Pattern Recognition 76 (2018) 404–415

Contents lists available at ScienceDirect

Pattern Recognition

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

TWCC: Automated Two-way Subspace Weighting Partitional Co-Clustering

Xiaojun Chen[a,*], Min Yang[b], Joshua Zhexue Huang[a], Zhong Ming[a]

[a] College of Computer Science and Software, Shenzhen University, Shenzhen 518060, PR China
[b] Department of computer science, The University of Hong Kong, Pokfulam, Hong Kong

ARTICLE INFO

Article history:
Received 6 March 2017
Revised 22 September 2017
Accepted 18 October 2017
Available online 20 October 2017

Keywords:
Data mining
Co-clustering
Subspace clustering
Variable weighting

ABSTRACT

A two-way subspace weighting partitional co-clustering method TWCC is proposed. In this method, two types of subspace weights are introduced to simultaneously weight the data in two ways, i.e., columns on row clusters and rows on column clusters. An objective function that uses the two types of weights in the distance function to determine the co-clusters of data is defined, and an iterative TWCC co-clustering algorithm to optimize the objective function is proposed, in which the two types of subspace weights are automatically computed. A series of experiments on both synthetic and real-life data were conducted to investigate the properties of TWCC, compare the two-way clustering results of TWCC with those of eight co-clustering algorithms, and compare one-way clustering results of TWCC with those of six clustering algorithms. The results have shown that TWCC is robust and effective for large high-dimensional data.

1. Introduction

In the past decade, many clustering methods have been proposed for clustering analysis of high-dimensional data, including subspace clustering [1], co-clustering [1,2], SVD-based clustering [3] and spectral clustering [4,5]. 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[1] [6]. It has good performance on sparse and high-dimensional data, even if we are interested in clustering along one-way of the data [7]. Co-clustering has received wide attention in various applications such as simultaneous clustering of documents and words in text mining [8,9], genes and experimental conditions in bioinformatics [10,11], users and movies in recommendation systems [12]. However, as more and more variables are included in data and some variables are noise and irrelevant variables, such data present a big challenge to co-clustering methods, which is the research problem of this paper.

Soft subspace weighting clustering has been an important research topic in the past decade [1]. It assigns weights to individual variables and uses the weights to identify important variables from which the subspace structures of clusters can be discovered. Experiments have shown that subspace weighting clustering methods have good performance on high-dimensional data [13–15]. Weighting technique has been introduced into co-clustering and many weighting co-clustering methods have been proposed [16–19]. These methods only compute weights for co-clusters. Since the number of co-clusters is much smaller than the number of variables, weighting individual variables will be preferable to weighting co-clusters. However, existing methods only weight one way variables and will result in biased co-clustering results.

In this paper, we propose a two-way subspace weighting partitional co-clustering method, called TWCC, for co-clustering high dimensional data with noise and irrelevant variables. In this method, we introduce two types of subspace weights, one for columns on row clusters and the other one for rows on column clusters. The two types of subspace weights are combined together to simultaneously weight the data in two ways and the contribution of each individual row/column to the co-clusters can be simultaneously identified from the weights. We present an objective function that uses the two types of subspace weights in the distance function to determine the co-clusters of data, and an iterative co-clustering algorithm to optimize the model.

Three sets of experiments were conducted on both synthetic and real-life data to study the TWCC algorithm. The first set was used to study the properties of TWCC. The second set was used to compare the clustering performance of TWCC with those of eight co-clustering algorithms on synthetic data. Finally, we selected six real-life data sets and compared the clustering perfor-

* Corresponding author.
E-mail addresses: xjchen@szu.edu.cn (X. Chen), myang@cs.hku.hk (M. Yang), zx.huang@szu.edu.cn (J. Zhexue Huang), mingz@szu.edu.cn (Z. Ming).

[1] Since we can consider the data as a matrix, we can refer to samples as rows and features as columns which is also easy to understand.

https://doi.org/10.1016/j.patcog.2017.10.026
0031-3203/

mance of TWCC with those of six clustering algorithms, including both co-clustering and one-way clustering algorithms.

The rest of this paper is organized as follows. Section 2 gives a brief review of related work on partitional co-clustering and subspace clustering. We state the problem of two-way subspace weighting partitional co-clustering in Section 3. Section 4 presents the objective function for two-way subspace weighting partitional co-clustering and the TWCC algorithm to optimize the model. Section 5 investigates the properties of the TWCC algorithm. The experimental results on both synthetic and real-life data are reported and discussed in Sections 6 and 7, respectively. Conclusions and future work are given in Section 8.

2. Related work

Co-clustering [20], also called two-mode clustering [21], bi-clustering [22], direct clustering [6], block clustering [23], is a process of simultaneously clustering rows and columns of a data matrix. The concept itself can be traced back to1960's and1970's [24,25], although it had been rarely used or even studied. Until recently, co-clustering has received wide attention in many areas such as text mining [8,9], bioinformatics [11,19] and recommendation systems [12]. Several co-clustering models have been formulated, including hierarchical co-clustering [26], sequential bi-clustering [27], spectral co-clustering [28] and partitional co-clustering [29]. In this paper, we only focus on the partitional co-clustering.

Partitional co-clustering, first introduced in 1972 [6], 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, where K is the number of row clusters and L is the number of column clusters. Some partitional co-clustering algorithms have been proposed to follow the partitional process. Lazzeroni et al. [30] 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. [31] proposed double conjugated clustering (DCC), a node-driven partitional co-clustering method, for microarray data analysis. Kluger et al. [32] 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 expression pattern across genes or conditions. Govaert et al. [33] 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. [7] 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. [34] 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. [29] 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.

Lijun et al. proposed a Locally Discriminative Coclustering (LDCC) to explore the relationship between samples and features as well as the intersample and interfeature relationships [35]. In this method, the sample-feature relationship is modeled by a bipartite graph between samples and features, and they apply local linear regression to discover the intrinsic discriminative structures of both sample space and feature space.

Liu et al. proposed a co-clustering algorithm to discover cancer subtypes from gene expression data [19]. In their method, they used a modified PageRank algorithm to assign weights to genes and the semi-Nonnegative Matrix Tri-Factorization method to simultaneously separate samples into subtypes and group genes into functionally relevant subclasses. However, their method is specially designed for gene data and cannot work for other types of data.

In the past decade, soft subspace clustering has been an important research topic in cluster analysis [13–15,36]. Such methods assign weights to variables and reduce the affection of noise features by assigning them with low weights, thus being useful for high-dimensional data [1,15]. Weighting technique was introduced into co-clustering to remove the affection of noise rows or columns. Tjhi et al. [16] 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. [17] 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. [18] 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.

However, current weighting co-clustering methods only compute weights for co-clusters. Computing two-way variable weights for co-clustering has rarely been considered.

3. Problem statement

Let $\boldsymbol{X} = [x_{ij}]_{N\times M}$ 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 $U = [u_{ig}]_{N\times K}$ and $V = [v_{jh}]_{M\times L}$, in which $u_{ig} = 1$ indicates that the ith row object is assigned to the gth row cluster and $v_{jh} = 1$ indicates that the jth column object is assigned to the hth column cluster. The Bregman Block Average Co-clustering (BBAC) defines the following objective function [29]:

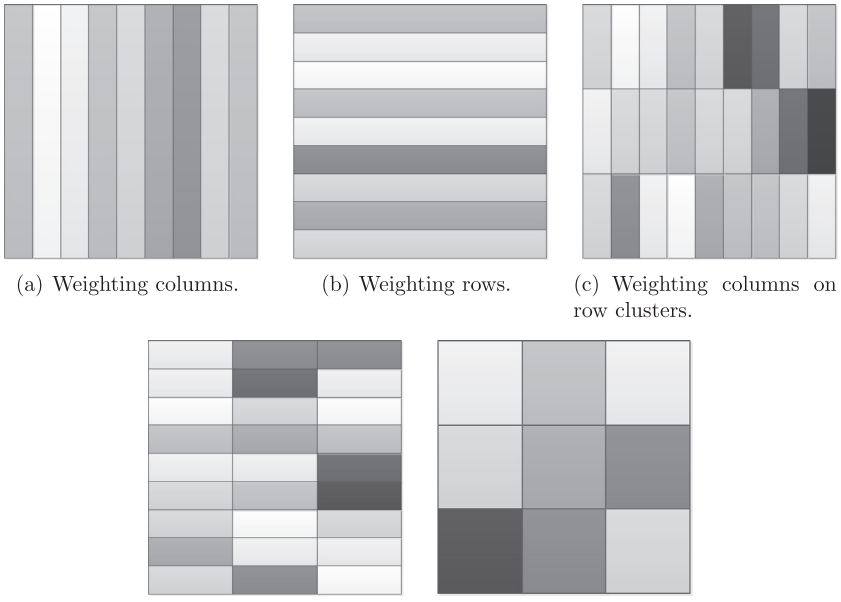
$$P(U, V, Z) = \sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M} u_{ig}v_{jh}d(x_{ij}, z_{gh}) \tag{1}$$

subject to

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

where $d(x_{ij}, z_{gh})$ is a Bregman divergence. By selecting different Bregman divergences and co-clustering basis, different variants can be derived, e.g., ITCC and MSSRCC.

