

Received February 27, 2020, accepted March 3, 2020, date of publication March 11, 2020, date of current version March 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979915

Co-Clustering Ensemble Based on Bilateral K-Means Algorithm

HUI YANG^{1,2}, HAN PENG^{1,2}, JIANYONG ZHU^{1,2}, AND FEIPING NIE^{1,3}

¹School of Electrical and Automation, East China Jiaotong University, Nanchang 330013, China

²Key Laboratory of Advanced Control and Optimization of Jiangxi Province, Nanchang 330013, China

³Center for Optical Image Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Feiping Nie (feipingnie@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61963015, Grant 61733005, Grant 61563015, and Grant 61863014, and in part by the Jiangxi Natural Science Fund Key under Grant 20171ACB21039 and Grant 20192BAB207024.

ABSTRACT Clustering ensemble technique has been shown to be effective in improving the accuracy and stability of single clustering algorithms. With the development of information technology, the amount of data, such as image, text and video, has increased rapidly. Efficiently clustering these large-scale datasets is a challenge. Clustering ensembles usually transform clustering results to a co-association matrix, and then to a graph-partition problem. These methods may suffer from information loss when computing the similarity among samples or base clusterings. Rich information between samples and base clusterings is ignored. Moreover, the results are not discrete. They need post-processing steps to obtain the final clustering result, which will deviate greatly from the real clustering result. To address this problem, we propose a co-clustering ensemble based on bilateral k-means (CEBKM) algorithm. Our algorithm can simultaneously cluster samples and base clusterings of a dataset, to fully exploit the potential information between the samples and the base clusterings. In addition, it can directly obtain the final clustering results without using other clustering algorithms. The proposed method, outperformed several state-of-the-art clustering ensemble methods in experiments conducted on real-world and toy datasets.

INDEX TERMS Clustering ensemble, bilateral k-means algorithm, co-clustering, base clustering.

I. INTRODUCTION

Clustering technique is applied in various fields, such as biology [1], image retrieval [2], information retrieval [3], and image processing [4]. Its overall purpose is to divide data into groups based on some criteria. The same type of data are divided into one group as much as possible. However, each clustering algorithm has its own bias due to the optimization of different criteria. An additional challenge for a single clustering method is that the data of the truth label are unavailable, hence it is difficult to validate the clustering results [5].

To resolve some of the challenges related to clustering, the idea of a clustering ensemble, also called consensus clustering or ensemble clustering, has been proposed. Clustering ensemble technique finds the underlying structure of data by combining different base clusterings into a

single consensus clustering, which can provide a more robust solution [6]–[14]. A clustering ensemble method consists of two steps: generation and consensus function. In the generation step, multiple base clusterings are obtained by operating traditional clustering methods on the same dataset. Several methods have been proposed to obtain the base clusterings, such as the same algorithm with different parameter initialization [15], [16], different clustering algorithms [17], combining multiple weak clustering algorithms [18], random projection [19], and data resampling [20], [21]. In the consensus function step, multiple base clusterings are combined into a matrix to improve the accuracy of the final clustering result [22]–[26]. Many methods have been proposed, using heuristics or meta-heuristics to find approximate solutions. These include co-association matrix-based methods [27], [28], graph-based methods [29], relabeling and voting methods [30], and locally adaptive cluster-based methods [5]. Other methods use an objective function to

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

measure the similarity between the clustering results and the consensus. These include k-means-based algorithms [18], nonnegative matrix factorization [31], the EM algorithm [32], and kernel-based algorithms [33].

In the above method, the co-association matrix-based methods have attracted the interest of many researchers [34]–[37]. The base clusterings are combined into a co-association matrix via analyzing how many times two samples belong together in the same cluster. Researchers try to derive pairwise similarity from clustering results and then transform the clustering ensemble to a graph-partition problem [34]. Graph-based methods usually transform clustering results into a co-association. This is very important in the clustering ensemble method, which determines how well to conduct graph partitioning for the final consensus partition. However, it only computes the similarity between data points, ignoring rich information between samples and base clusterings. The co-association matrix may lead to information loss, and possibly the inability to accurately find the underlying structure of the data.

