Subspace Weighting Co-Clustering of Gene Expression Data

Xiaojun Chen, Joshua Z. Huang, Qingyao Wu, and Min Yang

Abstract—Microarray technology enables the collection of vast amounts of gene expression data from biological experiments. Clustering algorithms have been successfully applied to exploring the gene expression data. Since a set of genes may be only correlated to a subset of samples, it is useful to use co-clustering to recover co-clusters in the gene expression data. In this paper, we propose a novel algorithm, called *Subspace Weighting Co-Clustering* (SWCC), for high dimensional gene expression data. In SWCC, a gene subspace weight matrix is introduced to identify the contribution of gene objects in distinguishing different sample clusters. We design a new co-clustering objective function to recover the co-clusters in the gene expression data, in which the subspace weight matrix is introduced. An iterative algorithm is developed to solve the objective function, in which the subspace weight matrix is automatically computed during the iterative co-clustering process. Our empirical study shows encouraging results of the proposed algorithm in comparison with six state-of-the-art clustering algorithms on ten gene expression data sets. We also propose to use SWCC for gene clustering and selection. The experimental results show that the selected genes can improve the classification performance of Random Forests.

Index Terms—Gene expression data, co-clustering, subspace clustering, gene selection

1 Introduction

Microarrays simultaneously measure the expression levels of thousands of genes in experiments. A single microarray chip is able to generate expression levels from thousands of genes, and the data is usually collected from multiple tissue samples, in multiple patients, with different environmental conditions. The need to analyze these high dimensional data is driving the development of automatically analysis methods. One of the most widely used approaches for exploratory analysis of gene expression data is clustering which seeks to partition objects into clusters to maximize within-cluster similarity or minimize between cluster similarity [1], [2]. Clustering is also frequently used as the basis of further computational analysis. For example, the functional properties of a gene can be predicted according to known functions of related genes from the same cluster.

In the past decades, many clustering methods have been proposed for clustering analysis of high-dimensional data, including subspace clustering [3], co-clustering [3] and spectral clustering [4]. Among them, co-clustering is a process of simultaneously clustering the rows of a data matrix into row clusters and the columns of the data matrix into column clusters [5]. It has good performance on sparse and high-dimensional data, even if only considering the clustering task along one-way of the data [6]. Co-clustering has received wide attention in various applications such as simultaneous clustering of documents and words in text mining [7], genes and experimental conditions in bioinformatics [8], [9], [10], users and movies in recommendation systems [11]. For gene expression data, the use of co-clustering not only allows us to capture the correlated genes behave similar in all samples, but also enables the identification of genes that only correlated in a subset of samples which might fail to recover by using a conventional clustering algorithm. On the other hand, as the gene expression data is usually high dimensional and a large portion of gene objects are often uninformative to the function of interests. This presents a big challenge to co-clustering methods, which is the research problem of this paper.

In this paper, we propose a novel algorithm, called *Subspace Weighting Co-Clustering* (SWCC), for co-clustering gene expression data. In SWCC, a subspace weight matrix is introduced for weighting gene objects on each sample cluster. With the subspace weights, the contribution of each individual gene to the co-clusters can be identified from the weights. We design an objective function that uses the subspace weights in the distance function to determine the co-clusters of data. We also develop an iterative algorithm to optimize the co-clustering model, in which the subspace weights are automatically computed.

Three sets of experiments were conducted on both synthetic and benchmark data to study the proposed SWCC algorithm. The first set was used to study the properties of SWCC. The second set was used to compare the clustering performance of SWCC with six state-of-the-art clustering algorithms

- *X. Chen and J.Z. Huang are with the College of Computer Science and Software, Shenzhen University, Shenzhen, PR 518060, China. E-mail: {xjchen, zx.huang}@szu.edu.cn.*
- *Q. Wu is with the School of Software Engineering, South China University of Technology, Guangzhou, China, and State Key Laboratory for Novel Software Technology Nanjing University, PR 518060, China. E-mail: qyw@scut.edu.cn.*
- *M. Yang is with the Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong. E-mail: myang@cs.hku.hk.*

Manuscript received 31 Aug. 2016, accepted 9 May 2017, Date of publication 18 May 2017; date of current version 29 Mar. 2019.
(Corresponding author: Qingyao Wu.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2017.2705686

on ten gene expression data sets. Finally, we conducted experiments to show how to use SWCC for gene clustering and selection, and encouraging results are reported.

The rest of this paper is organized as follows. Section 2 gives a brief review of related work on co-clustering gene expression data. We present the subspace weighting co-clustering (SWCC) method in Section 3. Section 4 investigates the properties of the SWCC algorithm. The comparison results of SWCC with six clustering algorithms are reported and discussed in Section 5. The gene clustering and gene selection results are presented in Section 6. Conclusions and future work are given in Section 7.

2 Related Work

Co-clustering [12], also called bi-clustering [13], direct clustering [5], block clustering [14], is a process of simultaneously clustering rows and columns of a data matrix. The concept itself can be traced back to 1960's and 1970's [15], [16], although it had been rarely used or even studied. Until recently, co-clustering has received wide attention in many areas such as text mining [6], [7], [17], [18], bioinformatics [8], [9], [10] and recommendation systems [11]. Several co-clustering models have been formulated, including hierarchical co-clustering, sequential bi-clustering, spectral co-clustering and partitional co-clustering [19]. In this paper, we only focus on the partitional co-clustering.

Partitional co-clustering, first introduced in 1972 [5], has an advantage of efficiency in clustering large data. It uses an iterative process to partition a data matrix into $K \times L$ disjoint co-clusters. Recently, quite a few partitional co-clustering algorithms have been proposed to follow the partitional process. Lazzeroni et al. [20] presented an algorithm for finding plaid models, which is a form of overlapping two-way clustering with an embedded ANOVA (ANalysis Of VAriance) in each layer. Busygin et al. [21] proposed double conjugated clustering (DCC), a node-driven partitional co-clustering method, for microarray data analysis. Kluger et al. [22] proposed the spectral biclustering algorithm to simultaneously cluster genes and conditions of expression data, which finds distinctive "checkerboard" structures in the matrix of expression data using eigenvectors corresponding to the characteristic by expression pattern across genes or conditions. Govaert et al. [23] proposed a block mixture model and used the Generalized EM algorithm (GEM) to maximize a variational approximation of the likelihood of the model.

Dhillon et al. [6] treated a data matrix as a joint probability distribution between two discrete random variables that take values over rows and columns, and proposed an information-theoretic co-clustering (ITCC) to minimize the mutual information loss between the original random variables and the clustered random variables. Cho et al. [8] proposed the MSSRCC (Minimum Sum-Squared Residue Co-clustering) algorithm to minimize two different objective functions based on two different squared residue measures. Banerjee et al. [19] introduced a minimum Bregman information (MBI) principle that simultaneously generalizes the maximum entropy and the standard least squares. They proposed a Bregman Block Average co-clustering algorithm (BBAC) in which the approximation error is measured using a large class of loss functions called Bregman divergences that include the squared Euclidean distance and the KL-divergence as special cases. By selecting different Bregman divergences and co-clustering basis, different variants can be derived, e.g., ITCC and MSSRCC. However, high-dimensional data presents a big challenge to BBAC because the rows and columns in co-clusters are equally treated in BBAC and the noise and irrelevant values cannot be identified.

Deodhar et al. proposed a Robust Overlapping Co-Clustering (ROCC) algorithm, which aims to find dense, arbitrarily positioned, possibly overlapping co-clusters from large, noisy datasets [24]. Zhang et al. proposed a co-clustering algorithm named Locally Discriminative Co-clustering (LDCC) to explore the relationship between samples and features as well as the intersample and interfeature relationships [25]. In their method, the sample-feature relationship is modeled by a bipartite graph between samples and features, and the intersample and interfeature relationships are captured from the intrinsic discriminative structures of both sample space and feature space. Cheng et al. prosed a hierarchical co-clustering algorithm, which successively performs row-wise and column-wise splits that lead to the maximal mutual information gain at each step [26].

In the past decade, soft subspace clustering has been an important research topic in cluster analysis [27], [28], [29], [30]. Such methods are specially designed for high-dimensional data, by assigning weights to variables and discovering clusters from subspaces of the features with large weights [3], [30]. Recently, weighting technique was introduced into co-clustering.

Tjhi et al. [31] proposed a fuzzy co-clustering algorithm, FFCFW, which simultaneously assigns weights to co-clusters and computes the associations between rows and column clusters, and the associations between columns and row clusters. Three parameters were introduced into the method to adjust the co-cluster weights and two types of associations. Ye et al. [32] proposed a FWITCC co-clustering method for co-clustering document data, which assigns each feature a weight computed from the mutual information shared by the features and the documents. The informative words will be assigned big weights and noisy words will be assigned small weights. Sarazin et al. [33] proposed a feature group weighting co-clustering method on topological maps model, which assigns weights to co-clusters and learns the weights during the topological biclustering process.

Cluster analysis has been successfully applied to gene expression data over the last few decades (see a survey in [34]). For co-clustering of gene expression [9], several comparison papers have been published [35], [36]. In [35], Prelić et al. conducted a systematic comparison and evaluation of co-clustering methods for gene expression data on both synthetic data and real data. Synthetic data was use to test the effects of experimental noise, and the real data sets results were compared using Gene Ontology (GO) annotations. In [36], Eren et al. used the BiBench package to compare 12 algorithms on a suite of synthetic data sets to measure the performance on data with varying conditions, varying noise, varying number of biclusters and overlapping biclusters. Eight large gene expression data sets from the Gene Expression Omnibus were also used to test the algorithms.