In real-life data, $\boldsymbol{X}$ often has very large N and M, and contains a lot of noisy values. To deal with such data, we define a weight matrix on each datum, $W = [w_{ij}]_{N\times M}$ where w_{ij} is the weight for the datum in the ith row and jth column. We use heat map to show the weight matrix W, where the horizontal blocks represent rows and the vertical blocks represent columns. Referring to



(a) Weighting columns. (b) Weighting rows. (c) Weighting columns on row clusters.

(d) Weighting rows on column clusters. (e) Weighting co-clusters.

Fig. 1. Weight matrices of five weighting methods for partitional co-clustering.

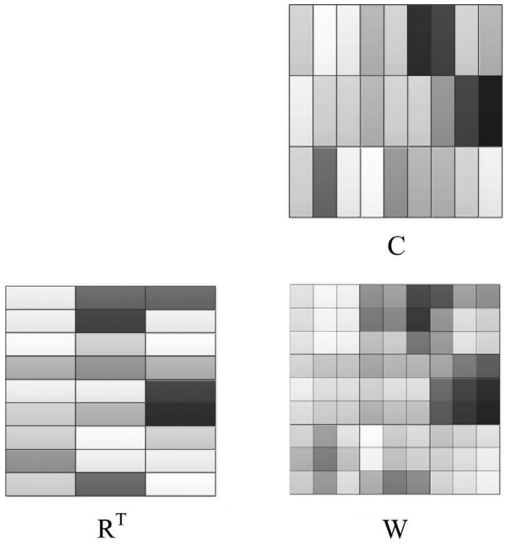


Fig. 2. Weight matrices of two-way subspace weighting for partitional co-clustering.

the current subspace weighting methods, we can define five different weighting methods for partitional co-clustering, whose weight matrices are shown in Fig. 1. The gray levels in these figures indicate different weight values, where the darker color means the higher weight. Same colors are assigned to the cells if the corresponding elements in the weight matrix have the same values. The five methods in Fig. 1(a)–(e) compute the same weights for each column, each row, each column in each row cluster, each row in each column cluster, and each co-clusters, respectively. However, the first four methods only compute weights for one-way, the contributions of individual rows/columns in co-clusters cannot be revealed in the last method.

To simultaneously weight the co-clusters and the individual variables in co-clusters, we define a two-way subspace weighting method as shown in Fig. 2. In this method, we define two types of subspace weights: $R = [r_{hi}]_{L\times N}$ for rows on column clusters, and $C = [c_{gj}]_{K\times M}$ for columns on row clusters. By combining the two types of weights, we obtain a two-way weight matrix $W = [w_{ij}]_{N\times M} = (UC) \circ (VR)^T$, which is the Hadamard product of two $N \times M$ weight matrices UC and $(VR)^T$.

From W, we cannot only capture the inner structures of co-clusters by the weights on rows and columns in these co-clusters, but also capture the structures of these co-clusters by summarizing the weights of rows and columns into the weights of these co-clusters. In the following section, we will propose a new partitional co-clustering algorithm with the two-way subspace weighting method.

4. TWCC clustering algorithm

4.1. Objective function

To cluster a data matrix $\boldsymbol{X}$ into K row clusters and L column clusters, we introduce two types of subspace weights into the distance function of BBAC (Bregman Block Average Co-clustering) with the Squared Euclidean distance [29], and formulate the problem as an iterative clustering process to minimize the following objective function:

$$P(U,V,Z,R,C) = \frac{1}{MN}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M} u_{ig}v_{jh}r_{hi}c_{gj}d(x_{ij},z_{gh}) + \frac{\lambda}{N}\sum_{h=1}^{L}\sum_{i=1}^{N} r_{hi}\log r_{hi} + \frac{\eta}{M}\sum_{g=1}^{K}\sum_{j=1}^{M} c_{gj}\log c_{gj} \quad (3)$$

subject to

$$\begin{cases} \sum_{g=1}^{K} u_{ig} = 1, & u_{ig} \in \{0,1\}, \quad 1 \le i \le N \\ \sum_{h=1}^{L} v_{jh} = 1, & v_{jh} \in \{0,1\}, \quad 1 \le j \le M \\ \sum_{i=1}^{N} r_{hi} = 1, & 0 < r_{hi} < 1, \quad 1 \le h \le L \\ \sum_{j=1}^{M} c_{gj} = 1, & 0 < c_{gj} < 1, \quad 1 \le g \le K \end{cases} \quad (4)$$

where

- $U = [u_{ij}]_{N\times K}$ is a binary matrix in which $u_{ig} = 1$ indicates that the ith row is assigned to the gth row cluster;

- $V = [v_{jh}]_{M\times L}$ is a binary matrix in which $v_{jh} = 1$ indicates that the jth column is assigned to the hth column cluster;
- $Z = [z_{gh}]_{K\times L}$ is the centers of $K \times L$ co-clusters;
- $R = [r_{hi}]_{L\times N}$ is a weight matrix where $r_{hi} > 0$ is the weight for the ith row in the hth column cluster;
- $C = [c_{gj}]_{K\times M}$ is a weight matrix where $c_{gj} > 0$ is the weight for the jth column in the gth row cluster;
- λ and η are two positive parameters;
- $d(x_{ij}, z_{gh})$ is a distance defined as:

$$d(x_{ij}, z_{gh}) = (x_{ij} - z_{gh})^2 \tag{5}$$

The first term in (3) is the weighted sum of within-cluster dispersion, the second and third terms are two penalty terms of negative weight entropies. By combining the three terms together, we can simultaneously minimize the within-cluster dispersion and maximize weight entropies to stimulate more rows and columns to contribute to the identification of co-clusters. In this way, we can avoid the problem of identifying co-clusters by few rows or columns in sparse data.

4.2. TWCC co-clustering algorithm

We can minimize (3) by iteratively solving the following five minimization problems:

1. Problem P_1: $\arg\min_U P(U, \hat{V}, \hat{Z}, \hat{R}, \hat{C})$;
2. Problem P_2: $\arg\min_V P(\hat{U}, V, \hat{Z}, \hat{R}, \hat{C})$;
3. Problem P_3: $\arg\min_Z P(\hat{U}, \hat{V}, Z, \hat{R}, \hat{C})$;
4. Problem P_4: $\arg\min_R P(\hat{U}, \hat{V}, \hat{Z}, R, \hat{C})$;
5. Problem P_5: $\arg\min_C P(\hat{U}, \hat{V}, \hat{Z}, \hat{R}, C)$.

We can verify that P_1, P_2 and P_3 are solved by

$$\begin{cases} u^*_{ig} = 1 & \text{if } P_{(g)} \le P_{(s)} \text{ for } 1 \le s \le K \text{ where} \\ P_{(s)} = \sum_{h=1}^{L}\sum_{j=1}^{M} \hat{v}_{jh}\hat{r}_{hi}\hat{c}_{sj} d(x_{ij}, \hat{z}_{sh}) \\ u^*_{is} = 0 & \text{for } s \ne g \end{cases} \tag{6}$$

$$\begin{cases} v_{jh} = 1 & \text{if } P_{(h)} \le P_{(t)} \text{ for } 1 \le t \le L \text{ where} \\ P_{(t)} = \sum_{g=1}^{K}\sum_{i=1}^{N} \hat{u}_{ig}\hat{r}_{ti}\hat{c}_{gj} d(x_{ij}, \hat{z}_{gt}) \\ v_{jt} = 0 & \text{for } t \ne h \end{cases} \tag{7}$$

$$z_{gh} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} \hat{u}_{ig}\hat{v}_{jh}\hat{r}_{hi}\hat{c}_{gj}x_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{M} \hat{u}_{ig}\hat{v}_{jh}\hat{r}_{hi}\hat{c}_{gj}} \tag{8}$$

Theorem 1. *P_4 is solved by:*

$$r^*_{hi} = \frac{\exp\{\frac{-F_{hi}}{\lambda}\}}{\sum_{i'=1}^{N}\exp\{\frac{-F_{hi'}}{\lambda}\}} \tag{9}$$

where

$$F_{hi} = \frac{1}{M}\sum_{g=1}^{K}\sum_{j=1}^{M} \hat{u}_{ig}\hat{v}_{jh}\hat{c}_{gj}\, d(x_{ij}, \hat{z}_{gh}) \tag{10}$$

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

$$L_{[r_{11},\ldots,r_{LN}]} = \frac{1}{N}\sum_{h=1}^{L}\left[\sum_{i=1}^{N} F_{hi}r_{hi} + \lambda\sum_{i=1}^{N} r_{hi}\log r_{hi} + \gamma_h\left(\sum_{i=1}^{N} r_{hi} - 1\right)\right]$$

where F_{hi} is a constant calculated by (10), and $[\gamma_1, \ldots, \gamma_L]$ are L Lagrange multipliers.