Some graph-based methods construct a bipartite graph that contains all of the samples and base clusterings as vertices [38]. It is based on a spectral graph partitioning algorithm [39]. However, when dealing with multipartitioning problems, it needs to relax the original discrete problem to a continuous problem, and then use a k-means algorithm or METIS algorithm [40] to obtain the final discrete results. The discrete-continuous-discrete transformation can cause the final result to deviate greatly from the actual result.

In light of the deficiencies of the above method, we propose a co-clustering ensemble based on a bilateral k-means algorithm, which simultaneously clusters both samples and base clustering of the datasets. Our algorithm can directly obtain the final clustering result without using a traditional clustering algorithm. Our contributions are summarized as follows:

- 1) Our algorithm involves constraints of indicator matrices, which store the clustering results of samples and base clusterings. It can perform ensemble clustering by simultaneously utilizing the sample information and base clustering information. It can make full use of the information between the sample and the base clusterings to effectively improve the accuracy of the cluster.
- 2) Our algorithm can directly obtain the final clustering result. Many existing clustering ensemble methods require post-processing steps to obtain the result. Some, such as CPSA, HGPA, MCLA, and HGPA use a k-means or METIS algorithm to obtain the final clustering result.
- 3) We performed a large number of experiments on several datasets. The clustering evaluation criteria NMI and AI were utilized to compare the algorithm's with that of other advanced clustering algorithms. Experimental results show that our method outperforms them.

The rest of this paper is organized as follows. In section II, we revisit related work on clustering ensembles. In section III,

we propose the co-clustering ensemble with BKM model and we introduce a clustering algorithm and its update process. We compare performance of the proposed model with several related methods in section IV. We end with conclusions, acknowledgements and references.

II. RELATED WORK

Clustering ensembles have been shown to effectively improve the accuracy and stability of single clustering algorithms. They generate a set of base clusterings from the same dataset and combine them into a final clustering.

The information of a group of base clusterings is combined in a co-association matrix, which counts how often two samples are in the same cluster [34], and it is used as a similarity matrix to apply the graph partitioning method to obtain the clustering result. Strehl and Ghosh proposed three graph-based algorithms [17], the cluster-based similarity partitioning algorithm (CSPA), hyperGraph partitioning algorithm (HGPA) and meta-clustering algorithm (MCLA). In CSPA, a binary similarity matrix is constructed for the cluster of the ensemble. Each column corresponds to a clustering that has an entry of 1 if the corresponding point belongs to the clustering, and 0 otherwise. The similarity matrix is used to recluster the samples using a graph-partitioning approach, where the vertices correspond to the samples and the edge weights correspond to the similarities. METIS [40] is used to partition the graph for the final clustering result. In HGPA, all of the hyperedges have the same weight, and each hyperedge represents a clustering of ensembles. HGPA looks for c unconnected components of the same size by dividing the hyperedges. The hypergraph partitioning package HMETIS [40] is used to generate the final clustering result. MCLA is based on the clustering of clusters. It provides object-wise confidence estimates of cluster membership. This algorithm groups and collapses related hyperedges and assigns an object to the collapsed hyperedge in which it participates most strongly.

Hybrid bipartite graph formulation (HBGF) was proposed by Fern and Brodley [38]. A bipartite graph is constructed, where both samples and base clusterings simultaneously act as vertices in the graph. In the bipartite graph, all of the edges have weight 1, and clustering vertices are only connected to sample vertices. Sample vertices are also only connected to clustering vertices. The multi-way spectral graph partitioning algorithm obtains the bipartite graph partitioning result, and METIS is used to obtain the final clustering result. A spectral co-clustering ensemble (SCCE) [29] also constructs a bipartite graph to convert a bipartite graph partition problem to matrix decomposition. The final clustering result was got by k-means. K-means-based Consensus Clustering (KCC) [41] was proposed as a theoretic framework, which provided the sufficient and necessary condition for KCC utility functions to exactly map the ensemble clustering to a K-means problem with theoretical support.