3 Subspace Weighting Co-Clustering Algorithm

3.1 Problem Statement

Let $\boldsymbol{X} = [x_{i,j}]_{N\times M}$ be a data matrix with N rows and M columns. The objective of partitional co-clustering is to partition $\boldsymbol{X}$ into K row clusters and L column clusters. Formally, we want to discover two binary matrices $U = [u_{i,g}]_{N\times K}$ and $V = [v_{j,h}]_{M\times L}$, in which $u_{i,g} = 1$ indicates that the ith row object is assigned to the gth row cluster and $v_{j,h} = 1$ indicates that the jth column object is assigned to the hth column cluster.

In the gene expression data, $\boldsymbol{X}$ often has a small N and a very large M. Each object in SNP data can be identified by a small subset of genes. To deal with such kind of data, we define a subspace weighting co-clustering method. In this method, we define a subspace weight matrix $C = [c_{g,j}]_{K\times M}$ for columns on row clusters, in which $c_{g,j}$ is the weight of the jth column in the gth row cluster. With the subspace weighting matrix C, we can identify which subset of genes is most related to a class of patients. In the following section, we will propose a new partitional co-clustering algorithm in which the subspace weight matrix is introduced.

3.2 Objective Function

To cluster a data matrix $\boldsymbol{X}$ into K row clusters and L column clusters, we introduce a subspace weight matrix $C = [c_{g,j}]_{K\times M}$ into the distance function of BBAC-S[1] and formulate the problem as an iterative clustering process to minimize the following objective function

$$\begin{aligned} P(U,V,Z,C) = &\frac{1}{MN}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M} u_{i,g}v_{j,h}c_{g,j}d(x_{i,j},z_{g,h}) \\ &+ \frac{\eta}{M}\sum_{g=1}^{K}\sum_{j=1}^{M} c_{g,j}\log c_{g,j}, \end{aligned} \tag{1}$$

subject to

$$\begin{cases} \sum_{g=1}^{K} u_{i,g} = 1, & u_{i,g}\in\{0,1\}, \quad 1\le i\le N \\ \sum_{h=1}^{L} v_{j,h} = 1, & v_{j,h}\in\{0,1\}, \quad 1\le j\le M \\ \sum_{j=1}^{M} c_{g,j} = 1, & 0 < c_{g,j} < 1, \quad 1\le g\le K, \end{cases} \tag{2}$$

where

- $U = [u_{i,j}]_{N\times K}$ is a binary matrix in which $u_{i,g} = 1$ indicates that the ith row is assigned to the gth row cluster;
- $V = [v_{j,h}]_{M\times L}$ is a binary matrix in which $v_{j,h} = 1$ indicates that the jth column is assigned to the hth column cluster;
- $Z = [z_{g,h}]_{K\times L}$ is the centers of $K \times L$ co-clusters;
- $C = [c_{g,j}]_{K\times M}$ is a weight matrix where $c_{g,j}$ is the weight for the jth column in the gth row cluster;
- η is a positive regularization parameter. The bigger η is, the flatter the weights in the matrix C;
- $d(x_{i,j}, z_{g,h})$ is a distance defined as

$$d(x_{i,j},z_{g,h}) = (x_{i,j} - z_{g,h})^2. \tag{3}$$

The first term in (1) is the weighted sum of within-cluster dispersion, the second term is the penalty terms of negative weight entropies. By combining the two terms together, we can simultaneously minimize the within-cluster dispersion and maximize the weight entropies to stimulate more columns to contribute to the identification of co-clusters. In this way, we can avoid the problem of identifying co-clusters by few columns in sparse data.

3.3 SWCC Co-Clustering Algorithm

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

1) Problem P_1: $\arg\min_U P(U,\hat{V},\hat{Z},\hat{C})$;
2) Problem P_2: $\arg\min_V P(\hat{U},V,\hat{Z},\hat{C})$;
3) Problem P_3: $\arg\min_Z P(\hat{U},\hat{V},Z,\hat{C})$;
4) Problem P_4: $\arg\min_C P(\hat{U},\hat{V},\hat{Z},C)$.

We can verify that P_1 is solved by

$$\begin{cases} u^*_{i,g} = 1 \quad \text{if } P_{(g)} \le P_{(s)} \text{ for } 1\le s\le K \text{ where} \\ \qquad P_{(s)} = \sum_{h=1}^{L}\sum_{j=1}^{M} \hat{v}_{j,h}\hat{c}_{s,j}d(x_{i,j},\hat{z}_{s,h}) \\ u^*_{i,s} = 0 \quad \text{for } s\ne g, \end{cases} \tag{4}$$

and P_2 is solved by

$$\begin{cases} v_{j,h} = 1 \quad \text{if } P_{(h)} \le P_{(t)} \text{ for } 1\le t\le L \text{ where} \\ \qquad P_{(t)} = \sum_{g=1}^{K}\sum_{i=1}^{N} \hat{u}_{i,g}\hat{c}_{g,j}d(x_{i,j},\hat{z}_{g,t}) \\ v_{j,t} = 0 \quad \text{for } t\ne h, \end{cases} \tag{5}$$

and P_3 is solved by

$$z_{g,h} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\hat{u}_{i,g}\hat{v}_{j,h}\hat{r}_{h,i}\hat{c}_{g,j}x_{i,j}}{\sum_{i=1}^{N}\sum_{j=1}^{M}\hat{u}_{i,g}\hat{v}_{j,h}\hat{c}_{g,j}}. \tag{6}$$

Theorem 1. *P_4 is solved by*

$$c^*_{g,j} = \frac{\exp\left\{\frac{-E_{g,j}}{\eta}\right\}}{\sum_{j'=1}^{M}\exp\left\{\frac{-E_{g,j'}}{\eta}\right\}}, \tag{7}$$

where

$$E_{g,j} = \frac{1}{N}\sum_{h=1}^{L}\sum_{i=1}^{N}\hat{u}_{i,g}\hat{v}_{j,h}\,d(x_{i,j},\hat{z}_{g,h}). \tag{8}$$

Proof. We form the following Lagrangian to minimize $P(\hat{U},\hat{V},\hat{Z},C)$

$$\begin{aligned} \mathcal{L}(C,\gamma,\tau) = \frac{1}{M}\sum_{g=1}^{K}\Bigg[&\sum_{j=1}^{M}E_{g,j}c_{g,j} + \eta\sum_{j=1}^{M}c_{g,j}\log c_{g,j} \\ &+ \gamma_g\left(\sum_{j=1}^{M}c_{g,j} - 1\right) - \sum_{j=1}^{M}c_{g,j}\tau_{g,j}\Bigg], \end{aligned} \tag{9}$$

1. Bregman Block Average Co-clustering with the Squared Euclidean distance.

where E_{gj} is constantly calculated by (8), $[\gamma_1, \ldots, \gamma_K]$ are K Lagrange multipliers and $[\tau_{11}, \ldots, \tau_{K,M}]$ are $K \times M$ positive Lagrange multipliers.

According to the KKT condition[37], we have,

$$\begin{cases} \frac{\partial \mathcal{L}(C,\gamma,\tau)}{\partial c_{g,j}} = \frac{1}{M}\left[E_{g,j} + \eta(1 + \log c_{g,j}) + \gamma_g - \tau_{g,j}\right] = 0 \\ \frac{\partial \mathcal{L}(C,\gamma,\tau)}{\partial \gamma} = \sum_{j=1}^{M} c_{g,j} - 1 = 0 \\ \tau_{g,j} = 0, \ \forall g, j. \end{cases} \tag{10}$$

From (10), we have

$$c_{g,j} = \exp\left\{-\frac{E_{g,j}}{\eta}\right\}\exp\left\{-\frac{\gamma_g + \eta}{\eta}\right\}. \tag{11}$$

Substituting (11) into $\sum_{j=1}^{M} c_{g,j} - 1 = 0$, we have

$$\exp\left\{-\frac{\gamma_g + \eta}{\eta}\right\} = \frac{1}{\sum_{j=1}^{M} \exp\left\{-\frac{E_{g,j}}{\eta}\right\}}. \tag{12}$$

Substituting the above equality into (11), we have

$$c_{g,j} = \frac{\exp\left\{-\frac{E_{g,j}}{\eta}\right\}}{\sum_{j'=1}^{M} \exp\left\{-\frac{E_{g,j'}}{\eta}\right\}}. \tag{13}$$

which is equivalent to $c^*_{h,j}$ in Eq. (7)

According to the second derivative test, if $\eta > 0$, the second order derivative of $\frac{\partial^2 \mathcal{L}(C,\gamma,\tau)}{\partial c^2_{g,j}} = \frac{\eta}{c_{g,j}} > 0$ at c^*_{gj}, then c^*_{gj} is the local minima of $P(\hat{U}, \hat{V}, \hat{Z}, C)$. □

Algorithm 1. SWCC

Input: X, K, L and η;
Output: U, V, Z and C;
Init: Let $c_{g,j} = \frac{1}{K}$ and start with an arbitrary partitional co-clustering.
$t := 0$
repeat
  Update U^{t+1} by (4);
  Update V^{t+1} by (5);
  Update Z^{t+1} by (6);
  Update C^{t+1} by (7) and (8);
  $t := t + 1$
until (1) obtains its local minimum value;

The SWCC algorithm that minimizes the objective function (1) is given as Algorithm 1. The objective function (1) can be minimized by alternatively solving the four minimization problems $P_1, \ldots, P_4$. Therefore, the sequence of $P(.,.,.)$ generated by Algorithm 1 is strictly decreasing. Finally, the algorithm will converge to a local solution.

If the SWCC algorithm needs r iterations to converge, its computational complexity is $O(rNMKL)$, which is the same as the computational complexity of BBAC. Therefore, the new clustering algorithm remains efficient in clustering large high-dimensional data.