By setting the gradients of $L_{[r_{11},\ldots,r_{LN}]}$ with respect to both γ_h and r_{hi} to zero, we obtain the optima of r_{hi} as shown in (9). The second order derivative of $L_{[r_{11},\ldots,r_{LN}]}$ is $\frac{L_{[r_{11},\ldots,r_{LN}]}}{\partial r_{hi}} = \frac{\lambda}{r_{hi}}$. Since $r_{hi} > 0$ and $\lambda > 0$, according to the second derivative test, r^*_{hi} is the minima of $P(\hat{U}, \hat{V}, \hat{Z}, R, \hat{C})$. □

Theorem 2. *P_5 is solved by*

$$c^*_{gj} = \frac{\exp\{\frac{-E_{gj}}{\eta}\}}{\sum_{j'=1}^{M}\exp\{\frac{-E_{gj'}}{\eta}\}} \tag{11}$$

where

$$E_{gj} = \frac{1}{N}\sum_{h=1}^{L}\sum_{i=1}^{N} \hat{u}_{ig}\hat{v}_{jh}\hat{r}_{hi}\, d(x_{ij}, \hat{z}_{gh}) \tag{12}$$

Theorem 2 can be proved in a similar way as Theorem 1.

The TWCC algorithm that minimizes the objective function (3) is given as Algorithm 1. In each iteration, U, V, R and C are iteratively updated until convergence. The final cluster assignments can be obtained from both U and V. According to (9) and (11), R is inversely proportional to F and C is inversely proportional to E. We also note that R depends on λ and C is depends on η. The impact of λ and η on R and C will be studied in Section 5.

Algorithm 1 TWCC.

Input: X, K, L, λ, η;
Output: U, V, Z, R, C;
Init: Let $r_{hi} = \frac{1}{L}$, $c_{gj} = \frac{1}{K}$ and start with an arbitrary partitional co-clustering.
$t := 0$
repeat
  Update U^{t+1} by (6);
  Update V^{t+1} by (7);
  Update Z^{t+1} by (8);
  Update R^{t+1} by (9) and (10);
  Update C^{t+1} by (11) and (12);
  $t := t + 1$
until (3) obtains its local minimum value;

4.3. Convergency and complexity analysis

The objective function (3) can be minimized by alternatively solving the five minimization problems P_1, ... ,P_5, each of which is a convex problem with respect to one variable. Therefore, the sequence of $P(., ., .)$ generated by Algorithm 1 is strictly decreasing. Finally, the algorithm will converge to a local solution.

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

5. Experiments on properties of TWCC

In this section, we define five weighting partitional co-clustering methods to study the properties of TWCC. We generated a typical synthetic data set and a series of experiments on the data set to demonstrate the impacts of the parameters on the weights and clustering performance in TWCC. We also compared the weights of TWCC with those of other five weighting partitional co-clustering methods.

Table 1
Five weighting partitional co-clustering algorithms.

Name	Objective functions	Weight formulae
CWCC	$J(U,V,Z,C)=\frac{1}{MN}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M}u_{ig}v_{jh}c_j d(x_{ij},z_{gh})$ $+\frac{\eta}{M}\sum_{j=1}^{M}c_j\log c_j$	$c_j=\frac{\exp\left\{\frac{-D_j}{\eta}\right\}}{\sum_{j'=1}^{M}\exp\left\{\frac{-D_{j'}}{\eta}\right\}}$ where $D_j=\frac{1}{N}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\hat{u}_{ig}\hat{v}_{jh}d(x_{ij},\hat{z}_{gh})$
RWCC	$J(U,V,Z,R)=\frac{1}{MN}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M}u_{ig}v_{jh}r_i d(x_{ij},z_{gh})$ $+\frac{\lambda}{N}\sum_{i=1}^{N}r_i\log r_i$	$r_i=\frac{\exp\left\{\frac{-D_i}{\lambda}\right\}}{\sum_{i'=1}^{N}\exp\left\{\frac{-D_{i'}}{\lambda}\right\}}$ where $D_i=\frac{1}{M}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{j=1}^{M}\hat{u}_{ig}\hat{v}_{jh}d(x_{ij},\hat{z}_{gh})$
CSWCC	$J(U,V,Z,C)=\frac{1}{MN}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M}u_{ig}v_{jh}c_{gj}d(x_{ij},z_{gh})$ $+\frac{\eta}{M}\sum_{g=1}^{K}\sum_{j=1}^{M}c_{gj}\log c_{gj}$	$c_{gj}=\frac{\exp\left\{\frac{-D_{gj}}{\eta}\right\}}{\sum_{j'=1}^{M}\exp\left\{\frac{-D_{gj'}}{\eta}\right\}}$ where $D_{gj}=\frac{1}{N}\sum_{h=1}^{L}\sum_{i=1}^{N}\hat{u}_{ig}\hat{v}_{jh}d(x_{ij},\hat{z}_{gh})$
RSWCC	$J(U,V,Z,R)=\frac{1}{MN}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M}u_{ig}v_{jh}r_{hi}d(x_{ij},z_{gh})$ $+\frac{\lambda}{N}\sum_{h=1}^{L}\sum_{i=1}^{N}r_{hi}\log r_{hi}$	$r_{hi}=\frac{\exp\left\{\frac{-D_{hi}}{\lambda}\right\}}{\sum_{i'=1}^{N}\exp\left\{\frac{-D_{hi'}}{\lambda}\right\}}$ where $D_{hi}=\frac{1}{M}\sum_{g=1}^{K}\sum_{j=1}^{M}\hat{u}_{ig}\hat{v}_{jh}d(x_{ij},\hat{z}_{gh})$
BWCC	$J(U,V,Z,W)=\frac{1}{MN}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M}u_{ig}v_{jh}w_{gh}d(x_{ij},z_{gh})$ $+\lambda\sum_{g=1}^{K}\sum_{h=1}^{L}w_{gh}\log w_{gh}$	$w_{gh}=\frac{\exp\left\{\frac{-D_{gh}}{\lambda}\right\}}{\sum_{g'=1}^{K}\sum_{h'=1}^{L}\exp\left\{\frac{-D_{g'h'}}{\lambda}\right\}}$ where $D_{gh}=\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\hat{u}_{ig}\hat{v}_{jh}d(x_{ij},\hat{z}_{gh})$

5.1. Five weighting partitional co-clustering methods

To systematically study the weighting methods in partitional co-clustering, we define five weighting partitional co-clustering algorithms from the five weighting methods as shown in Fig. 1:

- **CWCC** (Column Weighting partitional Co-clustering): weights individual columns, as shown in Fig. 1(a).
- **RWCC** (Row Weighting partitional Co-clustering): weights individual rows, as shown in Fig. 1(b).
- **CSWCC** (Column Subspace Weighting partitional Co-clustering): weights individual columns on each row cluster, as shown in Fig. 1(c).
- **RSWCC** (Row Subspace Weighting partitional Co-clustering): weights individual rows on each column cluster, as shown in Fig. 1(d).
- **BWCC** (Block Weighting partitional Co-clustering): weights co-clusters, as shown in Fig. 1(e).

The objective functions and formulae to compute the weights in these algorithms are given in Table 1. Although the weight formulae derived from these objective functions are different, the weights are all inversely proportional to the sums of the within-cluster variances of the data. We can optimize the five objective functions with a iterative co-clustering algorithm similar to Algorithm 1.

5.2. Experiment setup

Fig. 3 shows a typical synthetic data set D_1 with 100 rows and 120 columns, where the darker color indicates lower 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 properties of the TWCC algorithm and compared the weights of TWCC with those of the five partitional weighting co-clustering methods listed in Table 1. We set $K=L=3$ and chose a set of 20 real values for two parameters λ and η in six partitional co-clustering methods, which range from $1.2^0\times10^{-4}$ to $1.2^{19}\times10^{-4}$, in 20 equal steps on an exponential scale. For each combination of λ and η in six partitional co-clustering methods, we randomly generated 100 initial cluster centers and ran each algorithm to generate 100 results. Finally, we produced 240,000 co-clustering results for the following analysis. These clustering results were evaluated with the normalized mutual information (*NMI*) which is defined in [37].

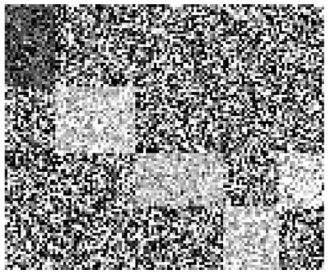


Fig. 3. Plot of a typical synthetic data set D_1.

