(t_1, \varphi_2, f(t_1))$. The starting point of the interval s and the end point t are determined eventually by the values of $t_1, \cdots, t_n$.

To solve an unknown problem, we will build a class of strategy set S_N to find an $S_N^* \in S_N$ such that

$$\sup_{f \in f} L(D(f, S_N^*)) \leq \inf_{s \in S_N} \sup_{f \in f} L(D(f, S_N)) + \epsilon \tag{2}$$

where we can find at least one proper searching strategy φ^* and consequently determine the best value for t.

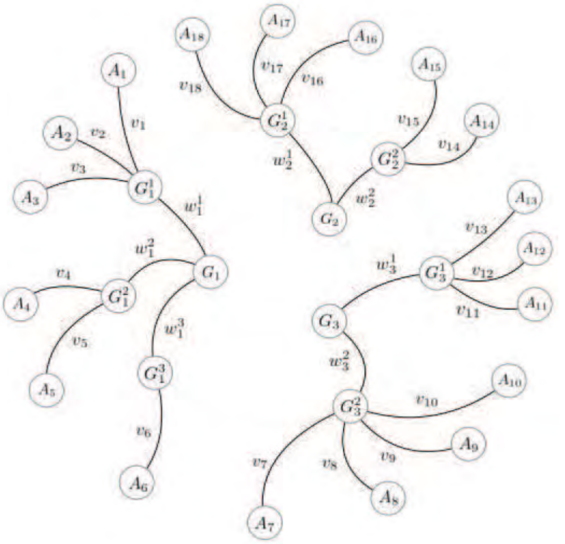


FIGURE 1. Feature mapping model of LFGL.

Usually, the n-th Fibonacci number U_n is used to build the sets of φ_n. A possible S_N^* may provide the strategy as follows: $x_1 = U_{N-1}/U_{N+1}$, $x_2 = lx_1 = U_N/U_{N+1}$. Since $L(D(f, S_N^*)) \leq \epsilon + l/U_{N+1}$, there would exist a procedure $\overline{S} \in S_{n+1}$ for which

$$\sup_{f \in f} L(D(f, \overline{S})) < 1/U_{n+2} \tag{3}$$

We may suppose that, under $\overline{S}$, φ can be proved as a constant [34], [35]. Here, an empirical value is used as follows: $t_2 = l - t_1 = -l/2 + 5^{1/2}/2 = 0.618 = \varphi$. Then, we do not need to determine the observation of t_1 randomly. Instead, we specify the value after several values are observed. In this way, the best strategy set S_N^* can be found, where more than one value of t are provided.

In the next several sections, we will use the method to find the values of t and build feature groups. We intend to build relations between the feature groups with different values of t and look for the hierarchy structure in the high-dimensional gene data.

IV. DETERMINE THE NUMBER OF FEATURE GROUPS

Although many features are used to describe data in high-dimensional spaces, only a few of them are needed to distinguish a specific cluster from others. Thus, the subspace clustering algorithms [8], [19], [20], [36] have obtained better performance than that of the general k-means algorithm. As the number of dimension increases, the strategy of searching the subspace of determinant features in the entire space of features often leads to suboptimal results, due to many noise features [10]. Meanwhile, it makes the performance of the subspace clustering deteriorated.

To solve the above problem, we map the high-dimensional features into low latent feature grouping space. Features are not independent in high-dimensional spaces. Rather, they gather together into nearly mutually exclusive groups. For some data sets, they exist in the hierarchy structure as illustrated in Fig.1.

However, the research in this area are mostly based on the prior knowledge of the number of feature groups, which may cause misunderstanding of the natural structure of the data. In this section, we search for the best number of groups t in high-dimensional data. In this paper, we propose an algorithm to determine the number of feature groups in high-dimensional data as in Algorithm 1.

Algorithm 1 Determine the Number of Feature Groups

```
Input: The data set $X \in \mathbb{R}^{n \times m}$, where $n$ is the number of objects and $m$ is the number of features; The set of unlabeled samples for current batch, $U_n$; The searching parameter $\varphi$;
Output: The number of feature groups, $t$;
1: Given $m$, build a set $T = \{t_1, t_2, \cdots, t_l\}$, where $t_1 = m$, $t_2 = [\varphi t_1] \cdots$ till $t_l$ becomes a single digit (the symbol [ ] means trunc);
2: For each value $t_i$ in $T$, build 20 chromosomes (in section 4.1) using the Darwinian method;
3: Evaluate all of the chromosomes by Davies Bouldin Index calculated from the latent feature grouping clustering algorithm (in section 4.1 and 4.2);
4: Keep the best half and build new generations by crossover and mutation between chromosomes with same number of feature groups $t_i$. If the number of chromosomes are not enough for the process, no new generation will be made;
5: Repeat step 3-4 for 10 times and find the $t_i$ with the highest accuracy or rand index in the last round, $t = t_i$;
6: Record the number of chromosomes from different $t_i$ in every loop;
7: return $t$;
```

V. LATENT FEATURE GROUP LEARNING(LFGL)

In this section, we propose an unsupervised clustering algorithm named latent feature group learning(LFGL). The algorithm is embedded in the process to determine the number of feature groups and provide a proper result of clustering eventually.