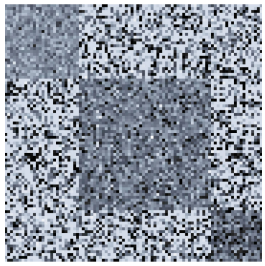


Fig. 1. Plot of a typical synthetic data set D_1.

4 Experiments on Properties of SWCC

In this section, we present a typical synthetic data and a series of experiments on the data set to investigate the properties of SWCC.

4.1 Experiment Setup

Fig. 1 shows a typical synthetic data set D_1 with 90 rows and 90 columns, where the darker color indicates the higher value. D_1 can be divided into 9 blocks, in which the three dark blocks along the diagonal line are co-clusters. Other blocks contain noise.

In the experiments, we used D_1 to investigate the subspace weights of the SWCC algorithm. We set $K = L = 3$ and chose a geometric sequence of 20 real values, $\{1.2^0E - 4, \ldots, 1.2^{19}E - 4\}$ for η. For each η, we randomly generated 100 initial cluster centers and ran each algorithm to generate 100 results. Finally, we produced 2, 000 co-clustering results for the following analysis.

We used the normalized mutual information (NMI) to evaluate the clustering result [38]

$$NMI(\Omega, \mathbb{C}) = \frac{I(\Omega, \mathbb{C})}{(H(\Omega) + H(\mathbb{C}))/2},$$

where $\Omega = [\omega_g]_K$ is the clusters obtained by the clustering algorithm and $\mathbb{C} = [c_g]_K$ is the true inherent classes, I is the mutual information defined as

$$I(\Omega, \mathbb{C}) = \sum_{k=1}^{K}\sum_{j=1}^{J} P(\omega_k \cap c_j)\log\frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_j)},$$

and H is the entropy defined as

$$H(\Omega) = -\sum_{k=1}^{K} P(\omega_k)\log P(\omega_k),$$

where $P(\omega_k)$ is the probability of an object being in cluster ω_k, $P(c_j)$ is the probability of an object being in class c_j, and $P(\omega_k \cap c_j)$ is the joint probability that an object lines in cluster ω_k and class c_j.

4.2 Impacts of η on C

For C in each clustering result, we first computed the entropy of 90 elements in each row cluster g, then we computed the average entropy of 3 row clusters. The higher the entropy, the evener the corresponding weights. The average entropies with each η were plotted in Fig. 2. We can see that the average entropy of C was stable at the start, then it decreased rapidly to its lowest value as η increased. After that, it increased rapidly as η increased and finally became stable.

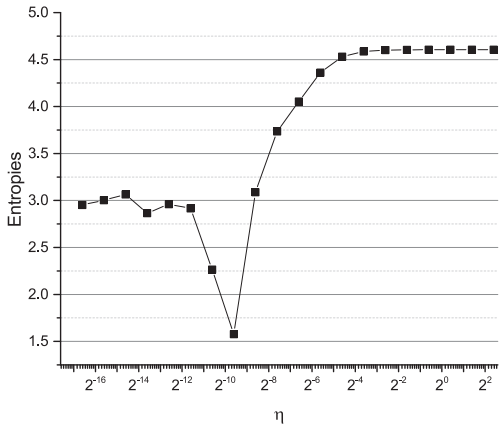
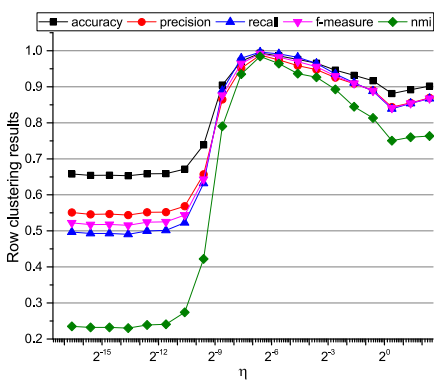


Fig. 2. The average entropies of C versus η from the results of SWCC on D_1.

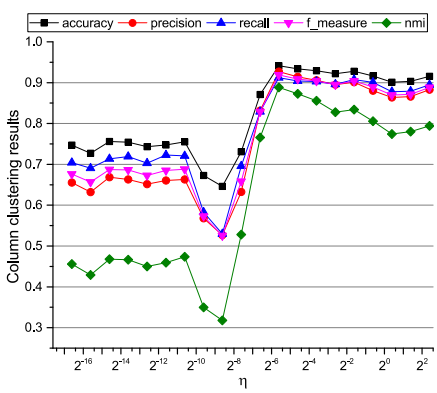
4.3 Impacts of η on Co-Clustering Performance

From each clustering result of SWCC, we separately computed the normalized mutual information NMI_R for row cluster results and NMI_C for column cluster results and computed the average values from 100 results for each η. The results are plotted in Fig. 3 (similar as Fig. 2).

From Fig. 3, we can see that both NMI_C and NMI_R were low with small η, then they increased rapidly to their highest values. Finally, they slightly dropped and became nearly stable. The weights are too concentrated on a few variables with too small parameters and are evenly distributed with too big parameters, and both situations often lead to bad clustering results. In order to produce good clustering results, η can not be set too big or too small. Moreover, cluster ensembles can be used to combine multiple clustering results to produce better clustering results [39].



(a) Row clustering results versus η.



(b) Column clustering results versus η.

Fig. 3. The NMI values of row and column clustering results over η from the results of SWCC on D_1.

Fig. 4. Weight matrix of SWCC on D_1.

4.4 Subspace Weight

NMI_R and NMI_C from each co-clustering result were computed and the highest values for SWCC are 1.0 and 0.76. For the best co-clustering result, we computed a weight matrix $W = [w_{ij}]_{m\times n}$ where $w_{ij} = \sum_g \sum_h u_{i,g} v_{j,h} c_{g,j}$ is the weight assigned to the ith row and jth column. The weight matrices were plotted in Fig. 4, in which the darker color indicates higher weight. We can see that the weights of SWCC reveals column subspace structure, which helps the identification of row clusters.

4.5 Scalability Analysis

We generated a group of synthetic data sets for comparing the scalability of SWCC with BBAC-S [19], ITCC [6], FFCFW [31] and HICC [26]. The number of rows in these data sets were fixed to 1,000, whereas the numbers of columns were set as 10 numbers ranging from 200 to 12,800. The input values of parameters of FFCFW and SWCC were set as 0.01. For each data, we randomly chose 100 different sets of initial cluster centers to produce 100 clustering results for each algorithm and the average execution times were plotted in Fig. 5. We can see that most algorithms have

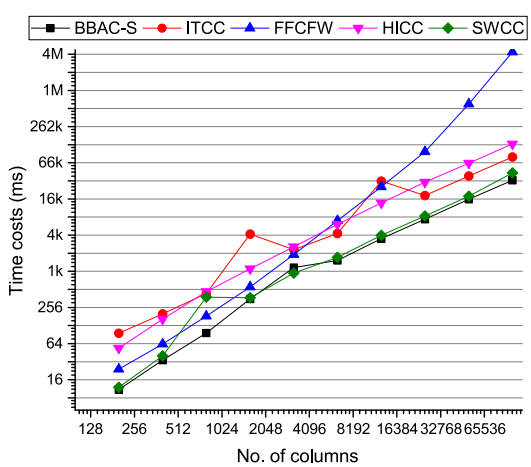


Fig. 5. Average time costs versus the number of columns.

TABLE 1
Characteristics of 10 Gene Expression Data Sets

Name	Abbr.	#Genes	#Patients	#Classes
adencarcinma	*ADE*	9,868	76	2
brain	*BRA*	5,597	42	5
breast.2.class	*BR2*	4,869	78	2
breast.3.class	*BR3*	4,869	96	3
colon	*COL*	2,000	62	2
leukemia	*LEU*	3,051	38	2
lymphoma	*LYM*	4,026	62	3
nci 60	*NCI*	5,244	61	8
prostate	*PRO*	6,033	102	2
srbct	*SRB*	2,308	63	4

nearly linear relationships between the execution time and the number of columns in data. FFCFW showed the worst scalability to the number of columns. The times costs of SWCC were comparable to those of BBAC-S. This indicates that SWCC scales well to large high-dimensional data.

5 Experiments on Clustering

In this section, we introduced ten gene expression data sets to compare the clustering results of SWCC with those of six clustering algorithms, including both one-way and co-clustering algorithms.