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

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:

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

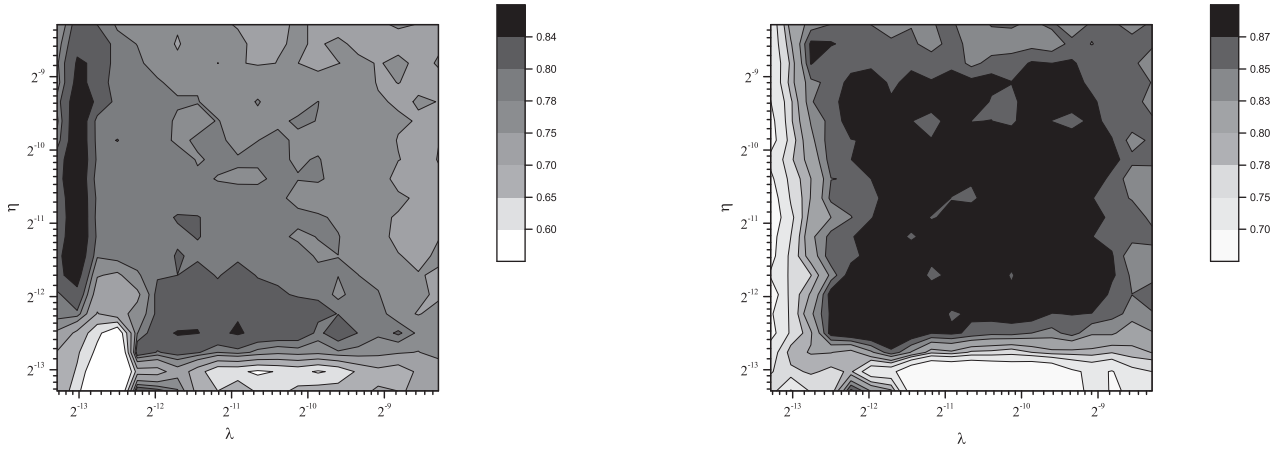
and H is the entropy defined as:

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

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 in cluster ω_k and class c_j.

5.3. Impacts of λ, η on co-clustering performance

From each clustering result of TWCC, 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 combination of λ and η. The results



(a) NMI_C versus λ and η. (b) NMI_R versus λ and η.

Fig. 4. The log-log contour maps of NMI_C and NMI_R over λ and η from the results of TWCC on D_1.

Table 2
The best NMI_R and NMI_C values of six algorithms.

Algorithms	NMI_R	NMI_C
CWCC	0.29	0.65
RWCC	0.74	0.29
CSWCC	0.78	0.78
RSWCC	0.89	0.65
BWCC	0.8	0.63
TWCC	0.94	0.91

are plotted in Fig. 4, in which the darker color indicates the higher value and the horizontal and vertical axes are both in logarithmic scale.

From Fig. 4(a) and (b), we can see that both NMI_C and NMI_R are low in the left lower corner, but are high in the right upper corner. This may be because that 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. The ratio of the region with $NMI \geq 0.85$ is about 50% in Fig. 4(a) and 65% in Fig. 4(b), and the intersection is about 42% of the whole area. Therefore, it has a high chance to produce good clustering results. Moreover, cluster ensembles can be used to combine multiple clustering results to produce good clustering results [38].

From Fig. 4, we can find that NMI_C and NMI_R are both sensitive to η and λ. In the previous subsection, we have found that R is mainly affected by λ and C is mainly affected by η. However, each iteration in TWCC involves five steps and these steps depend on each other. For example, R depends on C (see Eq. (9)) and C depends on R (see Eq. (11)). Therefore, both row and column clustering results of TWCC are sensitive to λ and η.

5.4. Weight comparisons

We computed NMI_R and NMI_C from each clustering result and the highest values of each method are listed in Table 2. We can see that TWCC and RSWCC achieved the highest NMI values. Three subspace weighting methods CSWCC, RSWCC and TWCC had better results than the other three methods. We also find that RWCC had the worst NMI_C result and CWCC had the worst NMI_R result, which indicates that one-way weighting may only improve one-way co-clustering performance.

For each co-clustering result in Table 2, we computed a weight matrix $W = [w_{ij}]_{m\times n}$ where w_{ij} is the weight assigned to the ith row and jth column. The six weight matrices were plotted in Fig. 5. The darker color indicates higher weight. We can see that the weights of CWCC are concentrated on only one column, while the weights of RWCC are concentrated on only one row. The weights of BWCC are evenly distributed so the weight value in each element is very small. Compared with the plot of D_1, we can see that the weights of the above three methods cannot reveal the inner subspace structure in the data. This can explain why the three clustering results were not good. CSWCC only reveals column subspace structure, which helps the identification of row clusters. This is why CSWCC produced better NMI_R results than NMI_C results. Similar phenomena can be observed in RSWCC. From the weights in TWCC, we can find the subspace structure from two levels. The three co-clusters in the diagonal line are most predominant. If we further take insight into co-clusters, we can see different rows/columns have different contributions to co-clusters. Therefore, TWCC can better reveal the subspace structure of the co-clusters in data.

5.5. Convergence study

We show the convergence curves of the objective function value in Fig. 6. From this figure, we can see that the objective function value of TWCC drops very fast and it converges in 10 iterations. Therefore, TWCC can quickly converge to a local minimization.

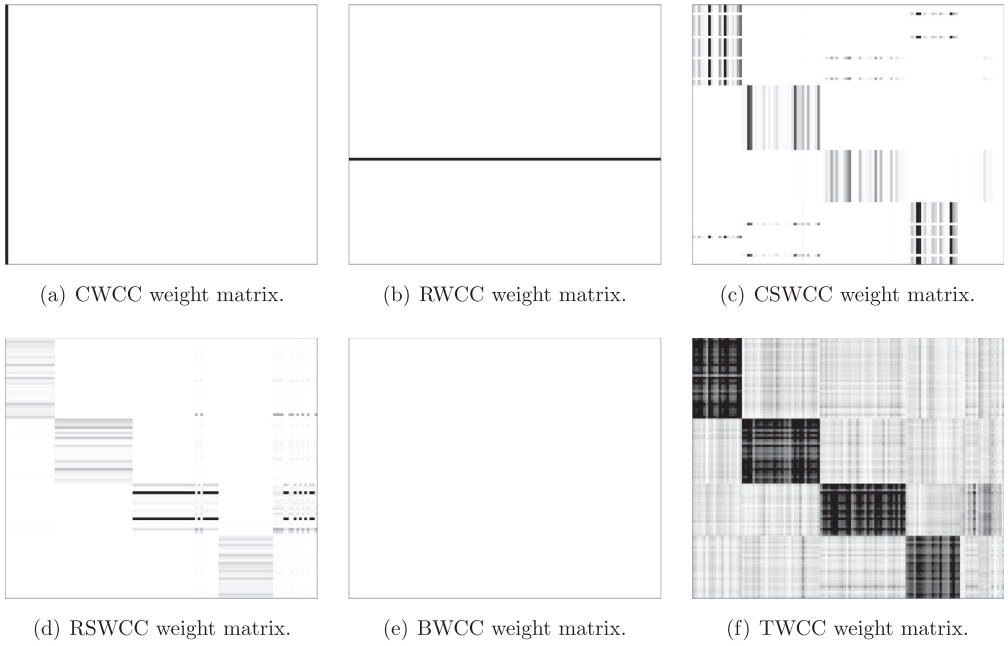
6. Co-clustering performance comparisons on synthetic data sets

We conducted experiments on synthetic data sets to investigate the robustness and scalability of TWCC. Details of the results are presented in this section.

6.1. Robustness comparison

To evaluate the robustness of different partitional co-clustering algorithms, Lomet et al. [39] designed a simulation method and generated a series of synthetic data sets for testing the performance of the co-clustering algorithms against noise.[2] We chose nine typical synthetic data sets from these data sets as benchmark data. Table 3 lists the characteristics of these data sets. All of them were generated from the latent-block generative model [40] with

[2] http://www.hds.utc.fr/coclustering.



(a) CWCC weight matrix. (b) RWCC weight matrix. (c) CSWCC weight matrix.

(d) RSWCC weight matrix. (e) BWCC weight matrix. (f) TWCC weight matrix.

Fig. 5. Plots of weight matrices of six co-clustering algorithms.