More recently, Yazhou Ren proposed three other consensus techniques to leverage the weighted samples [42].

The difficulty of clustering a sample is estimated by constructing the co-association matrix and embedding the corresponding information as weights associated to samples. Three algorithms were proposed Weighted-Object Meta Clustering Algorithm (WOMC) is a weighted version of the Meta-Clustering Algorithm (MCLA). The meta-graph is constructed, and it partitions the meta-graph into c meta-clusters, each representing a group of clusters, by using the similarity measure and applying the METIS [40] graph-partitioning algorithm. Each object is assigned to the meta-cluster in which it participates most strongly. The weighted-Object similarity partitioning algorithm (WOSP) moves centers once for each base clustering by using the original data and objects' weight information. Clustering with a boosting technique aims to generate a better single clustering result while we seek to find a better consolidated consensus solution, constructing the corresponding graph and partitioning the graph into c parts, utilizing METIS. Each represents a final clustering result. Weighted-Object hybrid bipartite graph partitioning(WOHB) reduces the clustering consensus problem to a bipartite graph partitioning problem, which simultaneously partitions both clustering vertices and sample vertices. The hybrid bipartite graph is constructed and METIS partitions it into c parts. The partition of the objects provides the final clustering result.

None of the above methods can directly obtain the final clustering result, and some cannot simultaneously cluster samples and base clusterings of a dataset. Almost all the graph partitioning methods need other clustering algorithms to get the final result. In this process, it relaxes the discrete solution (the final clustering result) to a continuous solution, which is obtained by eigendecomposition. Getting the final clustering result from eigenvectors usually involves k-means or the METIS algorithm. However, this will deviate from the optimal solution. Also, some methods only consider the relationship between samples, or between base clusterings, ignoring the relationship between samples and base clusterings. In contrast, our proposed ECBKM algorithm can directly obtain the final clustering result. Our algorithm can make full use of the information relating the sample and the base clusterings to improve the accuracy of the clustering ensemble.

III. CO-CLUSTERING ENSEMBLE BILATERAL K-MEANS ALGORITHM

In this section, we first introduce some notation, then we will briefly review the hybrid bipartite graph formulation (HBGF) techniques for solving co-clustering ensemble problems, and explain the co-clustering ensemble bilateral k-means algorithm in detail.

A. NOTATION

In this section, we will give some notations in this paper. We write matrices as boldface uppercase letters and vectors as boldface lowercase letters. The transpose of any matrix Z is denoted by Z^T , and the trace of Z is denoted by $Tr(Z)$.

TABLE 1. Details and other notations are summarized.

n	Number of samples;
d	Number of features;
c	Number of clusters;
m	The times of running k-means;
Z	For any matrix Z ;
Z^T	The transpose of the matrix Z ;
$Tr(Z)$	The trace of the matrix Z ;
$z_i \in \mathbb{R}^{1 \times n}$	The i -th row of Z ;
$z_j \in \mathbb{R}^{n \times 1}$	The j -th column of Z ;
$X = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^{n \times d}$	Data matrix;
$A \in \mathbb{R}^{(n+d) \times (n+d)}$	The adjacency matrix;
$D \in \mathbb{R}^{(n+d) \times (n+d)}$	The diagonal matrix;
$L \in \mathbb{R}^{(n+d) \times (n+d)}$	The Laplacian matrix;
$S \in \mathbb{R}^{c \times c}$	The diagonal matrix;
$F \in \mathbb{R}^{n \times c}$	The indicator matrix;
$G \in \mathbb{R}^{d \times c}$	The indicator matrix;
$R \in \mathbb{R}^{n \times (c \times m)}$	The new data matrix ;

We denote $z_i \in \mathbb{R}^{1 \times n}$ as the i -th row of Z , and the $z_j \in \mathbb{R}^{n \times 1}$ is the j -th column of Z .