TABLE 2
Comparisons of Clustering Results by SWCC with Six Clustering Algorithms on 10 Gene Expression Data Sets

Data	evaluation	k-means	EWKM	BBAC-S	ITCC	FFCFW	HICC	SWCC
ADE	Accuracy	0.61(0.73)	0.72(0.73)*	0.57(0.71)*	0.55(0.77)*	0.73(0.73)*	0.50(0.54)*	0.64(0.81)
	Precision	0.73(0.79)	0.73(0.74)*	0.73(0.79)	0.73(0.79)	0.73(0.73)*	0.73(0.76)*	0.74(0.85)
	Recall	0.73(1.00)*	0.97(1.00)*	0.66(0.94)*	0.61(0.93)*	0.68(1.00)*	0.50(0.56)*	0.80(1.00)
	F-Measure	0.72(0.84)	0.83(0.84)*	0.69(0.83)*	0.66(0.85)*	0.84(0.84)*	0.59(0.64)*	0.76(0.87)
	NMI	0.01(0.07)	0.00(0.03)*	0.00(0.07)*	0.01(0.16)*	0.00(0.00)*	0.01(0.09)*	0.02(0.30)
BRA	Accuracy	0.72(0.83)	0.45(0.72)*	0.77(0.82)*	0.20(0.20)*	0.20(0.20)*	0.63(0.70)*	0.73(0.86)
	Precision	0.39(0.54)	0.24(0.38)*	0.43(0.55)*	0.20(0.20)*	0.19(0.26)*	0.19(0.26)	0.37(0.63)
	Recall	0.64(0.76)*	0.78(1.00)*	0.48(0.58)*	1.00(1.00)*	1.00(1.00)*	0.28(0.59)*	0.55(1.00)
	F-Measure	0.48(0.63)	0.36(0.48)*	0.46(0.56)*	0.33(0.33)*	0.33(0.33)*	0.22(0.30)*	0.43(0.67)
	NMI	0.45(0.62)	0.19(0.43)*	0.45(0.58)*	0.00(0.00)*	0.00(0.00)*	0.13(0.25)*	0.40(0.70)
BR2	Accuracy	0.54(0.55)*	0.51(0.57)*	0.54(0.55)*	0.50(0.50)*	0.50(0.50)*	0.50(0.56)*	0.53(0.57)
	Precision	0.53(0.54)*	0.51(0.57)*	0.53(0.54)*	0.50(0.50)*	0.51(0.56)*	0.51(0.56)*	0.53(0.56)
	Recall	0.71(0.72)*	0.92(1.00)*	0.71(0.97)	1.00(1.00)*	1.00(1.00)*	0.50(0.56)*	0.69(1.00)
	F-Measure	0.61(0.61)*	0.65(0.67)*	0.61(0.66)*	0.67(0.67)*	0.67(0.67)*	0.50(0.56)*	0.59(0.67)
	NMI	0.05(0.07)	0.01(0.10)*	0.04(0.07)*	0.00(0.00)*	0.00(0.00)*	0.01(0.10)*	0.06(0.13)
BR3	Accuracy	0.62(0.63)*	0.46(0.62)*	0.62(0.64)*	0.36(0.36)*	0.37(0.62)*	0.53(0.56)*	0.60(0.65)
	Precision	0.48(0.49)*	0.40(0.49)*	0.48(0.50)*	0.36(0.36)*	0.37(0.49)*	0.36(0.39)*	0.46(0.52)
	Recall	0.57(0.73)	0.85(1.00)*	0.50(0.70)*	1.00(1.00)*	1.00(1.00)*	0.39(0.47)*	0.60(0.83)
	F-Measure	0.52(0.58)	0.53(0.58)*	0.49(0.57)*	0.53(0.53)*	0.53(0.58)*	0.38(0.42)*	0.52(0.58)
	NMI	0.21(0.26)	0.08(0.27)*	0.21(0.27)	0.00(0.00)*	0.00(0.26)*	0.02(0.07)*	0.22(0.29)
COL	Accuracy	0.52(0.64)	0.53(0.70)*	0.53(0.77)	0.53(0.53)*	0.53(0.53)*	0.50(0.53)*	0.51(0.80)
	Precision	0.55(0.63)	0.55(0.74)	0.56(0.80)	0.53(0.53)*	0.53(0.53)*	0.53(0.57)*	0.54(0.82)
	Recall	0.58(0.81)	0.83(1.00)*	0.56(0.76)*	1.00(1.00)*	1.00(1.00)*	0.52(0.71)*	0.60(1.00)
	F-Measure	0.56(0.71)	0.64(0.71)*	0.56(0.78)	0.70(0.70)*	0.70(0.70)*	0.52(0.62)*	0.56(0.81)
	NMI	0.04(0.21)	0.02(0.33)	0.04(0.45)	0.00(0.00)*	0.00(0.00)*	0.01(0.08)*	0.04(0.49)
LEU	Accuracy	0.70(0.95)	0.65(1.00)	0.72(1.00)*	0.58(0.58)*	0.58(0.58)*	0.50(0.58)*	0.67(1.00)
	Precision	0.75(0.94)	0.67(1.00)*	0.77(1.00)*	0.58(0.58)*	0.58(0.58)*	0.56(0.61)*	0.72(1.00)
	Recall	0.74(1.00)	0.90(1.00)*	0.75(1.00)	1.00(1.00)*	1.00(1.00)*	0.61(0.78)*	0.71(1.00)
	F-Measure	0.74(0.96)	0.75(1.00)*	0.76(1.00)*	0.73(0.73)*	0.73(0.73)*	0.58(0.68)*	0.71(1.00)
	NMI	0.42(0.83)	0.21(1.00)*	0.43(1.00)	0.00(0.00)*	0.00(0.00)*	0.05(0.20)*	0.34(1.00)
LYM	Accuracy	0.73(0.97)	0.63(0.95)*	0.73(0.95)*	0.50(0.50)*	0.54(0.88)*	0.50(0.57)*	0.68(0.97)
	Precision	0.82(0.99)*	0.62(0.91)*	0.83(0.98)*	0.50(0.50)*	0.54(0.86)*	0.50(0.59)*	0.74(0.99)
	Recall	0.57(0.96)	0.80(1.00)*	0.58(0.92)	1.00(1.00)*	0.98(1.00)*	0.40(0.66)*	0.56(1.00)
	F-Measure	0.67(0.97)	0.69(0.95)*	0.68(0.95)*	0.67(0.67)*	0.69(0.89)*	0.44(0.58)*	0.63(0.97)
	NMI	0.61(0.93)*	0.33(0.85)*	0.62(0.88)*	0.00(0.00)*	0.07(0.69)*	0.04(0.18)*	0.47(0.93)
NCI	Accuracy	0.86(0.89)*	0.47(0.76)*	0.87(0.88)*	0.11(0.11)*	0.49(0.79)*	0.65(0.75)*	0.84(0.91)
	Precision	0.40(0.51)*	0.15(0.27)*	0.44(0.58)*	0.11(0.11)*	0.17(0.29)*	0.12(0.16)*	0.35(0.62)
	Recall	0.50(0.62)*	0.74(1.00)*	0.49(0.60)*	1.00(1.00)*	0.77(1.00)*	0.32(0.68)*	0.44(0.88)
	F-Measure	0.45(0.55)*	0.25(0.38)*	0.46(0.59)*	0.21(0.21)*	0.27(0.38)*	0.17(0.22)*	0.39(0.61)
	NMI	0.59(0.69)*	0.23(0.47)*	0.61(0.71)*	0.00(0.00)*	0.24(0.51)*	0.15(0.28)*	0.53(0.73)
PRO	Accuracy	0.51(0.51)*	0.50(0.52)*	0.51(0.51)*	0.50(0.50)*	0.50(0.50)*	0.50(0.52)*	0.51(0.54)
	Precision	0.50(0.50)*	0.50(0.51)*	0.50(0.50)*	0.50(0.50)*	0.50(0.50)*	0.49(0.51)*	0.50(0.54)
	Recall	0.54(0.54)*	0.79(1.00)*	0.54(0.54)*	1.00(1.00)*	1.00(1.00)*	0.60(0.74)*	0.56(0.87)
	F-Measure	0.52(0.52)*	0.60(0.66)*	0.52(0.52)*	0.66(0.66)*	0.66(0.66)*	0.54(0.61)*	0.53(0.63)
	NMI	0.02(0.02)*	0.01(0.07)*	0.02(0.02)*	0.00(0.00)*	0.00(0.00)*	0.01(0.08)*	0.03(0.09)
SRB	Accuracy	0.67(0.78)*	0.46(0.67)*	0.64(0.76)*	0.27(0.27)*	0.27(0.27)*	0.58(0.63)*	0.61(0.75)
	Precision	0.43(0.60)*	0.30(0.44)*	0.35(0.55)*	0.27(0.27)*	0.27(0.27)*	0.28(0.34)*	0.32(0.55)
	Recall	0.47(0.59)	0.71(1.00)*	0.37(0.56)*	1.00(1.00)*	1.00(1.00)*	0.34(0.49)*	0.48(1.00)
	F-Measure	0.44(0.59)*	0.42(0.58)*	0.36(0.56)	0.43(0.43)*	0.43(0.43)*	0.30(0.37)*	0.37(0.55)
	NMI	0.39(0.61)*	0.14(0.43)*	0.26(0.59)*	0.00(0.00)*	0.00(0.00)*	0.07(0.18)*	0.20(0.51)

Note: Underlined numbers are the best results on the corresponding data sets.
** Significant at 5 percent level.*

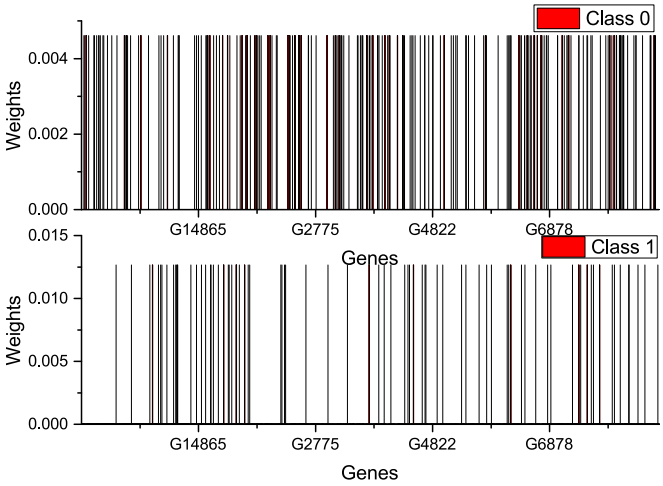


Fig. 6. The gene weights of the 10 gene clusters on the 2 classes of the ADE data set.