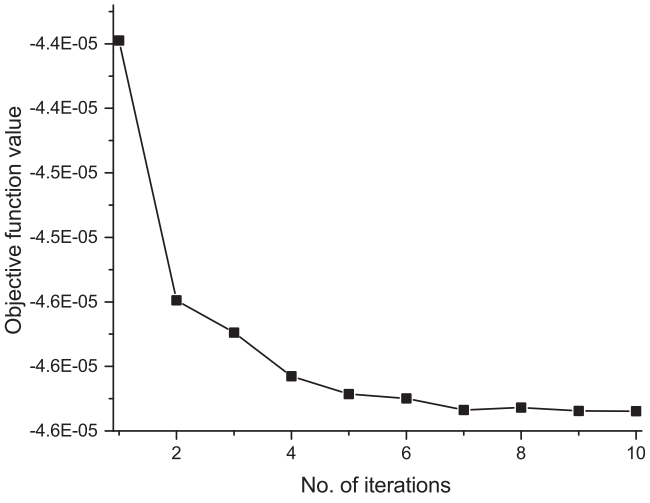


Fig. 6. Objective function values of TWCC on D_1.

Table 3
Characteristics of nine synthetic benchmark data sets.

Name	Size	Clusters	Error rate
D_{21}	100×100	3×3	0.05
D_{22}			0.12
D_{23}			0.2
D_{31}	200×200	3×3	0.05
D_{32}			0.12
D_{33}			0.2
D_{41}	500×500	3×3	0.05
D_{42}			0.12
D_{43}			0.2

normally distributed entries, and an error rate was computed for each data set using the method in [39]. Intuitively, the higher the error rate of a data set, the more difficult to cluster the data set.

The co-clustering performance of TWCC was compared with the five weighting co-clustering algorithms in Table 1 and the following three co-clustering methods:

- **BBAC** (Bregman Block Average co-clustering) with the Squared Euclidean distance [29].
- **ITCC** (Information-Theoretic Co-Clustering) [7].
- **FFCFW** (Fuzzy weighting co-clustering) [16].

We conducted experiments on the above nine synthetic data sets plus data set D_1, in a similar way as those in Section 5.2. The same sequence $\{1.2^0 \times 10^{-4}, \ldots, 1.2^{19} \times 10^{-4}\}$ was chosen for three parameters in FFCFW. For each combination of three parameters in FFCFW, we randomly generated 100 sets of object and feature memberships to produce 100 clustering results.

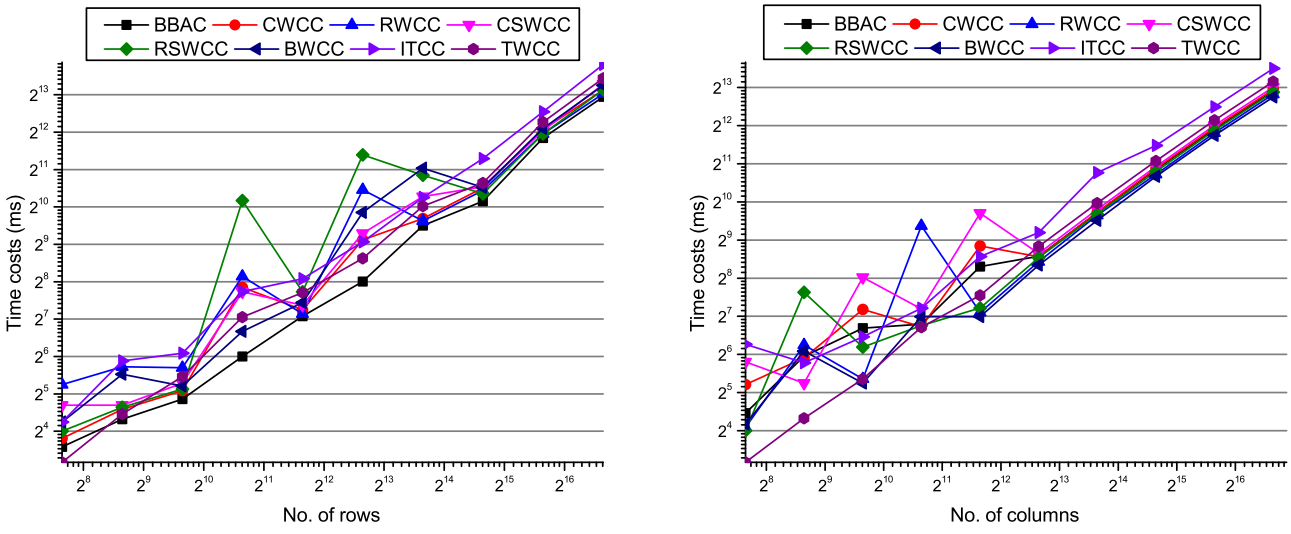
For each clustering result, we computed NMI_R and NMI_C. For each algorithm on each data set, the average and maximum NMI_R and NMI_C values were computed and listed in Table 4, where the value ahead of bracket is the average value and the value in bracket is the maximum value. For each data set, the best average and maximum NMI_R and NMI_C are rendered in bold separately. A "$*$" is placed close to the brackets if the difference of average value between the corresponding algorithm (excluding TWCC) and TWCC is significant by t-test, i.e., the $p$value of t-test is less than 0.05. The best average and maximum NMI_R/NMI_C values for each evaluation index on each data set are rendered in bold.

From the results in Table 4, we can see that BWCC achieved none of co-clustering results with specific number of co-clusters. Comparatively, CWCC, RWCC, CSWCC and RSWCC achieved worse results than non-weighting BBAC. This may be because that the one-way weighting technique is instable and cannot improve the clustering performance of co-clustering. We also noticed that TWCC significantly outperformed almost all other algorithms in all evaluation indices. FFCFW produced better clustering results than other seven co-clustering algorithms except TWCC.

We also noticed that, as the size and error rate of the data increased, both average and maximum NMI values of seven co-clustering algorithms (excluding FFCFW and TWCC) soon dropped to 0, indicating that the result can be considered as random assignments. However, the average NMI values of TWCC dropped much slower than the other algorithms and the maximum NMI

Table 4
Comparisons of co-clustering results by nine co-clustering algorithms on ten synthetic data.

Data	evaluation	BBAC	CWCC	RWCC	CSWCC	RSWCC	BWCC	ITCC	FFCFW	TWCC
D_1	NMI_R	0.78(0.96)*	0.50(0.70)*	0.17(0.25)*	0.40(0.87)*	0.39(1.00)*	–(–)	0.53(0.88)*	0.40(0.69)*	**0.84**(**1.00**)
	NMI_C	0.78(0.96)*	0.50(0.70)*	0.17(0.25)*	0.40(0.87)*	0.39(1.00)*	–(–)	0.53(0.88)*	0.40(0.69)*	**0.84**(**1.00**)
D_{21}	NMI_R	0.49(0.83)*	0.12(0.25)*	0.03(0.10)*	0.09(0.28)*	0.05(0.34)*	–(–)	0.00(0.00)*	0.42(0.56)*	**0.59**(**0.96**)
	NMI_C	0.49(0.83)*	0.12(0.25)*	0.03(0.10)*	0.09(0.28)*	0.05(0.34)*	–(–)	0.00(0.00)*	0.42(0.56)*	**0.59**(**0.96**)
D_{22}	NMI_R	0.16(0.52)*	0.06(0.12)*	0.04(0.10)*	0.05(0.19)*	0.02(0.11)*	–(–)	0.00(0.00)*	**0.39**(0.47)	0.36(**0.67**)
	NMI_C	0.16(0.52)*	0.06(0.12)*	0.04(0.10)*	0.05(0.19)*	0.02(0.11)*	–(–)	0.00(0.00)*	**0.39**(0.47)	0.36(**0.67**)
D_{23}	NMI_R	0.14(0.44)*	0.07(0.13)*	0.03(0.07)*	0.07(0.22)*	0.03(0.12)*	–(–)	0.00(0.00)*	–(–)	**0.41**(**0.74**)
	NMI_C	0.14(0.44)*	0.07(0.13)*	0.03(0.07)*	0.07(0.22)*	0.03(0.12)*	–(–)	0.00(0.00)*	–(–)	**0.41**(**0.74**)
D_{31}	NMI_R	0.27(0.78)*	0.04(0.12)*	0.02(0.04)*	0.06(0.17)*	0.02(0.11)*	–(–)	0.00(0.00)*	0.34(0.38)*	**0.58**(**0.85**)
	NMI_C	0.27(0.78)*	0.04(0.12)*	0.02(0.04)*	0.06(0.17)*	0.02(0.11)*	–(–)	0.00(0.00)*	0.34(0.38)*	**0.58**(**0.85**)
D_{32}	NMI_R	0.10(0.29)*	0.05(0.14)*	0.01(0.02)*	0.03(0.15)*	0.01(0.08)*	–(–)	0.00(0.00)*	0.34(0.38)*	**0.50**(**0.72**)
	NMI_C	0.10(0.29)*	0.05(0.14)*	0.01(0.02)*	0.03(0.15)*	0.01(0.08)*	–(–)	0.00(0.00)*	0.34(0.38)*	**0.50**(**0.72**)
D_{33}	NMI_R	0.06(0.20)*	0.03(0.09)*	0.00(0.02)*	0.03(0.13)*	0.02(0.07)*	–(–)	0.00(0.00)*	0.35(0.35)	**0.42**(**0.74**)
	NMI_C	0.06(0.20)*	0.03(0.09)*	0.00(0.02)*	0.03(0.13)*	0.02(0.07)*	–(–)	0.00(0.00)*	0.35(0.35)	**0.42**(**0.74**)
D_{41}	NMI_R	0.10(0.45)*	0.06(0.08)*	0.01(0.01)*	0.03(0.09)*	0.01(0.08)*	–(–)	0.00(0.00)*	0.34(0.35)*	**0.70**(**0.75**)
	NMI_C	0.10(0.45)*	0.06(0.08)*	0.01(0.01)*	0.03(0.09)*	0.01(0.08)*	–(–)	0.00(0.00)*	0.34(0.35)*	**0.70**(**0.75**)
D_{42}	NMI_R	0.02(0.08)*	0.02(0.04)*	0.01(0.01)*	0.02(0.05)*	0.01(0.04)*	–(–)	0.00(0.00)*	0.33(0.33)	**0.49**(**0.73**)
	NMI_C	0.02(0.08)*	0.02(0.04)*	0.01(0.01)*	0.02(0.05)*	0.01(0.04)*	–(–)	0.00(0.00)*	0.33(0.33)	**0.49**(**0.73**)
D_{43}	NMI_R	0.01(0.05)*	0.01(0.03)*	0.01(0.01)*	0.01(0.03)*	0.01(0.04)*	–(–)	0.00(0.00)*	**0.34**(0.34)	0.24(**0.71**)
	NMI_C	0.01(0.05)*	0.01(0.03)*	0.01(0.01)*	0.01(0.03)*	0.01(0.04)*	–(–)	0.00(0.00)*	**0.34**(0.34)	0.24(**0.71**)