Given a data matrix $X = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^{n \times d}$, the vector x_i denotes the i -th row of X , and the vector x_j is the j -th column of X . \mathcal{h}^c is represented as a clustered set of samples. τ^c is represented as a clustered set of features. $A \in \mathbb{R}^{(n+d) \times (n+d)}$ is the adjacency matrix. $D \in \mathbb{R}^{(n+d) \times (n+d)}$ is the diagonal matrix. $L \in \mathbb{R}^{(n+d) \times (n+d)}$ is the Laplacian matrix, $L = D - A$.

$F = [f_1^T, f_2^T, \dots, f_n^T]^T \in \{0, 1\}^{n \times c}$ and $G = [g_1^T, g_2^T, \dots, g_d^T]^T \in \{0, 1\}^{d \times c}$ are two partitioned matrices that represent the clustering results of samples and features. If the i -th sample x_i belongs to cluster \mathcal{h}^j , then $f_{ij} = 1$, and if the j -th feature x_i belongs to cluster τ^j , then $g_{ij} = 1$. Each row in F and G has one and only one nonzero element, and the others are all zero. We call this an indicator matrix, and denote the set of them as \emptyset , i.e., $F \in \emptyset^{n \times c}$ and $G \in \emptyset^{d \times c}$. Details and other notations are summarized in Table 1.

B. OTHER RECOMMENDATIONS

C. HYBRID BIPARTITE GRAPH FORMULATION

Hybrid bipartite graph formulation(HBGF) uses the spectral method to convert the problem of co-clustering to a partition problem on a bipartite graph. HBGF combines multiple base clusterings of a dataset and samples to generate a bipartite graph $G = \langle V, A \rangle$, where all of the base clusterings of a dataset and samples are contained in V as a vertex. The adjacency matrix L can be written as:

$$A = \begin{bmatrix} 0 & M^T \\ M & 0 \end{bmatrix}, \quad (1)$$

where M is the edge-weights matrix, whose rows and columns correspond to the samples and base clusterings, respectively. Base clustering vertices are only connected to sample vertices and the edges in the graph have weights of 1, otherwise 0.

$$M(i, j) = \begin{cases} 1 & \text{sample } i \text{ belongs to the } j\text{-th cluster} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The multi-way spectral graph partitioning algorithm is used in a bipartite graph. Spectral clustering models use the eigenvalues of the Laplacian matrix for the data to perform dimensionality reduction before clustering. The Laplacian matrix L of G can be defined as:

$$L = D - A, \tag{3}$$

where D is a diagonal matrix, which can be expressed as $D(ii) = \sum_j A(ij)$. Spectral clustering models seek to optimize the normalized cut criterion. The objective of normalized cut is defined as:

$$\min_Q Tr(Q^T L Q) \quad s.t. \quad Q^T D Q = I, \tag{4}$$

where $Q \in \mathbb{R}^{n \times c}$ is the identity matrix and c is a reduced dimension. The optimization problem in equation (4) is converted to compute the normalized weight matrix $L = D^{-1}A$. It aims to find L 's c (number of clusters) largest eigenvectors, to get the final clustering solution by using the METIS algorithm.

D. REVISITING THE BILATERAL K-MEANS ALGORITHM

Given a data matrix $X \in \mathbb{R}^{n \times d}$, its i -th row is denoted as x_i , and its j -th column as x_j . The columns represent features and the rows represent samples. Co-clustering based on diagonal co-clusters aims to group the samples into c clusters and the features into c clusters.