A. FEATURE GROUP WEIGHTING

In this part of study, we build the latent feature grouping model as in Fig.2. We define a mapping g from the data space $\mathbb{R}^m$ to the latent space $\mathbb{R}^t$. The first layer in Fig.1 shows the features in a data set $X_{i,j} \in \mathbb{R}^{n \times m}$. This set of features $\mathcal{A} = \{x_1, x_2, \cdots, x_m\}$ are mapped into t groups $\{g_1, g_2, \cdots, g_t\}$ in the middle layer. The weights are kept in the feature-group matrix $V \in \mathbb{R}_+^{t \times m}$. Then, the feature groups are weighted with $W \in \mathbb{R}_+^t$ to result in the weighted group values $g(x)_1, g(x)_2, \cdots, g(x)_t$. The process is based on the observation that not all groups are useful to identify a specific

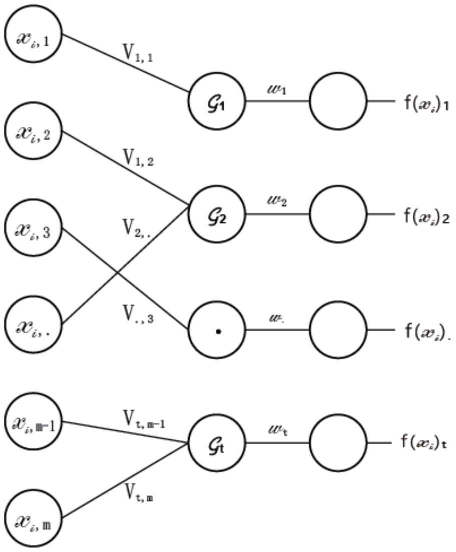


FIGURE 2. Feature mapping model of LFGL.

cluster and different clusters should be identified by different sets of groups.

Therefore, we delegate the task to the vector-value function g, which can be defined as needed for different sets of concerns. We set $V = V_0 \circ V_l$ to separate the partition and weights of feature groups into two matrices. Here, we define the linear mapping g as

$$g(x) = W_l(V_0 \circ V_l)x^T. \tag{4}$$

The none zero elements in V_l are the same as those in V_0. V_ls are initialized and optimized on the constraint of $V_l V_l^T = I$ so we have

$$\sum_{j=1}^{6} v_{i,j}^2 = 1, \quad for\ 1 \leq i \leq 3 \tag{5}$$

Since $V_l = V_0 \circ V_l$, the feature group structure V_0 and the individual feature weights V_l in each cluster can be optimized separately. In the current feature grouping weighting algorithms such as FG-k-means, V_0 is supposed to be known in advance and V_ls are optimized in the clustering process. However, V_0 is not known in many real world data sets. Therefore, the current algorithms are not able to learn the feature grouping structure automatically. Here, we propose the first method to learn the feature grouping structure automatically in the clustering process.

B. REVISED FG-K-MEANS

The objective function of the revised FG-k-means [9] is defined as follows:

$$P(U, Z, V, W) = \sum_{l=1}^{k}[\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{j\in G_t} h_{i,l} w_{l,t} v_{l,j} d(x_{i,j}, z_{l,j}) + \lambda \sum_{t=1}^{T} w_{l,t}\log(w_{l,t}) \tag{6}$$

subject to

$$\begin{cases} \sum_{l=1}^{k} h_{i,l} = 1, & \text{i} = 1, \ldots, \text{n} \\ \sum w_l = 1, & \text{l} = 1, \ldots, \text{k} \\ v_l v_l^T = I, & \text{l} = 1, \ldots, \text{k} \end{cases}$$

where $X = \{x_i \mid x_i \in \mathbb{R}^m\}_{i=1}^n$ is the set of n objects each with m features, $H = \{h_{i,l} \mid h_{i,l} \in \{0, 1\}\}_{i=1,l=1}^{n,k}$ is the set of indicators of memberships of k objects in clusters, and $Z = \{z_l \mid z_l \in \mathbb{R}^m\}_{l=1}^k$ is the set of k cluster centers.

We revise the original FG-k-means by setting the orthogonal constraint on the matrices of the individual feature weights V_l.

We minimize the cost function Eq.(10) by iteratively solving the following four minimization problems. In each problem, we fix three parameters among Z, V, W, U, and optimize the last one eventually.

The optimization of U is solved by:

$$\begin{cases} u_{i,l} = 1, & \text{if } D_l \leq D_s \text{ for } 1 \leq s \leq k; \\ u_{i,s} = 0, & \text{otherwise.} \end{cases}$$

where $D_s = \sum_{t=1}^{T} w_{s,t} \sum_{j\in G_t} v_{s,j} d(x_{i,j}, z_{s,j})$.

The optimization of Z is solved by updating the centers of the clusters by Algorithm 1.