(a) Average time costs versus the number of rows.

(b) Average time costs versus the number of columns.

Fig. 7. The execution time versus the size of data.

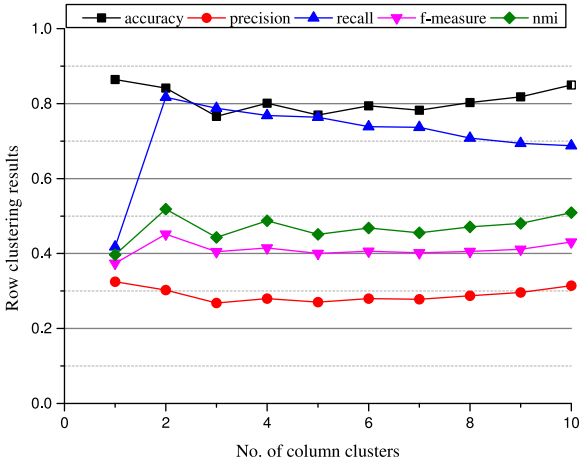
values were nearly stable. This can be explained from an optimization point of view. TWCC cannot be easily trapped in local optimum without cluster merging and the result was usually good. The other algorithms can easily converge to local optima without cluster merging, but the result may not be good. In a summary, TWCC were more robust than other co-clustering algorithms.

6.2. Scalability analysis

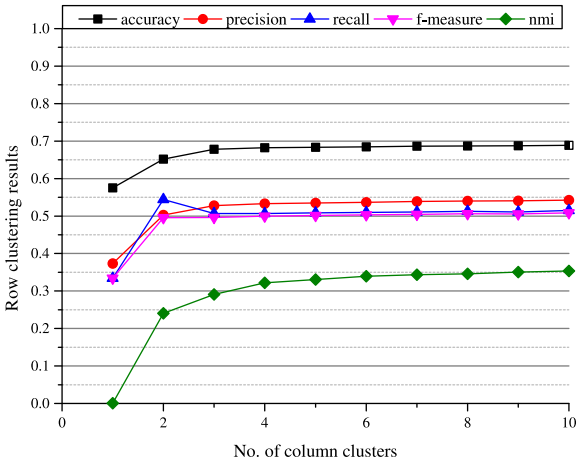
We generated two groups of synthetic data sets for scalability comparison. In the first group, we set the number of columns in the data set as 1000 and changed the numbers of rows as 10 integers from 10 numbers range from 200 to 12800. In the second group, the number of rows in the data set was fixed to 1000, whereas the numbers of columns were also set as 10 numbers range from 200 to 12800. The input values of parameters of six co-clustering algorithms (excluding BBAC and ITCC) 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. 7. We can see that most algorithms have a nearly linear relationship between the execution time and the size of data. Compared with RWCC, RSWCC, CWCC and CSWCC, the execution time of BBAC and TWCC was more stable as the size of data increased. Interestingly, FFCFW

Table 5
Characteristics of six real-life data sets.

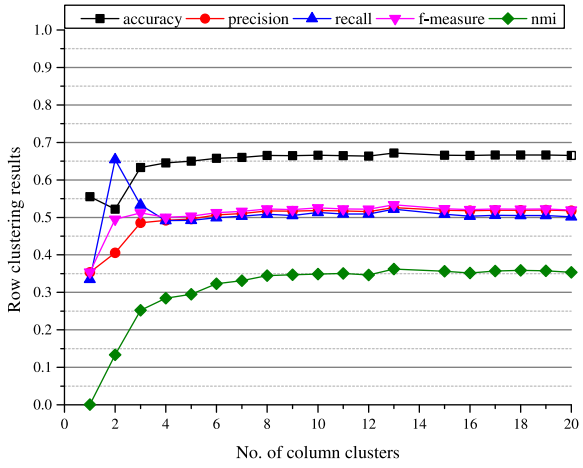
Name	Size	Clusters
Image Segmentation (**IS**)	2310 × 19	7
Waveform Database Generator (Version 1) (**WD1**)	5000 × 21	3
Waveform Database Generator (Version 2) (**WD2**)	5000 × 40	3
Cardiotocography (Version 2) (**CD2**)	2126 × 21	10
Leukemia1(**L1**)	72 × 7129	3
Brain Tumor2(**BT2**)	50 × 10368	4



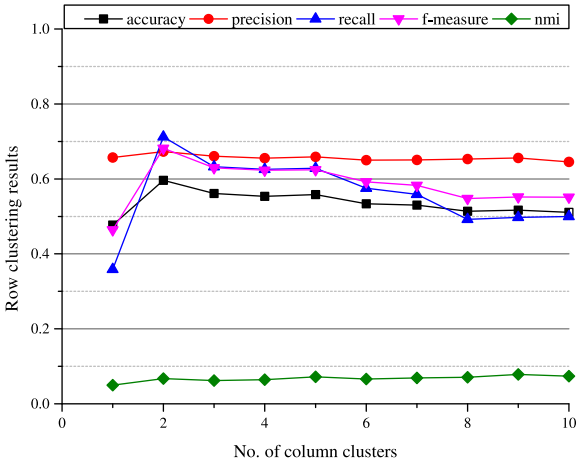
(a) Clustering results on **IS** data set.



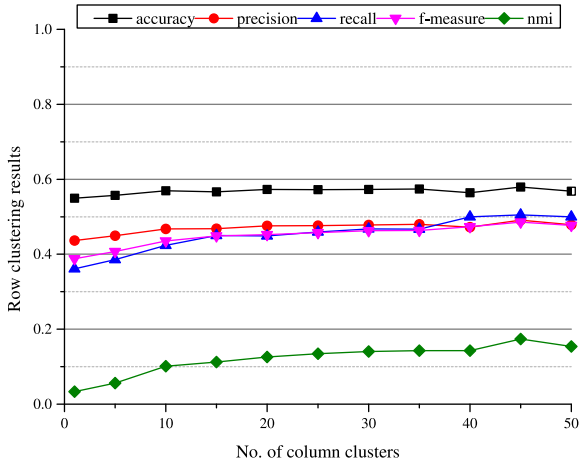
(b) Clustering results on **WD1** data set.



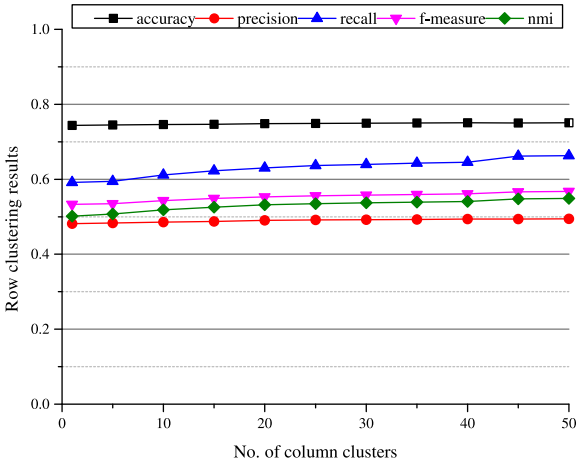
(c) Clustering results on **WD2** data set.