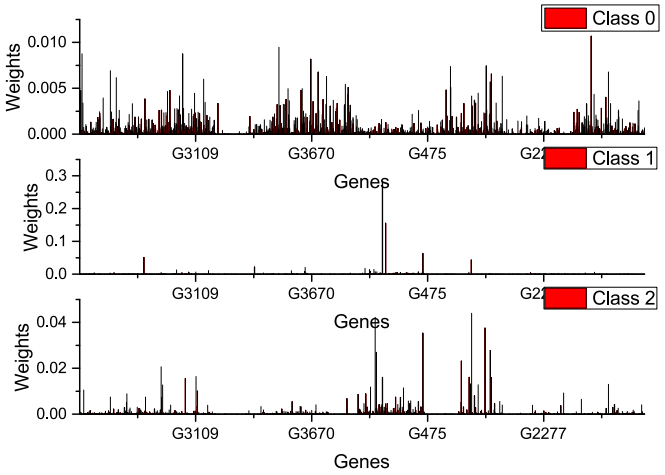


Fig. 8. The gene weights of the 15 gene clusters on the 3 classes of the $BR3$ data set.

5.1 Experiment Setup

To verify the effectiveness of our method for gene expression data, we conducted experiments on 10 gene expression data sets which were used in [40]. The characteristics of these data sets are given in Table 1. Since the number of genes of these data sets are much larger than the number of patients, it makes sense to use the subspace weighting co-clustering method.

In this experiment, we selected ten gene expression data sets to compare SWCC with six clustering methods, i.e., k-means, EWKM [28], BBAC-S [19], ITCC [6], the fuzzy weighting co-clustering method FFCFW [31] and HICC [26].

For each data set, we set K as the number of classes in the data. The number of column clusters were set as four values, i.e., $L = \{5, 10, 15, 20\}$. We also chose a geometric sequence of 20 real values $\{1.2^0E-4, \ldots, 1.2^{19}E-4\}$ for EWKM, FFCFW and SWCC. 100 initial cluster centers were randomly selected to run each algorithm with each parameter setting.

In order to compare the classification performance in the first experiment, we used precision, recall, f-measure, accuracy and NMI to evaluate the results. The precision is calculated as the fraction of correct objects among those that the algorithm is believed to belong to the relevant class. The recall is the fraction of actual objects that were identified. The f-measure is the harmonic mean of precision and recall and accuracy is the proportion of correctly classified objects. NMI is defined in Section 5.1. All five indices use the corresponding actual classification as the reference classification. To statistically compare the clustering performance, the paired t-test comparing SWCC with the other clustering methods was computed from each of the five evaluation indices.

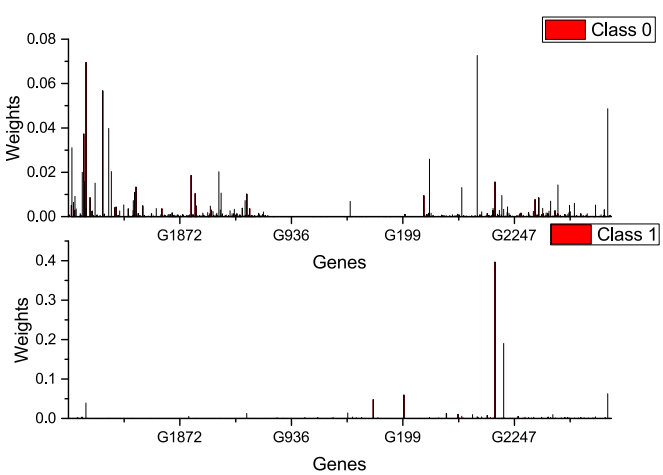


Fig. 7. The gene weights of the 10 gene clusters on the 2 classes of the $BR2$ data set.

5.2 Experimental Results and Analysis

For each clustering result, we computed five validations introduced in the last section. The results were listed in Table 2, where the value ahead of bracket is the average value and the value in bracket is the maximum value. A "*" is closely placed to the brackets if the difference of average value between the corresponding algorithm (excluding SWCC) and SWCC is significant by t-test, i.e., the p-value of t-test is less than 0.05. The best average and maximum values for each evaluation index on each data set are underlined.

k-means outperformed all co-clustering methods and EWKM on the SRB data set, which indicates that there exists no subspace structure on this data set. On most data sets, BBAC-S produced better clustering results than two one-way clustering methods, k-means and EWKM. This indicates co-clustering method is effective than one-way clustering method for gene expression data. Although ITCC and FFCFW produced good results in terms of accuracy, recall and f-measure, their NMI values on 8 data sets were nearly close to zero, indicating that these results can be considered as random assignments. In terms of maximal results, SWCC significantly outperformed other six algorithms on most data sets. However, BBAC-S outperformed SWCC on most data sets in terms of average results. This is due to the introduction of subspace weights, SWCC can not produce stable results as BBAC-S. In real applications, we can run SWCC multiple times to select good results.

6 Experiments on Gene Clustering and Selection

Since the class labels in the ten data sets in Table 1 are known, we can incorporate the label information of these

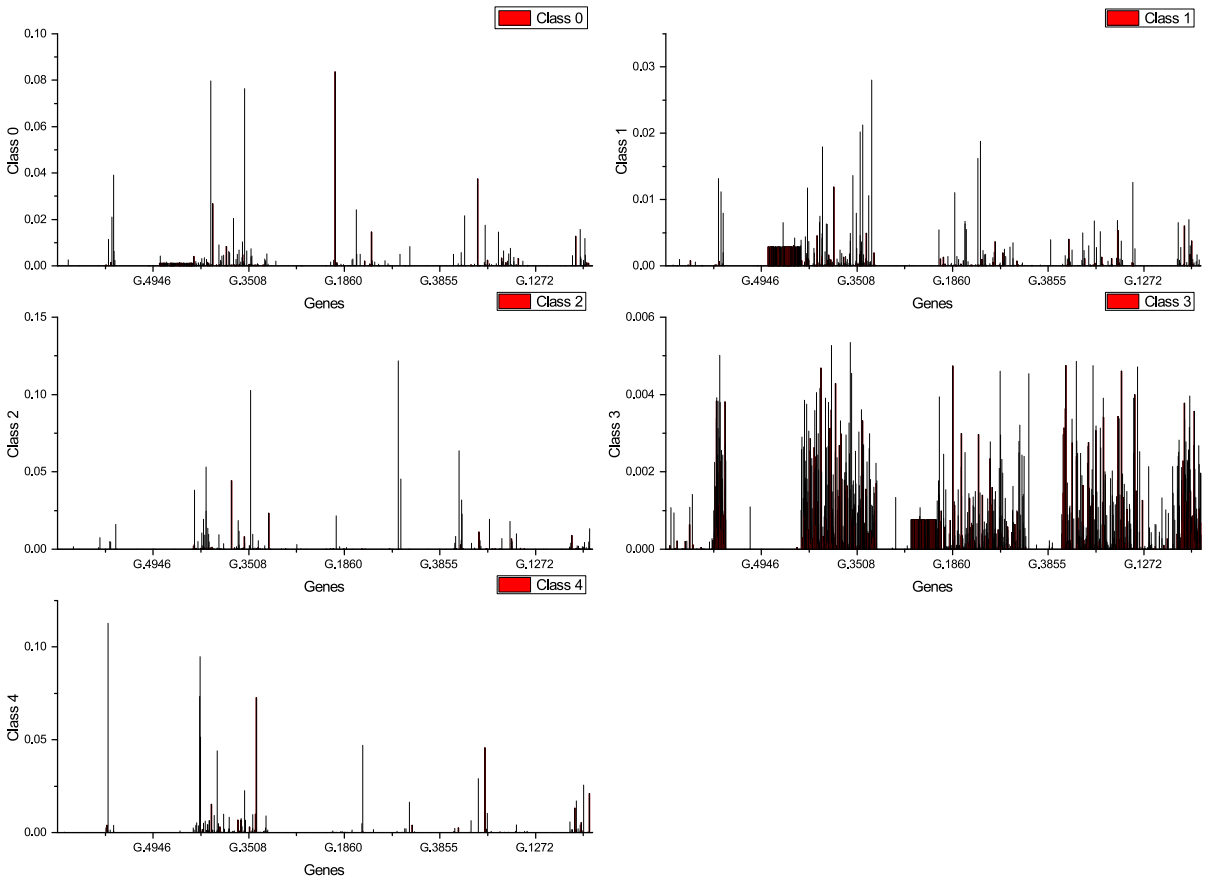


Fig. 9. The gene weights of 15 gene clusters on 5 classes of the BRA data set.