Constructing a bipartite graph $G' = \langle V, A \rangle$ between samples and features of a dataset $X \in \mathbb{R}^{n \times d}$, where V is the set of vertices corresponding to samples and features, A is the adjacency matrix constructed by data matrix X :

$$A = \begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix}. \tag{5}$$

BKM aims to find the multipartitioning normalized cuts of G' by solving the optimization problem

$$\min_Y \sum_{k=1}^c \frac{y_k^T L y_k}{y_k^T D y_k}, \quad s.t. \quad Y \in \phi^{(n+d) \times c}, \tag{6}$$

where D is the diagonal matrix, the diagonal element $D_{ii} = \sum_{j=1}^{n+d} A_{ij}$, each row of Y has only one nonzero element, and the Laplacian matrix $L = D - A$. The first n numbers of $Y = [y_1^T, y_2^T, \dots, y_{(n+d)}^T]^T$ indicate the clustering results of samples, and the remaining numbers indicate the clustering results of features. since D is a diagonal matrix, $Y^T D Y$ is also a diagonal matrix, with the diagonal elements $(Y^T D Y)_{kk}$ equal to $y_k^T D y_k$; thus the optimization problem in equation (6) can be rewritten as

$$\min_Y Tr(Y^T L Y (Y^T D Y)^{-1}) \quad s.t. \quad Y \in \phi^{(n+d) \times c}, \tag{7}$$

where $Tr(\cdot)$ is the trace of the matrix. Since $L = D - A$ and $Y^T = [F^T, G^T]$, we can rewrite (7) as

$$\min_Y Tr(Y^T L Y (Y^T D Y)^{-1})$$

$$\begin{aligned} &= \min_Y Tr(Y^T (D - A) Y (Y^T D Y)^{-1}) \\ &= \min_Y Tr(I - Y^T A Y (Y^T D Y)^{-1}) \\ &= \min_{F,G} Tr(I - 2F^T X G (Y^T D Y)^{-1}). \end{aligned} \tag{8}$$

So, the optimization problem in equation (8) is changed to

$$\begin{aligned} &\min_{F,G} Tr(-F^T X G (Y^T D Y)^{-1}) \\ &s.t. \quad F \in \phi^{n \times c}, G \in \phi^{d \times c}. \end{aligned} \tag{9}$$

Since the problem is NP-complete [43], we solve the optimization problem in equation (9) by relaxing the discrete constraint problem, adding the two parts $Tr((Y^T D Y)^{-1} F^T F (Y^T D Y)^{-1} G^T G)$ and $Tr(X^T X)$. So, the optimization problem in equation (9) can be rewritten as

$$\begin{aligned} &\min_{F,G} \|X - F(Y^T D Y)^{-1} G^T\|_F^2 \\ &s.t. \quad F \in \phi^{n \times c}, G \in \phi^{d \times c}. \end{aligned} \tag{10}$$

We simplify the solution process of the optimization problem in equation (10) by replacing the $(Y^T D Y)^{-1}$ diagonal matrix with the S diagonal matrix, taking S as a new parameter, so the optimization problem in equation (10) is simplified to:

$$\begin{aligned} &\min_{F,G,S} \|X - F S G^T\|_F^2 \\ &s.t. \quad F \in \phi^{n \times c}, G \in \phi^{d \times c}, S \in diag, \end{aligned} \tag{11}$$

where $diag$ represents the diagonal matrix, F and G are the indicator matrices, indicating the clustering results of samples and the clustering results of features.

E. CLUSTERING ENSEMBLE

The clustering ensemble has two steps: generation and consensus function. The first step uses k-means to generate multiple clustering results, each with a random initialization of cluster centers. Each cluster is rewritten as an indicator matrix to generate them, then we use the new matrix to obtain a bipartite graph.