Algorithm 2 Cluster Center Updating

```
Input: $U, W, V$;
Output: The updating cluster centers $Z$;
Generate a similarity matrix $S \in \mathbb{R}^{n*n}$, where $s_{i,j} = s_{j,i} = u_{i,j}d(x_i, x_j)$;
Generate a distance sum matrix $S_{sum} \in \mathbb{R}^{1*n}$, where $s_i = \sum_{j=1}^{n} s_{j,i}$;
Generate a density matrix $D \in \mathbb{R}^{1*n}$, find the smallest value in every column $m$ of $D * U$ and consider the corresponding object as the center of the cluster $m$ for $m = 1, 2, \cdots, k$.
return $Z$;
```

The optimization of W is solved by **Theorem 1**:

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

$$w_{l,t} = \exp\frac{-E_{l,j}}{\lambda} / \sum_{j\in G_t} \exp\frac{-E_{l,j}}{\lambda} \tag{7}$$

where

$$D_{l,t} = \sum_{i=1}^{n} \widehat{u_{i,l}}\widehat{v_{l,j}} d(x_{i,j}, \widehat{z_{l,j}}) \tag{8}$$

Proof: Given $U = \hat{U}$, $Z = \hat{Z}$ and $V = \hat{V}$, we minimize the objective function with respect to W. Since there exist a set of $k \times T$ constraints $\sum_{t=1}^{T} w_{l,t} = 1$, we form the Lagrangian by isolating the terms which contain

$\{w_{l,1}, w_{l,2}, \cdots, w_{l,t}\}$ and adding the appropriate Lagrangian multipliers as

$$\begin{aligned} L_{\{w_{l,1},w_{l,2},\cdots,w_{l,T}\}} = & \sum_{t=1}^{T}[w_{l,t}D_{l,t} \\ & + \lambda \sum_{t=1}^{T}\sum_{j\in G_t} w_{l,t}\log w_{l,t} v_{l,j}\log v_{l,j} \\ & + \gamma_{l,t}(\sum_{j\in G_t} w_{l,t} - 1)] \end{aligned} \quad (9)$$

where $D_{l,t}$ is a constant in the t-th feature group on the l-th cluster for fixed $U = \hat{U}, Z = \hat{Z}$ and $V = \hat{V}$, and calculated before.

By setting the gradient of $L_{\{w_{l,1},w_{l,2},\cdots,w_{l,T}\}}$ with respect to γ and $w_{l,t}$ to zero, we obtain

$$\frac{\partial L_{\{w_{l,1},w_{l,2},\cdots,w_{l,T}\}}}{\partial \gamma} = \sum_{t=1}^{T} w_{l,t} - 1 = 0 \quad (10)$$

and

$$\begin{aligned} \frac{\partial L_{\{w_{l,1},w_{l,2},\cdots,w_{l,T}\}}}{\partial w_{l,t}} = & D_{l,t} \\ & + \lambda \sum_{j\in G_t} v_{l,j}\log v_{l,j}(1 + \log w_{l,t}) + \gamma = 0 \end{aligned} \quad (11)$$

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

Then, we obtain

$$w_{l,t} = \exp\frac{-D_{l,t}}{\lambda} / \sum_{t=1}^{T} \exp\frac{-D_{l,t}}{\lambda} \quad (12)$$

The optimization of V is solved as follows.

Because of the additivity of the objective function (10), the matrix W can be divided into k subproblems for k clusters, respectively. Let

$$\begin{aligned} Q_l &= \text{diag}(w_l)^T\text{diag}(w_l) \quad \text{and} \\ q_{i,l} &= h_{i,l}(x_i - z_l), \end{aligned} \quad (13)$$

the lth subproblem of the original problem can be written as

$$\begin{aligned} & \min_{V\in\mathbb{R}_+^{t\times m}} \sum_{i=1}^{n} q_{i,l}^T V_l^T Q_l V_l q_{i,l} \\ & s.t. \\ & subjectto\ V_l V_l^T = I. \end{aligned} \quad (14)$$

The subproblem (21) has nonnegative and orthogonal constraints simultaneously on the matrix V_l, which makes the problem NP hard to solve directly. The methods used here are analogous to that of non-negative matrix factorization (NMF).

We replace the orthogonal constraint with a F-norm measurement of orthogonality as the relaxation, that is

$$\min_{V\in\mathbb{R}_+^{t\times m}} \sum_{i=1}^{n} q_{i,l}^T V^T Q_l V q_{i,l} + \frac{\eta}{2}(VV^T - I)_F^2, \quad (15)$$

where $\eta \geq 0$ is a parameter to control the orthogonality of V explicitly. The Lagrangian of (22) is

$$L(V, \Lambda) = \sum_{i=1}^{n} q_{i,l}^T V^T Q_l V q_{i,l} + \frac{\eta}{2}(VV^T - I)_F^2 - tr(\Lambda V^T), \quad (16)$$

where Λ is the Lagrange multiplier for the constraint $V \in \mathbb{R}_+^{t\times m}$. By $\nabla_V L = 0$, we have

$$2QVSS^T + 2\eta(VV^T - I)V - \Lambda = 0, \quad (17)$$

where $S = [q_{1,l}, q_{2,l}, \ldots, q_{n,l}] \in \mathbb{R}^{m\times n}$. According to the KKT complementary condition on $[V]_{i,j} \geq 0$, by making a Hadamard product with V on both sides of Eq.(24), we obtain

$$(QVSS^T + \eta(VV^T - I)V) \circ V = 0. \quad (18)$$

The multiplicative updating rule for V_l is derived as

$$V_{i,j} \leftarrow V_{i,j}\frac{[QV(SS^T)^- + \eta V]_{i,j}}{[QV(SS^T)^+ + \eta VV^TV]_{i,j}}, \quad (19)$$

where η is a parameter to control the orthogonality among different rows of V, $()^+$ and $()^-$ are the operators to get the positive and negative parts of the input matrix, respectively, i.e.,

$$[(A)^+]_{i,j} = \begin{cases} [A]_{i,j} & \text{if} \quad [A]_{i,j} > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (20)$$

$$[(A)^-]_{i,j} = \begin{cases} |[A]_{i,j}| & \text{if} \quad [A]_{i,j} < 0 \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

C. EVOLUTIONARY METHOD TO SELECT THE BEST FEATURE GROUPING STRUCTURE V_0

The Darwinian evolutionary process [37] is used to search for the best feature grouping structure V_0. In this process, the feature grouping structures V_0 are encoded as chromosomes and the revised FG-k-means is used as the fitness function to evaluate the chromosomes. The best V_0 is selected through evolutions of generations. To our knowledge, this is the first attempt to use the evolutionary process to search for the best feature grouping structure from high-dimensional data.