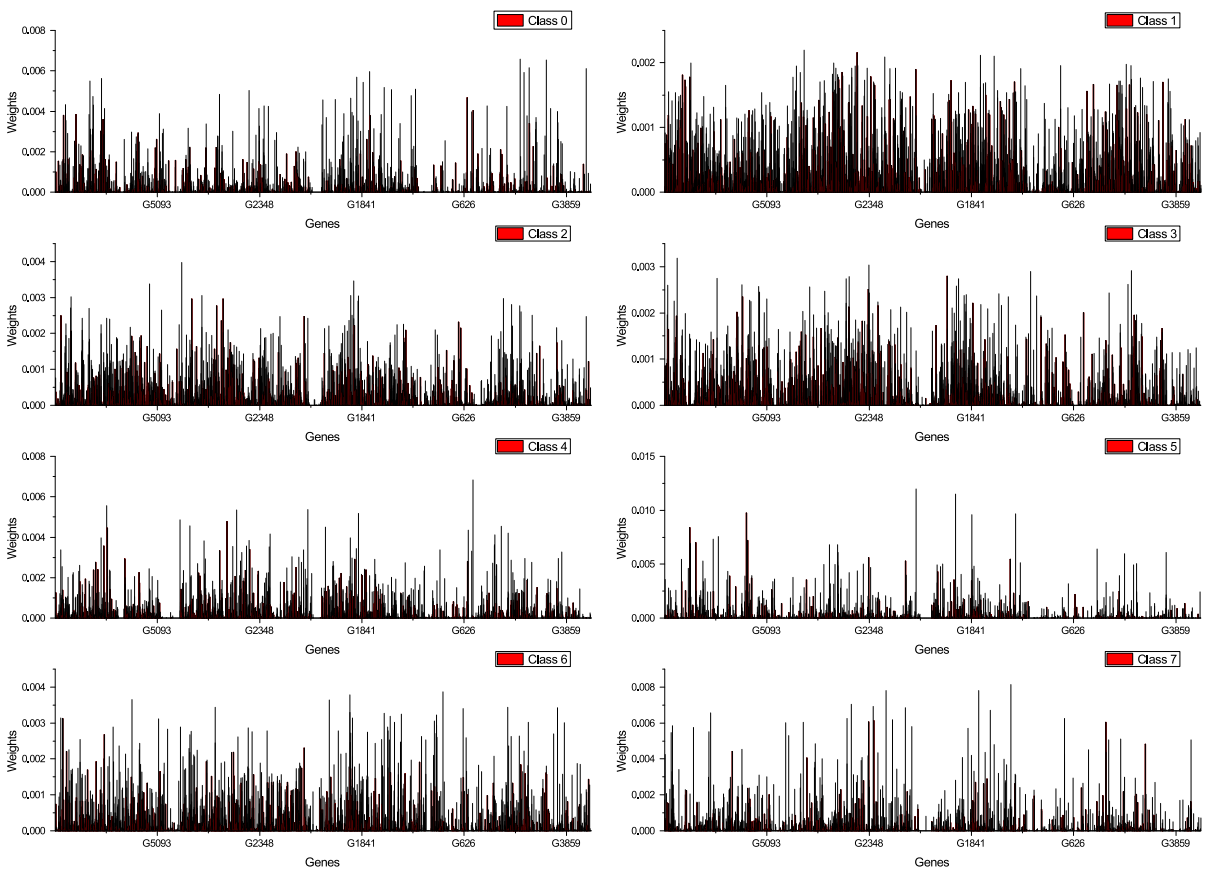


Fig. 10. The gene weights of 10 gene clusters on 8 classes of the NCI data set.

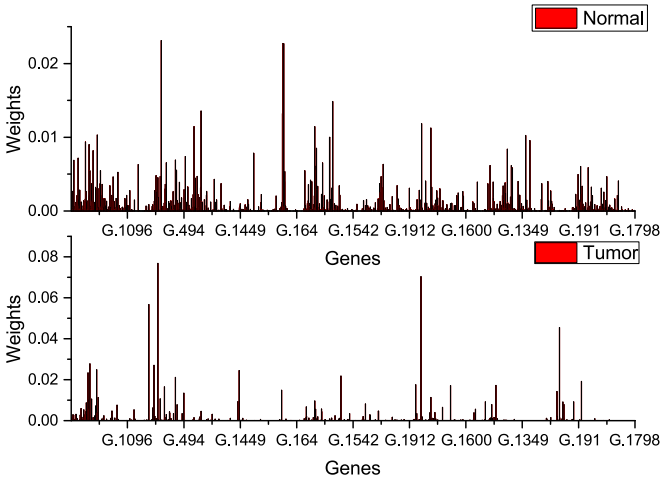


Fig. 11. The gene weights of 10 gene clusters on 2 classes of the COL data set.

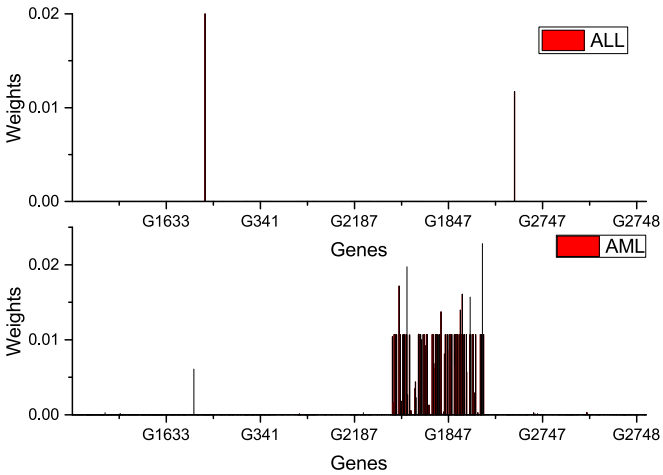


Fig. 12. The gene weights of 5 gene clusters on 2 classes of the LEU data set.

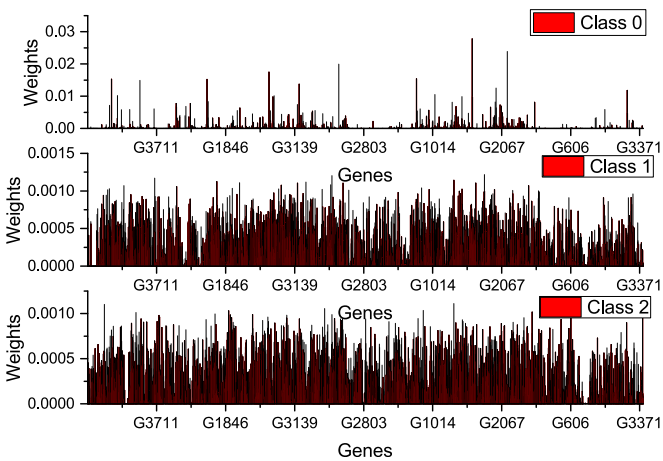


Fig. 13. The gene weights of 5 gene clusters on 3 classes of the LYM data set.

data sets to improve co-clustering. In this section, we used SWCC to cluster genes with the given class labels. We also used SWCC to select important genes, and the results show that the classification method can produce good results with a few important genes.

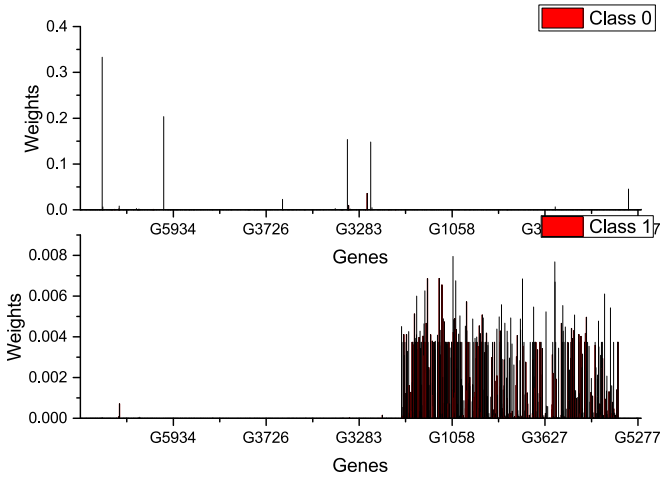


Fig. 14. The gene weights of 5 gene clusters on 2 classes of the PRO data set.

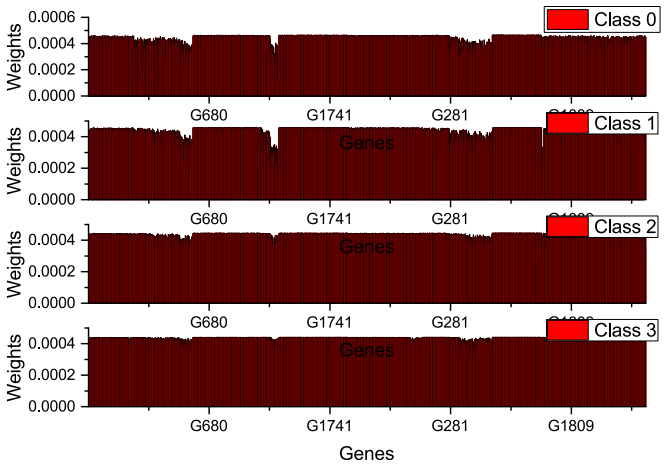


Fig. 15. The gene weights of 20 gene clusters on 4 classes of the SRB data set.