(d) Clustering results on **CD2** data set.



(e) Clustering results on **L1** data set.



(f) Clustering results on **BT2** data set.

Fig. 8. The relationship between the row clustering results and the number of column clusters L of TWCC.

showed good scalability to the number of rows, but bad scalability to the number of columns. Although TWCC was an extension to BBAC, the execution time of TWCC was less than that of BBAC. This may be because that TWCC converged in less iterations than BBAC. In a summary, TWCC scales well to large high-dimensional data.

7. One-way clustering performance comparisons on real-life data sets

Although TWCC is a co-clustering algorithm, it can be used for clustering along one-way of the data [7]. In this experiment, we used six real-life data sets to compare the one-way clustering

Table 6
Comparisons of clustering results of seven clustering algorithms on six real-life data sets.

Data	evaluation	*k*-means	EWKM	BBAC	ITCC	FFCFW	LDCC	TWCC
IS	Accuracy	0.79(0.85)*	0.79(0.85)*	0.80(0.85)*	0.59(0.78)*	–(–)	0.79(0.86)*	**0.81**(**0.88**)
	Precision	0.33(0.48)*	0.33(0.48)*	0.36(0.49)	0.22(0.32)*	–(–)	0.34(0.48)*	**0.36**(**0.51**)
	Recall	0.46(0.62)*	0.46(0.62)*	0.48(0.63)*	0.61(**1.00**)*	–(–)	0.60(0.75)	**0.68**(0.94)
	F-Measure	0.38(0.52)*	0.38(0.52)*	0.41(0.52)	0.31(0.39)*	–(–)	0.40(0.52)*	**0.41**(**0.56**)
	NMI	0.43(0.59)*	0.43(0.59)*	0.45(0.59)*	0.25(0.40)*	–(–)	0.45(0.60)*	**0.47**(**0.66**)
WD1	Accuracy	0.66(0.74)*	0.65(0.74)*	0.65(0.74)*	0.56(0.58)*	–(–)	0.65(0.74)*	**0.67**(**0.77**)
	Precision	0.49(0.62)*	0.48(0.62)*	0.48(0.60)*	0.37(0.39)*	–(–)	0.48(0.61)*	**0.51**(**0.64**)
	Recall	0.53(0.71)*	0.53(0.71)*	0.49(0.72)*	0.47(**1.00**)*	–(–)	**0.54**(0.82)	0.50(0.80)
	F-Measure	0.52(0.64)*	**0.52**(0.64)*	0.48(0.64)*	0.40(0.50)*	–(–)	0.50(0.63)	0.49(**0.68**)
	NMI	0.35(0.48)*	**0.35**(0.48)*	0.30(0.47)*	0.06(0.10)*	–(–)	0.30(0.40)	0.29(**0.48**)
WD2	Accuracy	0.67(0.75)*	**0.67**(0.74)*	0.65(0.74)*	0.33(0.33)*	–(–)	0.66(0.72)	0.65(**0.77**)
	Precision	0.48(0.62)*	0.48(0.62)*	0.47(0.60)*	0.33(0.33)*	–(–)	0.48(0.61)*	**0.51**(**0.65**)
	Recall	0.52(0.70)*	0.52(0.70)*	0.48(0.71)*	**1.00**(**1.00**)*	–(–)	0.52(0.78)	0.50(0.96)
	F-Measure	0.48(0.63)*	0.48(0.63)*	0.47(0.65)*	0.50(0.50)*	–(–)	0.49(0.64)*	**0.52**(**0.69**)
	NMI	0.34(0.45)*	0.34(0.45)*	0.28(0.45)*	0.00(0.00)*	–(–)	**0.35**(0.45)	0.27(**0.49**)
CD2	Accuracy	0.50(0.62)*	0.50(0.62)*	0.49(0.71)*	0.52(0.65)*	–(–)	0.51(0.66)*	**0.53**(**0.72**)
	Precision	0.67(0.80)*	0.67(0.80)*	0.66(0.83)*	0.65(0.77)*	–(–)	0.66(0.82)*	**0.68**(**0.84**)
	Recall	0.42(0.73)*	0.42(0.73)*	0.39(0.67)*	0.46(1.00)*	–(–)	0.48(0.85)*	**0.55**(**1.00**)
	F-Measure	0.51(0.70)*	0.51(0.70)*	0.49(0.74)*	0.53(0.78)*	–(–)	0.54(0.79)*	**0.58**(**0.81**)
	NMI	0.08(0.19)*	0.08(0.19)*	0.07(0.25)*	**0.09**(0.20)	–(–)	0.08(0.26)	0.07(**0.30**)
L1	Accuracy	0.55(0.77)*	0.55(0.77)*	0.55(0.82)*	0.41(0.41)*	–(–)	0.55(0.85)*	**0.57**(**0.92**)
	Precision	0.45(0.71)*	0.45(0.71)*	0.44(0.77)*	0.41(0.41)*	–(–)	0.44(0.80)*	**0.47**(**0.89**)
	Recall	0.60(0.94)*	0.60(0.94)*	0.44(0.92)*	**1.00**(**1.00**)*	–(–)	0.50(0.90)	0.46(0.96)
	F-Measure	0.51(0.72)*	0.51(0.72)*	0.44(0.78)*	0.58(0.58)*	–(–)	**0.59**(0.80)*	0.45(**0.88**)
	NMI	0.18(0.60)*	**0.18**(0.60)*	0.10(0.58)*	0.00(0.00)*	–(–)	0.14(0.70)	0.12(**0.75**)
BT2	Accuracy	0.73(0.80)*	0.73(0.78)*	0.72(0.79)*	0.25(0.25)*	–(–)	0.72(0.79)*	**0.75**(**0.82**)
	Precision	0.47(0.61)*	0.47(0.57)*	0.46(0.56)*	0.25(0.25)*	–(–)	0.45(0.60)*	**0.49**(**0.63**)
	Recall	0.68(0.94)*	0.60(0.79)*	0.63(0.94)	**1.00**(**1.00**)*	–(–)	0.52(0.92)*	0.64(0.94)
	F-Measure	0.53(0.65)*	0.54(0.62)*	0.53(0.69)*	0.40(0.40)*	–(–)	0.51(0.65)*	**0.55**(**0.69**)
	NMI	0.52(0.67)*	0.52(0.60)*	0.51(0.71)*	0.00(0.00)*	–(–)	0.51(0.68)*	**0.53**(**0.72**)

performance of TWCC with two one-way clustering algorithms *k*-means and EWKM (Entropy Weighting *k*-Means) [13], and the following four co-clustering algorithms:

- **BBAC** (Bregman Block Average co-clustering) with the Squared Euclidean distance [29].
- **ITCC** (Information-Theoretic Co-Clustering) [7].
- **FFCFW** (Fuzzy weighting co-clustering) [16].
- **LDCC** is the Locally Discriminative Coclustering proposed in [35].

We selected four real-life data sets from UCI Machine Learning Repository and two gene data sets from http://www.gems-system.org for the following experiments. The characteristics of these data were listed in Table 5. Since all real-life data sets contained only one-way labels, we treated the number of column clusters as a parameter and investigated the relationship between the row clustering results and the number of column clusters in TWCC. We specified the number of column clusters L as 10 integers range from 1 to 10 for the data sets **IS**, **WD1** and **CD2**. For data set **WD2**, we set L as 20 integers range from 1 to 20. For data sets **L1**, L was set as {1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50}. For each combination of K and L, we produced 40,000 clustering results for each clustering algorithm except FFCFW in a similar way as those in Section 5.2, and 800,000 clustering results for FFCFW in a similar way as those in Section 6.1. Finally, we only kept the clustering results with specified number of clusters/co-clusters. We computed precision, recall, f-measure, accuracy and NMI to evaluate the results, each of which uses the corresponding actual classification as the reference classification.

7.1. Impact of L on row clustering performance of TWCC

For each data set, we computed *NMI* from each row clustering result of TWCC and then computed the average *NMI* values of the clustering results with the same L. We draw the relationship between the row clustering results and L in Fig. 8. From these figures, we can see that almost all evaluation indices increased as L increased, and they became nearly stable when L became large. We also found that f-measure, recall and accuracy decreased as L increased on **CD2**, and most results were instable on **IS**. This may be because that the two data sets contain two few features, TWCC tends to be instable when L was too big. As a summary, L does not affect the clustering result of TWCC too much if it is not too small or too big, with respect to the number of columns.

7.2. Results and analysis

Table 6 summarized the clustering results in a similar way as Table 4. The "*" also indicates the difference between the average value from the corresponding algorithm (excluding TWCC) and the average value by the TWCC algorithm is significant by t-test.