We first present a classical evolutionary method where the population looks for the best grouping from the set of features. Each individual chromosome encodes a feature grouping structure V_0. The chromosome $A^{i,g}$ of the ith individual in the gth generation is defined as

$$A^{i,g} = (V_1^{i,g}, \cdots, V_k^{i,g}, \cdots, V_m^{i,g}) \quad (22)$$

where $A^{i,g}$ is a binary sequence of length t, $V_k^{i,g}$ is the kth column of the matrix V_0 and m is the number of features in the data set. For example, a partition of 6 features into 3 groups is specified in the partition matrix V_0 as follows.

$$V_0 = \begin{array}{c} \\ \mathcal{G}_1 \\ \mathcal{G}_2 \\ \mathcal{G}_3 \end{array} \begin{array}{c} \begin{array}{cccccc} A_1 & A_2 & A_3 & A_4 & A_5 & A_6 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array} \quad (23)$$

where the rows of V_0 represent the feature groups and the element 1 in each column indicates the feature group the feature is assigned to. The matrix V_0 is equivalent to the chromosome $A^{i,g}$ as follows:

$$A^{i,g} = \{(1, 0, 0),\ (1, 0, 0),\ (0, 0, 1),\ (0, 1, 0),\ (0, 0, 1),\ (0, 1, 0)\} \tag{24}$$

We can see that each binary sequence $V_k^{i,g}$ has only one element as 1 and the rest as 0. This is a constraint on the structure of chromosomes. To start the evolutionary process, 20 chromosomes are generated randomly as the first generation of individuals. To generate the binary sequences for each chromosome, one position is randomly selected from t possible positions, and is set as value 1. The rest t-1 positions are set as 0.

After all chromosomes are initialized, they are evaluated by the revised FG-k-means algorithm with the input data set. From each chromosome, the matrix V_0 is constructed. Matrix V_l is initialized by solving $V_l V_l^T = I$. Since the solutions are not unique, different initial V_ls for different clusters are initialized. Then, the initial V_ls are obtained by $V_l = V_0 \circ V_l$.

The initial feature group weights and initial cluster centers are generated and selected randomly. The number of clusters k is given. The revised FG-k-means algorithm is executed on the input data set once for each chromosome to produce one clustering result. The DBI (Davies Bouldin Index) is used to evaluate the clustering result and score the chromosome.

After all chromosomes are scored, the genetic operations like selection, crossover and mutation are applied to the chromosomes to produce new individual chromosomes for the next generation as follows:

10 strongest chromosomes are selected according to the scores. The crossover is performed in the following steps: randomly divide the 10 chromosomes into 5 pairs. For each pair of chromosome i and chromosome j, the corresponding binary sequence $V_k^{i,g}$ and $V_k^{j,g}$ are compared. If two sequences are same, the sequence is copied as the new generation of $V_k^{s,g+1}$. For the remaining pairs of different binary sequences, we randomly select one sequence from one chromosome to replace the corresponding sequence of another chromosome by the probability $\alpha_k \in [0, 1]$. Finally, we encode V_0 as a new chromosome in the next generation. The rule to generate $V_k^{s,g+1}$ is defined as follows

$$V_k^{s,g+1} = \begin{cases} V_k^{i,g} & \text{if} \quad V_k^{i,g} = V_k^{j,g} \quad or \quad \alpha_k \geq 0.5 \\ V_k^{j,g} & \text{otherwise.} \end{cases} \tag{25}$$

where α_k is randomly generated for each $V_k^{s,g+1}$.

For the process of mutation, we randomly choose 5 chromosomes from the selected 10 chromosomes. For each chromosome $A^{i,g}$, we generate a random new chromosome $A_k^{rand} = (V_1^{rand}, \cdots, V_k^{rand}, \cdots, V_m^{rand})$. The rule to generate $V_k^{i,g+1}$ is

$$V_k^{i,g+1} = \begin{cases} V_k^{i,g} & \text{if} \quad \alpha_k \geq 0.5 \\ V_k^{rand} & \text{otherwise.} \end{cases} \tag{26}$$

where α_k is randomly generated for each $V_k^{i,g+1}$.

In this way, we generate 10 new chromosomes and combine them with the 10 strongest chromosomes to form a new population for exploration and exploitation in the next generation of evolution.