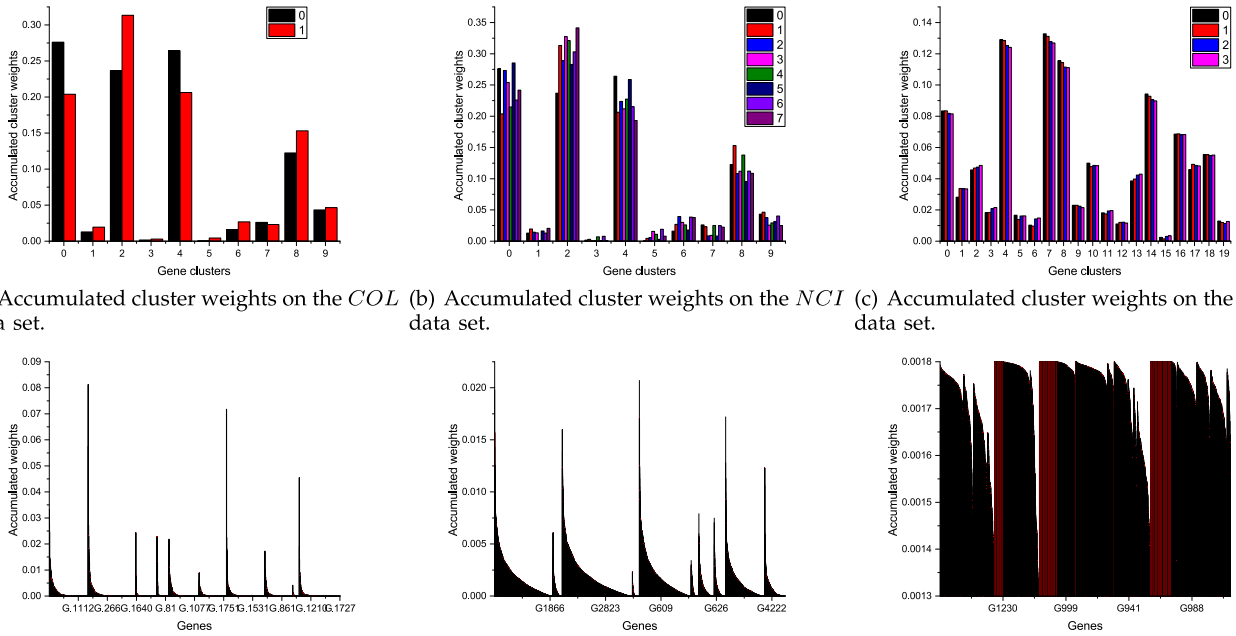
6.1 Experiment Setup

In the experiment, we incorporate the label information into SWCC to improve clustering genes on the 10 data sets in Table 1. Instead of co-clustering both rows and columns of the data, we fix the row clusters as the label prior in a data set, and only cluster the genes to seek the gene clusters. Similar as the experiment in Section 5.1, the number of gene clusters were set as $L = \{5, 10, 15, 20\}$. We also choose a geometric sequence of 20 real values $\{1.2^0E-4, \ldots, 1.2^{19}E-4\}$ to run SWCC.

6.2 Gene Clustering

To visualize the weights of SWCC, we computed the average weights of gene clusters on different classes of 10 gene expression data sets, and plotted them in Figs. 6∼15. In these figures, the genes were sorted such that genes in the same gene cluster were close to each other. The important genes can be identified for different data sets according to the weight values in these figures.

From Fig. 15, we can see that the cluster weights on the 4 gene clusters are almost even. According to (7), the subspace weight $c_{g,j}$ is inversely proportional to the within cluster dispersion of the gth cluster on the jth gene. The results



(a) Accumulated cluster weights on the COL data set. (b) Accumulated cluster weights on the NCI data set. (c) Accumulated cluster weights on the SRB data set.

(d) Accumulated gene weights of 10 gene clusters on the COL data set. (e) Accumulated gene weights of 10 gene clusters on the NCI data set. (f) Accumulated gene weights of 20 gene clusters on the SRB data set.

Fig. 16. The accumulated cluster weights and accumulated gene weights by SWCC on the COL, NCI and SRB data sets.

in Fig. 15 indicate that each gene has nearly equal within cluster dispersion on different clusters. This can explain why k-means outperforms SSWC on the SRB data set (See Table 2).

On the other 9 data sets except the SRB data set, we can observe clear subspace structures. For example on the COL data set, the genes with high weights lie in different gene clusters on different classes. Similar phenomena can be observed on the other 8 data sets. We also observe that on the PRO and LEU data sets, almost all genes in some gene clusters have high weights, while almost all genes in some other gene clusters have weights which are close to zero.

To check the characteristic of the gene cluster weights, we selected the weights on the COL, NCI and SRB data sets, and computed the accumulated gene weights on all classes for each gene. The results are plotted in Fig. 16. From these figures, we can see that on the COL data set, the clusters 0 and 1 have the highest weights accumulated on both classes. On the NCI data set, the clusters 2 and 0 and 4 have the highest weights accumulated on eight classes. On the SRB data set, the clusters 7 and 4 and 8 have the highest weights accumulated on four classes.

6.3 Gene Selection

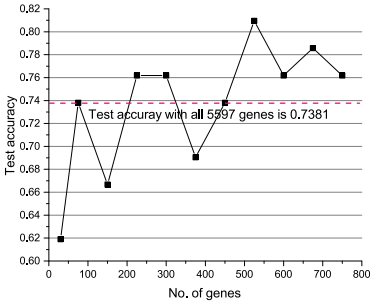
Selecting a subset genes which are highly correlated to the classes is a very important task [41]. In the gene clustering result by SWCC, genes in the same gene cluster have smaller distance to each other than to those in other gene clusters. If we select genes with high accumulated weights, the selected genes may concentrate on a small number of gene clusters. For example, we computed the accumulated gene weights of the COL, NCI and SRB data sets and the results are plotted in Fig. 16. From these figures, we can see that if we select a small number of genes only according to its accumulated weight, the selected genes may concentrate on only one gene cluster.

In this paper, we propose a gene-cluster-based gene selection method. We first compute the accumulated gene weights on all classes. For each cluster, we select h genes with the highest accumulated weights. The genes selected from all gene clusters are combined together to sample a subset of gene expression data.

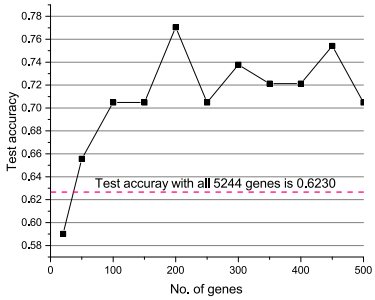
We set $h = \{2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$ to form 11 subset data sets for each of 10 data sets. To test the effectiveness of our gene selection method, the 11 subset data sets and the full set of each of 10 data sets were tested by a 10-fold Random Forests (RF) [42], which is a popular classification method for clustering gene expression data. 100 decision trees were built for each model. We computed the test accuracies of each classification result, the results were drawn in Fig. 17. We can see that on almost all data sets, RF produced good results with subsets of genes. On the BRA, NCI and SRB data sets, RF produced better results with a small proportion of selected genes than the results with all genes. For example, on the NCI data set, RF produced better result with less than 1 percent genes than the result with all genes. Surprisingly, on the ADE, $BR2$ and $BR3$ data sets, RF produced better results with subsets of genes than the result with all genes. On the other four data sets, with a small proportion of selected genes, RF produced results which is very close to the results with all genes. For example, on the LEU data set, RF produced the same result with less than 1 percent genes as the result with all genes. This indicates that SWCC can be used to select a subset of important genes without performance deterioration.

7 Conclusion

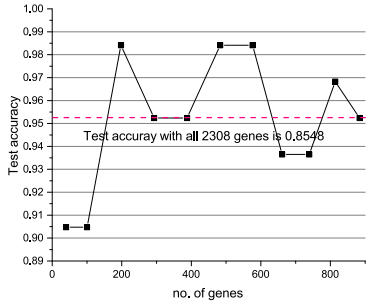
In this paper, we have presented a novel *Subspace Weighting Co-Clustering* algorithm. In the new method, a set of subspace weights have been introduced to weight gene



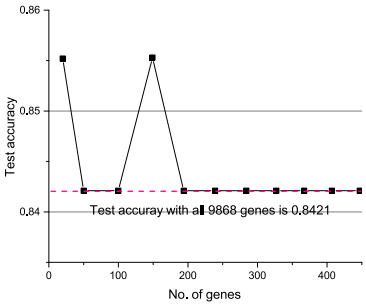
(a) Results on the BRA data set.



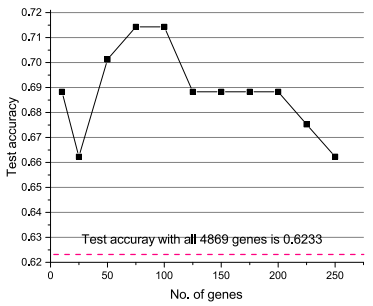
(b) Results on the NCI data set.



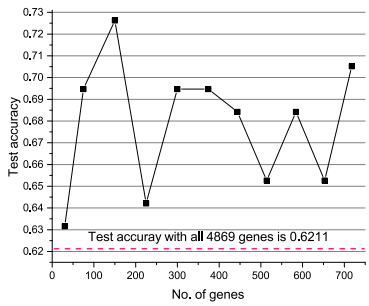
(c) Results on the SRB data set.



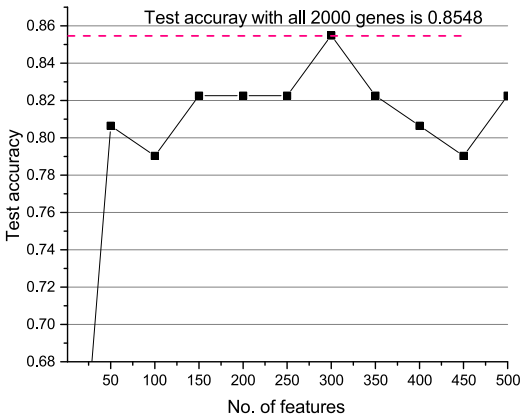
(d) Results on the ADE data set.



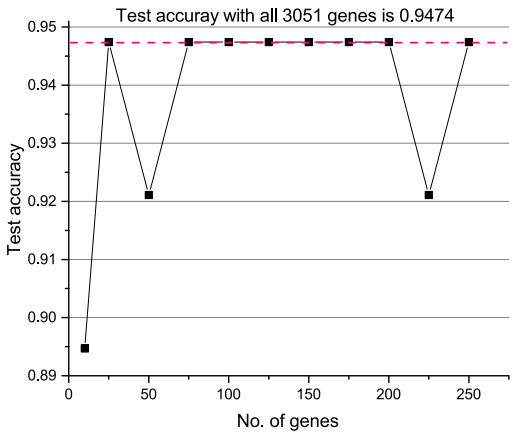
(e) Results on the $BR2$ data set.