From the results in Table 6, we can see that *k*-means and EWKM produced similar average values but lower maximum values than most partitional co-clustering algorithms, especially on **L1**. Among the eight co-clustering algorithms, FFCFW performed the worst, even produced zero NMI results on all data sets. ITCC produced zero NMI results on three data sets. In summary, TWCC significantly outperformed other six clustering algorithms in most clustering results, especially on **CD2** and **BT2**. For example, TWCC had 19 best average values and 22 best maximum values, which were far more than all other algorithms. This again indicates that the two-way weighting improves the performance of co-clustering.

8. Conclusions

In this paper, we have presented TWCC, a two-way subspace weighting partitional co-clustering method. In this method, two types of subspace weights have been introduced to simultaneously weight columns on row clusters and rows on column clusters. A series of experiments were conducted on both synthetic and real-life data sets and the results have demonstrated that TWCC is robust and scalable.

In the future work, we will study the ensemble clustering of TWCC and combine the two-way subspace weighting method with other techniques such as fuzzy clustering, semi-supervised clustering and apply TWCC to more real applications.

Acknowledgment

This research was supported by NSFC under Grant no. 61305059, 61773268 and 61473194.

References

[1] H.-P. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering, ACM Trans. Knowl. Discov. Data 3 (1) (2009) 1–58.
[2] X. Chen, J.Z. Huang, Q. Wu, M. Yang, Subspace weighting co-clustering of gene expression data, IEEE/ACM Trans. Comput. Biol. Bioinform. PP (99) (2017) 1-1.
[3] Y.-X. Wang, Y.-J. Zhang, Nonnegative matrix factorization: a comprehensive review, IEEE Trans. Knowl. Data Eng. 25 (6) (2013) 1336–1353.
[4] A.Y. Ng, M.I. Jordan, Y. Weiss, et al., On spectral clustering: analysis and an algorithm, Adv. Neural Inf. Process Syst. 2 (2002) 849–856.
[5] X. Chen, F. Nie, J.Z. Huang, M. Yang, Scalable normalized cut with improved spectral rotation, IJCAI (2017) 1518–1524.
[6] J. Hartigan, Direct clustering of a data matrix, J. Am. Stat. Assoc. (1972) 123–129.
[7] I. Dhillon, S. Mallela, D. Modha, Information-theoretic co-clustering, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003, pp. 89–98.
[8] S. Alzahrani, B. Ceran, S. Alashri, S.W. Ruston, S.R. Corman, H. Davulcu, Story forms detection in text through concept-based co-clustering, in: IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom), 2016, pp. 258–265.
[9] R.E. Thomas, S.S. Khan, Co-clustering with side information for text mining, in: 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), 2016, pp. 105–108.
[10] S.Y. Huang, H.J. Sun, C.D. Huang, I.F. Chung, C.H. Su, A modified fuzzy co-clustering (MFCC) approach for microarray data analysis, in: 2014 IEEE International Conference on Fuzzy Systems, 2014, pp. 267–272.
[11] S.F. Hussain, M. Ramazan, Biclustering of human cancer microarray data using co-similarity based co-clustering, Expert Syst. Appl. 55 (C) (2016) 520–531.
[12] A.L. Vizine Pereira, E.R. Hruschka, Simultaneous co-clustering and learning to address the cold start problem in recommender systems, Knowl. Based Syst. 82 (C) (2015) 11–19.
[13] L. Jing, M. Ng, Z. Huang, An entropy weighting *k*-means algorithm for subspace clustering of high-dimensional sparse data, IEEE Trans. Knowl. Data Eng. 19 (8) (2007) 1026–1041.
[14] X. Chen, X. Xu, Y. Ye, J.Z. Huang, TW-k-means: automated two-level variable weighting clustering algorithm for multi-view data, IEEE Trans. Knowl. Data Eng. 25 (4) (2013) 932–944. http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.262.
[15] X. Chen, Y. Ye, X. Xu, J.Z. Huang, A feature group weighting method for subspace clustering of high-dimensional data, Pattern Recognit. 45 (1) (2012) 434–446.
[16] W.-C. Tjhi, L. Chen, Flexible fuzzy co-clustering with feature-cluster weighting, in: ICARCV'06. 9th International Conference on Control, Automation, Robotics and Vision, IEEE, 2006, pp. 1–6.
[17] Y. Ye, X. Li, B. Wu, Y. Li, Feature weighting information-theoretic co-clustering for document clustering, in: 2009 2nd International Conference on Computer Science and its Applications, 2009, pp. 1–6.
[18] T. Sarazin, M. Lebbah, H. Azzag, A. Chaibi, Feature group weighting and topological biclustering, in: Neural Information Processing, Springer, 2014, pp. 369–376.
[19] Y. Liu, Q. Gu, J.P. Hou, J. Han, J. Ma, A network-assisted co-clustering algorithm to discover cancer subtypes based on gene expression, BMC Bioinform. 15 (1) (2014) 37.
[20] G. Govaert, M. Nadif, Co-Clustering, Wiley-ISTE, 2013. 31.
[21] B. Mirkin, P. Arabie, L.J. Hubert, Additive two-mode clustering: the error-variance approach revisited, J. Classif. 12 (2) (1995) 243–263, doi:10.1007/BF03040857.
[22] B. Mirkin, Mathematical Classification and Clustering: From How to What and Why, Springer Berlin Heidelberg, 1998.
[23] D. Duffy, J. Quiroz, A permutation-based algorithm for block clustering, J. Classif. 8 (1) (1991) 65–91.
[24] 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).
[25] R. Tryon, D. Bailey, Cluster Analysis, New York ; Maidenhead : McGraw-Hill, 1970.
[26] J. Li, B. Shao, T. Li, M. Ogihara, Hierarchical co-clustering: a new way to organize the music data, IEEE Trans. Multimedia 14 (2) (2012) 471–481, doi:10.1109/TMM.2011.2181151.
[27] Y. Cheng, G. Church, Biclustering of expression data, in: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, 1, 2000, pp. 93–103.
[28] T. Wu, A.R. Benson, D.F. Gleich, General tensor spectral co-clustering for higher-order data, in: D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems 29, Curran Associates, Inc., 2016, pp. 2559–2567.
[29] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, D. Modha, A generalized maximum entropy approach to Bregman co-clustering and matrix approximation, J. Mach. Learn. Res. 8 (2007) 1919–1986.
[30] L. Lazzeroni, A. Owen, Plaid models for gene expression data, Stat. Sin. 12 (1) (2002) 61–86.
[31] S. Busygin, G. Jacobsen, E. Krämer, C. Ag, Double conjugated clustering applied to leukemia microarray data, In 2nd SIAM ICDM, Workshop on Clustering High Dimensional Data, Citeseer, 2002.
[32] Y. Kluger, R. Basri, J. Chang, M. Gerstein, Spectral biclustering of microarray data: coclustering genes and conditions, Genome Res. 13 (4) (2003) 703–716.
[33] G. Govaert, M. Nadif, An em algorithm for the block mixture model, Pattern Anal. Mach. Intell. IEEE Trans. 27 (4) (2005) 643–647.
[34] H. Cho, I.S. Dhillon, Y. Guan, S. Sra, Minimum sum-squared residue co-clustering of gene expression data, in: Proceedings of the Fourth SIAM International Conference on Data Mining, 7, 2004, pp. 328–335.
[35] L. Zhang, C. Chen, J. Bu, Z. Chen, D. Cai, J. Han, Locally discriminative coclustering, IEEE Trans. Knowl. Data Eng. 24 (6) (2012) 1025–1035.
[36] Z. Huang, M. Ng, H. Rong, Z. Li, Automated variable weighting in k-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 657–668.
[37] C. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, 1, Cambridge University Press Cambridge, 2008.
[38] A. Strehl, J. Ghosh, Cluster ensembles-a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2003) 583–617.
[39] A. Lomet, G. Govaert, Y. Grandvalet, Design of Artificial Data Tables for Co–Clustering Analysis, Technical Report, Université de Technologie de Compiègne, France, 2012.
[40] G. Govaert, M. Nadif, Clustering with block mixture models, Pattern Recognit. 36 (2) (2003) 463–473.

Dr. Xiaojun Chen received his Ph.D. 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, and massive data mining.

Min Yang received the PhD degree from the University of Hong Kong, in February 2017. She is currently an assistant professor in the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. Her current research interests include machine learning and natural language processing.

Prof. Joshua Huang received his Ph.D. 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.

Prof. Zhong Ming is a professor at the College of Computer and Software Engineering of Shenzhen University. He is a member of a council and senior member of the Chinese Computer Federation. His major research interests are in software engineering and embedded systems. He led two projects of the National Natural Science Foundation, and two projects of the Natural Science Foundation of Guangdong province, China.