D. LFGL ALGORITHM

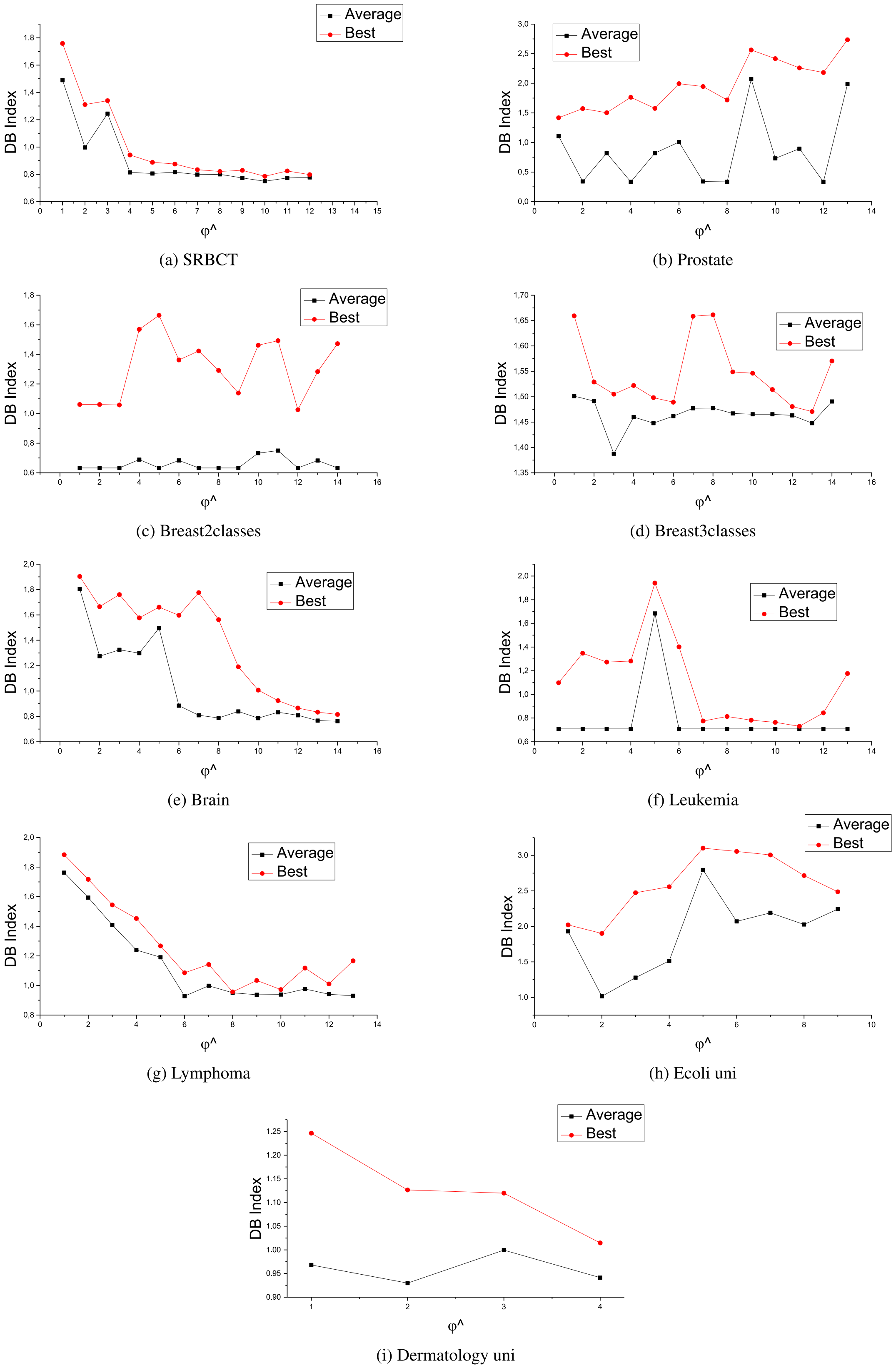
The process of learning the latent feature grouping structure V_0, the individual feature weights, the feature group weights and a chromosome score from the input data set consist of three stages. The initialization stage generates the first generation of 20 chromosomes representing 20 initial V_0s. The second stage uses the revised FG-k-means algorithm to score the 20 chromosomes. The third stage selects the 10 strongest chromosomes according to the scores and perform the genetic operations on the selected chromosomes to produce the new generation of chromosomes for evolution. This process continues until the termination criterion is met. The LFGL algorithm implements the evolution process in Algorithm 2.

Algorithm 3 LFGL

```
Input: The dataset $X$, the number of clusters $k$, two positive
    parameters $\lambda$, $\eta$, the number of feature groups $t$;
Output: Local optimal values of $\mathcal{H}$, $\mathcal{Z}$, $\mathcal{V}$, $\mathcal{W}$;
1: Initialize 20 chromosomes representing 20 different possibilities of feature grouping;
2: For each chromosome, we initialize $\mathcal{W}$ by sampling positive values $[w_l]_i \sim \mathcal{N}(1, 0.01)$, then normalize $w_l$ so that $1^T w_l = 1$;
3: Initialize $\mathcal{V}$ by the algorithm in Equation (27) and (28) to build the $V$ matrix, then normalize $V_l$ so that the $\ell^2$-norm of each row $V_l$ of is 1;
4: Randomly choose $k$ cluster centers $Z^0$;
5: Update $H^{t+1}$, $Z^{t+1}$, $W^{t+1}$ and $V^{t+1}$ respectively;
6: The objective function $P$ obtains its local minimum value, then update $V^{t+1}$ and go back to the step 9;
7: Calculate the BIC of the 20 clustering results from 20 chromosomes, choose the best 10 ones and make 10 new chromosomes by crossover and mutation in Section 4.4.
8: Repeat until ten times and find the best solution of clustering.
```

VI. EXPERIMENT AND ANALYSIS

Nine real world high-dimensional data sets were used in the experiments to evaluate the algorithm 2. Seven genetic data sets were downloaded from *http : //archive.ics.uci.edu/ml/ datasets.html*, and the other two data sets were obtained from *http : //www.escience.cn/people/fpnie/papers.html*.