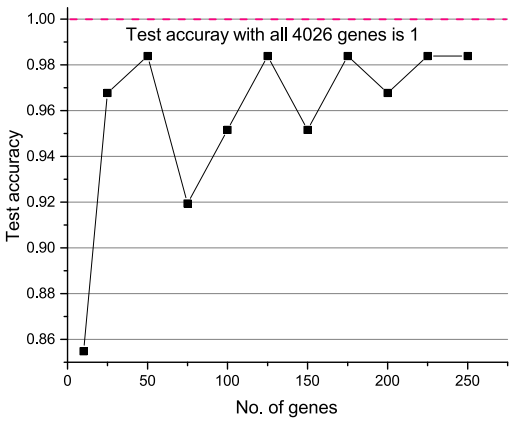
(f) Results on the $BR3$ data set.



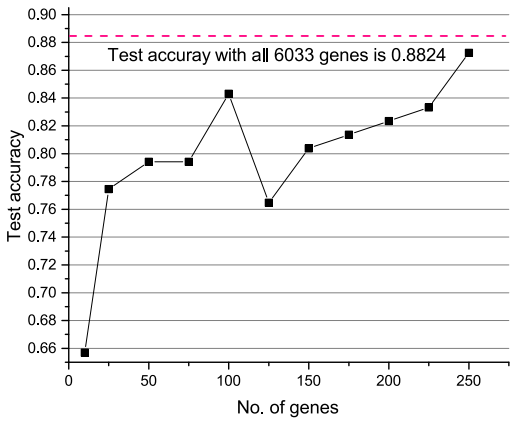
(g) Results on the COL data set.



(h) Results on the LEU data set.



(i) Results on the LYM data set.



(j) Results on the PRO data set.

Fig. 17. Test accuracy by the Random Forests algorithm versus the no. of selected genes on 10 gene expression data sets.

objects on sample clusters. The subspace weights can be automatically computed during the co-clustering process. The important genes can be identified from the subspace weights. A series of experiments were conducted and the results have demonstrated that SWCC is robust and scalable. We propose to use SWCC for gene clustering in which the sample labels are consumed to improve the clustering performance. The experimental results show that the

clustering results can be used for further biological investigations. We also propose a gene selection method to select important genes, which can be used to improve the classification performance of classification method.

In the future work, we will improve SWCC by introducing other techniques such as fuzzy clustering, and semi-supervised clustering. The further investigation of the use of SWCC is also our future work.

ACKNOWLEDGMENTS

This research was supported by the National Natural Science Foundation of China (NSFC) under Grant no. 61305059, 61473194 and 61502177, and Guangzhou Key Laboratory of Robotics and Intelligent Software under Grant No. 15180007.

REFERENCES

[1] Z. Du, Y. Wang, and Z. Ji, "Pk-means: A new algorithm for gene clustering," *Comput. Biol. Chemistry*, vol. 32, no. 4, pp. 243–247, 2008.
[2] H. Li, C. Li, J. Hu, and X. Fan, "A resampling based clustering algorithm for replicated gene expression data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 12, no. 6, pp. 1295–1303, Nov./Dec. 2015.
[3] 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.
[4] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inf. Process. Syst.*, vol. 2, pp. 849–856, 2002.
[5] J. Hartigan, "Direct clustering of a data matrix," *J. Amer. Statist. Assoc.*, vol. 67, pp. 123–129, 1972.
[6] I. Dhillon, S. Mallela, and D. Modha, "Information-theoretic co-clustering," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 89–98.
[7] Y. Song, S. Pan, S. Liu, F. Wei, M. X. Zhou, and W. Qian, "Constrained text co-clustering with supervised and unsupervised constraints," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1227–1239, Jun. 2013.
[8] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra, "Minimum sum-squared residue co-clustering of gene expression data," in *Proc. 4th SIAM Int. Conf. Data Mining*, 2004, pp. 328–335.
[9] S. Madeira and A. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 1, no. 1, pp. 24–45, Jan. 2004.
[10] R. Pensa and J. Boulicaut, "Constrained co-clustering of gene expression data," in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 25–36.
[11] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," in *Proc. 5th IEEE Int. Conf. Data Mining*, 2005, pp. 625–628.
[12] G. Govaert and M. Nadif, *Co-Clustering*. Hoboken, NJ, USA: Wiley, 2013.
[13] Y. Cheng and G. Church, "Biclustering of expression data," in *Proc. 8th Int. Conf. Intell. Syst. Molecular Biol.*, 2000, pp. 93–103.
[14] D. Duffy and J. Quiroz, "A permutation-based algorithm for block clustering," *J. Classification*, vol. 8, no. 1, pp. 65–91, 1991.
[15] R. Tryon, "Cluster Analysis; Correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality," *Edwards Brother, Inc., Litho Printers and Publishers*, 1939.
[16] R. Tryon and D. Bailey, *Cluster Analysis*. New York, NY, USA: McGraw-Hill, 1970.
[17] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2001, pp. 269–274.
[18] B. Gao, T. Liu, X. Zheng, Q. Cheng, and W. Ma, "Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2005, pp. 41–50.
[19] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. Modha, "A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation," *J. Mach. Learn. Res.*, vol. 8, pp. 1919–1986, 2007.
[20] L. Lazzeroni and A. Owen, "Plaid models for gene expression data," *Statistica Sinica*, vol. 12, no. 1, pp. 61–86, 2002.
[21] S. Busygin, G. Jacobsen, E. Krämer, and C. Ag, "Double conjugated clustering applied to leukemia microarray data," in *Proc. 2nd SIAM ICDM Workshop Clustering High Dimensional Data*, 2002.
[22] Y. Kluger, R. Basri, J. Chang, and M. Gerstein, "Spectral biclustering of microarray data: Coclustering genes and conditions," *Genome Res.*, vol. 13, no. 4, pp. 703–716, 2003.
[23] G. Govaert and M. Nadif, "An EM algorithm for the block mixture model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 643–647, Apr. 2005.
[24] M. Deodhar, G. Gupta, J. Ghosh, H. Cho, and I. Dhillon, "A scalable framework for discovering coherent co-clusters in noisy data," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 241–248.
[25] L. Zhang, C. Chen, J. Bu, Z. Chen, D. Cai, and J. Han, "Locally discriminative coclustering," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 1025–1035, Jun. 2012.
[26] W. Cheng, X. Zhang, F. Pan, and W. Wang, "HICC: An entropy splitting-based framework for hierarchical co-clustering," *Knowl. Inf. Syst.*, vol. 46, no. 2, pp. 343–367, 2016.
[27] Z. Huang, M. 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.
[28] L. Jing, M. 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.
[29] X. Chen, X. Xu, Y. Ye, and J. Z. Huang, "TW-k-means: Automated two-level variable weighting clustering algorithm for multi-view data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 932–944, Apr. 2013.
[30] X. Chen, Y. Ye, X. Xu, and J. Z. Huang, "A feature group weighting method for subspace clustering of high-dimensional data," *Pattern Recog.*, vol. 45, no. 1, pp. 434–446, 2012.
[31] W.-C. Tjhi and L. Chen, "Flexible fuzzy co-clustering with feature-cluster weighting," in *Proc. 9th Int. Conf. Control Autom. Robotics Vis.*, 2006, pp. 1–6.
[32] Y. Ye, X. Li, B. Wu, and Y. Li, "Feature weighting information-theoretic co-clustering for document clustering," in *Proc. 2nd Int. Conf. Comput. Sci. Appl.*, 2009, pp. 1–6.
[33] T. Sarazin, M. Lebbah, H. Azzag, and A. Chaibi, "Feature group weighting and topological biclustering," in *Proc. Neural Inf. Process.*, 2014, pp. 369–376.
[34] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1370–1386, Nov. 2004.
[35] A. Prelić, et al., "A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinf.*, vol. 22, no. 9, pp. 1122–1129, 2006.
[36] K. Eren, M. Deveci, O. Küçüktunç, and Ü. V. Çatalyürek, "A comparative analysis of biclustering algorithms for gene expression data," *Briefings Bioinf.*, vol. 14, no. 3, pp. 279–292, 2013.
[37] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, Cambridge, U.K.: Cambridge University Press, 2004.
[38] C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, Cambridge, U.K.: University Press Cambridge, 2008.
[39] A. Strehl and J. Ghosh, "Cluster ensembles-a knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2003.
[40] R. Díaz-Uriarte and S. A. De Andres, "Gene selection and classification of microarray data using random forest," *BMC Bioinf.*, vol. 7, no. 1, pp. 1–13, 2006.
[41] S. Zheng and W. Liu, "An experimental comparison of gene selection by lasso and dantzig selector for cancer classification," *Comput. Biol. Med.*, vol. 41, no. 11, pp. 1033–1040, 2011.
[42] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

Xiaojun Chen received the PhD degree from Harbin Institute of Technology in 2011. He is now an assistant professor at College of Computer Science and Software, Shenzhen University. His research interests include subspace clustering, topic model, massive data mining.

Joshua Zhexue Huang received the PhD degree from the Royal Institute of Technology in Sweden. He is now a professor at College of Computer Science and Software, Shenzhen University, and professor and chief scientist at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, and Honorary professor at Department of Mathematics, The University of Hong Kong. His research interests include data mining, machine learning and clustering algorithms.

Qingyao Wu received the BS degree in software engineering from the South China University of Technology and received the MS and PhD degrees in computer science from the Harbin Institute of Technology, China, in 2007, 2009, and 2013, respectively. He is currently an associate professor with the School of Software Engineering, South China University of Technology, China. He was a post-doctoral research fellow with the School of Computer Engineering, Nanyang Technological University, Singapore, from 2014 to 2015. His current research interests include machine learning, data mining and big data learning.

Min Yang is currently working toward the PhD degree in the Department of Computer Science, The University of Hong Kong, Hong Kong. She received the BS degree in software engineering from the Sichuan University, China, in 2012. Her current research interests include machine learning, data mining and natural language processing.