(a) SRBCT

(b) Prostate

(c) Breast2classes

(d) Breast3classes

(e) Brain

(f) Leukemia

(g) Lymphoma

(h) Ecoli uni

(i) Dermatology uni

FIGURE 3. **The Davies Bouldin Index values based on different numbers of gene groups in 9 data sets.**

TABLE 1. Dataset description.

Data	Abbr	Objects	Features	Classes
SRBCT	SRB	63	2308	4
Lymphoma	lym	62	4026	3
Prostate	Pro	102	6033	2
Leukemia	Leu	38	3052	2
Breast2classes	Br2	77	4870	2
Brain	Bra	42	5598	5
Breast3classes	Br3	94	4870	3
Ecoli uni	Eco	42	5598	5
Dermatology uni	Der	94	4870	3

The common characteristics of these data sets are small numbers of objects with large numbers of features. The details of the data sets are listed in Table.1.

A. THE EVALUATION OF GROUP NUMBERS

We use two measures to evaluate the number of feature groups: the Davies Bouldin Index and the Dunn Index, both of which calculate the effectiveness of clustering without the participation of real classes of the data.

Davies Bouldin Index is defined as:

$$S_i = \{\frac{1}{T_i}\sum_{j=1}^{T_i} |X_j - A_i|^q\}^{1/q} \tag{27}$$

where X_j is the jth member in ith cluster and A_i is the center. T_i is the number of members in the ith cluster. We use $q = 1$ in this paper to indicate the mean value of the distances from all members to their center in one cluster. The DBI is expected to be high to represent the compactness of one cluster.

Dunn Index is defined as:

$$DI_m = \frac{\min_{1\leq i<j\leq m} \delta(C_i, C_j)}{\max_{1\leq k\leq m} \Delta_k} \tag{28}$$

where $\delta(C_i, C_j)$ is this inter-cluster distance metric, between clusters C_i and C_j.

The average value and the best value of the Davies Bouldin Index are presented in Fig.3. We also record the Davies Bouldin Index and Dunn Index in Table.2. For most of the data sets, we observe fluctuations of the two values during the decrease of the group number t. The high values indicate the good partition of the data set. It is worth investigating the relations between the groupings of the best several choices of t.

B. HIERARCHICAL STRUCTURE OF HIGH-DIMENSIONAL DATA

In this section, we compare the groupings of different group number t and discover the overlap between the good groupings. In table 3, we record the rates of overlaps for the 9 data sets, which may provide hierarchical structures in the following investigation.

C. CLUSTERING

We have tested the latent feature grouping learning (LFGL) algorithm on several data sets in high dimensions. The results are compared with other five popular clustering algorithms. In this section, we present the experiments and demonstrate the results of the LFGL algorithm in comparison with those of five existing algorithms: Kmeans, TWKM, EWKM, LAC and the original FG-k-means.

TABLE 2. The DBI and DI of the data set.

Data	Average DBI	Best DBI	Average DI	Best DI
SRB	.863	1.770	.985	1.114
Lym	1.112	1.847	.959	1.105
Pro	.510	1.616	1.444	1.661
Leu	.982	1.732	1.051	1.211
Br2	.632	1.736	1.175	1.250
Bra	0.822	1.808	0.92	1.073
Br3	.661	1.381	1.02	1.167
Eco	2.071	3.102	.951	.969
Der	.941	1.247	.853	1.031

TABLE 3. The rate of overlaps between the feature structures.

Data	Highest rate	Weighted highest rate	t_1	t_2
SRB	.328	.403	3	11
Lym	.355	.476	7	13
Pro	.330	.506	9	13
Leu	.315	.555	3	7
Br2	.542	.572	5	14
Bra	.378	.603	5	13
Br3	.413	.603	7	8
Eco	.357	.503	5	8
Der	.324	.603	1	3

Each algorithm was run on each data set for 100 times to produce 100 results. The average value of the measures of the 100 results on each data set by an algorithm is used as the measure of the algorithm. We use Rand Index and Accuracy to measure the clustering algorithms.

The number of clusters k was given as the number of classes in the data sets for all algorithms. For the LFGL algorithm, the two parameters λ and η, included in the objective function Eq.(6) and the subproblem (22), were set as $\lambda = 1$ and $\eta = 1$ for all data sets. Since the LFGL algorithm is used to measure the effectiveness of the feature grouping, we will not investigate in the optimization of the two parameters in this paper.

The clustering results of the 9 Genetic data are shown in Table 4. From the results, we can see that LFGL outperformed all other five clustering algorithms on most data sets. If we consider all clustering results, LFGL significantly outperformed all other five clustering algorithms on almost all data sets. On other data sets, LFGL produced similar results as the other five clustering algorithms. If we consider the 10 best clustering results by the five measures, we can see that LFGL significantly outperformed all five algorithms on almost all data sets. These results show that LFGL is effective in clustering high-dimensional data. The algorithm LFGL is established particularly with a target on the Genetic

TABLE 4. Summary of clustering results by six clustering algorithms.

Data	Evaluation	Kmeans	EWKM	TWKM	LAC	FG-kmeans	LFGL
SRB	Rand Index	.661	.603	.654	.597	.654	.777
	Accuracy	.528	.476	.530	.413	.555	**.781**
Lymp	Rand Index	.727	.804	.612	.488	.921	.929
	Accuracy	855	.838	.755	.419	.951	**.951**
Pro	Rand Index	.507	.504	.507	.495	.509	.524
	Accuracy	.578	.606	.576	.500	.578	.617
Leu	Rand Index	.728	.830	.805	.788	.752	.999
	Accuracy	.842	.842	.832	.815	.742	.999
Br2	Rand Index	.546	.507	.479	.491	.508	.555
	Accuracy	.662	.584	.574	.470	.584	.611
Bra	Rand Index	.493	.506	.496	.500	.512	.841
	Accuracy	.661	.661	.661	.615	.555	.738
Br3	Rand Index	.507	.507	.496	.488	.501	.643
	Accuracy	.578	.563	.516	.528	.545	.631
Eco	Rand Index	.493	.506	.496	.500	.512	.841
	Accuracy	.661	.661	.661	.615	.555	.738
Der	Rand Index	.507	.507	.496	.488	.501	.643
	Accuracy	.578	.563	.516	.528	.545	.631

data sets to investigate the relations between human genes and diseases.

VII. CONCLUSIONS

In this paper, we propose an algorithm to search for the number of feature groups in human gene by sequential minimax method. Afterwards, the feature grouping and group weights are investigated from the high-dimensional gene and text data. The latent feature group learning (LFGL) algorithm is proposed to evaluate the effectiveness of the number of feature groups and provide a method of subspace clustering. When the several proper feature groupings are determined, we compare the groupings and record the overlaps between them. Therefore, the multi-layer groupings are discovered, which form a hierarchy structure of the gene data. The weights of the features and feature groups in every layer are calculated too. Meanwhile, the clustering results provided by LFGL outstands some representative algorithms. The future work will focus on how to use the hierarchy structure to investigate the relations between gene data and diseases.

REFERENCES

[1] D. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," Amer. Math. Soc.-Math. Challenges 21st Century, Los Angeles, VA, USA, Tech. Rep. 4, Aug. 2000.

[2] K. J. J. Handl and D. B. Kell, *Computational Cluster Validation in Post-Genomic Data Analysis*. London, U.K.: Oxford Univ. Press, 2005.

[3] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 2002.

[4] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 1, no. 1, pp. 24–45, Jan./Mar. 2004.

[5] S. Brodowski, "A validity criterion for fuzzy clustering," in *Proc. Int. Conf. Comput. Collective Intell., Technol. Appl.*, 2011, pp. 132–151.

[6] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: A review," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 90–105, Jun. 2004.

[7] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 1, pp. 1–58, 2009.

[8] L. Jing, M. K. Ng, and Z. Huang, "An entropy weighting *k*-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 8, pp. 1026–1041, Aug. 2007.

[9] X. Chen, Y. Ye, X. Xu, and J. Z. Huang, "A feature group weighting method for subspace clustering of high-dimensional data," *Pattern Recognit.*, vol. 45, no. 1, pp. 434–446, 2012.

[10] X. Chen, X. Xu, Y. Ye, and J. Z. Huang, "TW-k-means: Automated two-level variable weighting clustering algorithm for multiview data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 932–944, Apr. 2013.

[11] R. Gnanadesikan, J. Kettenring, and S. Tsao, "Weighting and selection of variables for cluster analysis," *J. Classification*, vol. 12, no. 1, pp. 113–136, 1995.

[12] G. De Soete, "Optimal variable weighting for ultrametric and additive tree clustering," *Qual. Quantity*, vol. 20, nos. 2–3, pp. 169–180, 1986.

[13] G. Soete, "OVWTRE: A program for optimal variable weighting for ultrametric and additive tree fitting," *J. Classification*, vol. 5, no. 1, pp. 101–104, 1988.

[14] E. Fowlkes, R. Gnanadesikan, and J. Kettenring, "Variable selection in clustering," *J. Classification*, vol. 5, no. 2, pp. 205–228, 1988.

[15] V. Makarenkov and B. Leclerc, "An algorithm for the fitting of a tree metric according to a weighted least-squares criterion," *J. Classification*, vol. 16, no. 1, pp. 3–26, 1999.

[16] V. Makarenkov and P. Legendre, "Optimal variable weighting for ultrametric and additive trees and k-means partitioning: Methods and software," *J. Classification*, vol. 18, no. 2, pp. 245–271, 2001.

[17] D. Modha and W. Spangler, "Feature weighting in *k*-means clustering," *Mach. Learn.*, vol. 52, no. 3, pp. 217–237, 2003.

[18] J. H. Friedman and J. J. Meulman, "Clustering objects on subsets of attributes," *J. Roy. Statist. Soc. B*, vol. 66, no. 4, pp. 815–849, Nov. 2004.

[19] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 657–668, May 2005.

[20] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos, "Locally adaptive metrics for clustering high dimensional data," *Data Mining Knowl. Discovery*, vol. 14, no. 1, pp. 63–97, Feb. 2007.

[21] P. Hoff, "Model-based subspace clustering," *Bayesian Anal.*, vol. 1, no. 2, pp. 321–344, 2006.

[22] C. Bouveyron, S. Girard, and C. Schmid, "High dimensional data clustering," *Comput. Statist. Data Anal.*, vol. 52, no. 1, pp. 502–519, 2007.

[23] C.-Y. Tsai and C.-C. Chiu, "Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm," *Comput. Statist. Data Anal.*, vol. 52, no. 10, pp. 4658–4672, 2008.

[24] Z. Deng, K.-S. Choi, F.-L. Chung, and S. Wang, "Enhanced soft subspace clustering integrating within-cluster and between-cluster information," *Pattern Recognit.*, vol. 43, no. 3, pp. 767–781, 2010.

[25] H. Cheng, K. A. Hua, and K. Vu, "Constrained locally weighted clustering," *Proc. VLDB Endow.*, vol. 1, pp. 90–101, Aug. 2008.

[26] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Nat. Acad. Sci. USA*, vol. 95, no. 25, pp. 14863–14868, 1999.

[27] K. Iwao, R. Matoba, N. Ueno, A. Ando, Y. Miyoshi, K. Matsubara, S. Noguchi, and K. Kato, "Molecular classification of primary breast tumors possessing distinct prognostic properties," *Hum. Mol. Genet.*, vol. 11, no. 2, pp. 199–206, 2002.

[28] P. S. Vasisht, "Computational analysis of microarray data," *Nature Rev. Genet.*, vol. 2, pp. 418–427, Jun. 2003.

[29] Y. Bilu and M. Linial, "The advantage of functional prediction based on clustering of yeast genes and its correlation with non-sequence based classifications," *J. Comput. Biol.*, vol. 9, no. 2, pp. 193–210, 2002.

[30] R. Goodacre, É. M. Timmins, R. Burton, N. Kaderbhai, A. M. Woodward, D. B. Kell, and P. J. Rooney, "Rapid identification of urinary tract infection bacteria using hyperspectral whole-organism fingerprinting and artificial neural networks," *Microbiology*, vol. 144, no. 5, pp. 1157–1170, 1998.

[31] N. Kaplan, M. Friedlich, M. Fromer, and M. Linial, "A functional hierarchical organization of the protein sequence space," *BMC Bioinformat.*, vol. 5, p. 196, Dec. 2004.

[32] N. Krasnogor and D. A. Pelta, "Measuring the similarity of protein structures by means of the universal similarity metric," *Bioinformatics*, vol. 20, no. 7, pp. 1015–1021, 2004.

[33] Y. Cai, X. Chen, P. Peng, and J. Huang, "A lda feature grouping method for subspace clustering of text data," in *Intelligence and Security Informatics* (Lecture Notes in Computer Science), vol. 8440, M. Chau, H. Chen, G. Wang, and J.-H. Wang, Eds. Tainan, Taiwan: Springer, 2014, pp. 78–90.

[34] J. Kiefer, "Sequential minimax search for a maximum," *Proc. Amer. Math. Soc.*, vol. 4, no. 3, pp. 502–506, 1953.

[35] D. G. Chapman and H. Robbins, "Minimum variance estimation without regularity assumptions," *Ann. Math. Statist.*, vol. 22, no. 4, pp. 581–586, 1951.

[36] H. Xia, J. Zhuang, and D. Yu, "Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data," *Pattern Recognit.*, vol. 46, no. 9, pp. 2562–2575, 2013.

[37] P. Gancarski and A. Blansche, "Darwinian, lamarckian, and baldwinian (co)evolutionary approaches for feature weighting in *K*-means-based algorithms," *IEEE Trans. Evol. Comput.*, vol. 12, no. 5, pp. 617–629, Oct. 2008.

WENTING WANG was born in Nanjing, China, in 1982. She received the B.S. degree in engineering from Shandong University, in 2004, the M.S. degree in engineering from the Nanjing University of Aeronautics and Astronautics, in 2009, and the Ph.D. degree in mathematics from University College London, in 2017.

She is currently a Senior Researcher with the Big Data Institute, Shenzhen University, China. Her research interests include the optimization methods in machine learning, the Bayesian variational inference, and the high-dimensional data learning.

XINGXING ZHOU was born in Hunan, China, in 1987. He received the B.S. degree in engineering from Hunan University, in 2010, and the Ph.D. degree in engineering from the University of Technology Sydney (UTS), in 2014.

He is currently an Associate Professor with the Guangdong Academy of Agricultural Sciences, China. His research interests include technology transfer, incubate, and technology innovation.

FUZHONG CHEN was born in Sichuan, China, in 1983. He received the B.S. degree in marketing and the M.S. degree in industrial economics from the Nanjing University of Aeronautics and Astronautics, in 2007 and 2010, respectively, and the Ph.D. degree in industrial economics from the Renmin University of China, in 2014.

He is currently an Associate Professor with the School of International Trade and Economics, University of International Business and Economics, China. His research interests include economic optimization methodology, the application of topology in economic field, and foreign trade and international industry.

BEISHAO CAO is currently pursuing the bachelor's degree with the Department of mathematics, Sun Yat-sen University, Guangzhou, China.

He is a Trainee with the Shenzhen big data technology and Application Research Institute. His research interests include machine learning and subspace clustering.

• • •