Given an input dataset $X = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^{n \times d}$, we run the k-means algorithm m times to get base clusterings $h' = [h^1, h^2, \dots, h^m] \in \mathbb{R}^{n \times m}$. Each clustering result h^m simply divides the dataset X into c disjoint parts; then we treat each clustering result as a feature vector, expressed as $h^m = [h_1^m, h_2^m, \dots, h_c^m]^T \in \mathbb{R}^{n \times c}$. Simultaneously, each clustering result h^m is rewritten into an indicator matrix $r^m \in \{0, 1\}^{n \times c}$; if the sample $x_i (0 \leq i \leq n)$ belongs to the cluster c , then $r_{ic} = 1$. Each row in r^m has one and only one nonzero element, and the other elements are zero. The given initial dataset X is converted to $R = [r^1, r^2, \dots, r^m]$, $R \in \{0, 1\}^{n \times (c \times m)}$.

Considering a simple example, we run the k-means three ($m = 3$) times on the dataset $X = [x_1, x_2, x_3, x_4, x_5, x_6]^T$, where each time the dataset is grouped into three ($c = 3$) classes. The clustering results are shown in Fig. 1.

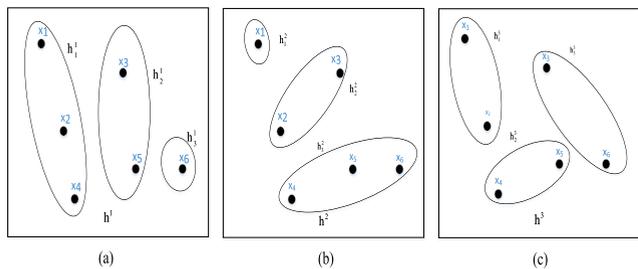


FIGURE 1. (a), (b), (c) represent the result of clustering result.

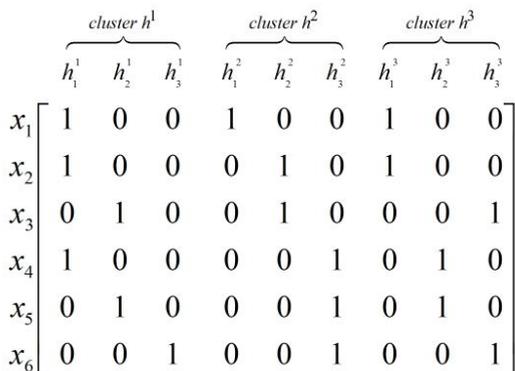


FIGURE 2. The label vectors is changed to an indicator matrix and treating them as a feature.

The sample label vectors can be represented as

$$h^1 = [1, 1, 2, 1, 2, 3]^T \quad h^2 = [1, 2, 2, 3, 3, 3]^T$$

$$h^3 = [1, 1, 3, 2, 2, 3]^T$$

Then this result of label vectors is changed to an indicator matrix. Connecting these three partial indicator matrices and treating them as a new feature, we obtain a new matrix, defined as an R matrix in Fig. 2.

We use the new data matrix R to generate a bipartite graph. Using the bipartite graph model [44], the base clustering clusters $\tau^c (1 \leq c \leq k)$ and sample clusters $h^c (1 \leq c \leq k)$ are determined as follows. A given sample x_i belongs to sample cluster h^m if its association with the base clustering cluster τ^m is greater than its association with any other base clustering cluster. Thus,

$$h^m = \{x_i : \sum_{j \in \tau^m} X_{ij} \geq \sum_{j \in \tau^l} X_{ij}, \forall l = 1, 2, \dots, k\}, \quad (12)$$

Each of the sample cluster is determined by the base clustering. Similarly the base clustering cluster is given by

$$\tau^m = \{x_j : \sum_{i \in h^m} X_{ij} \geq \sum_{i \in h^l} X_{ij}, \forall l = 1, 2, \dots, k\}, \quad (13)$$

We can see that there is a recursive relationship between τ^c and h^c , and the relations described in Eq.(12) and Eq.(13) determine bipartite spectral graph partition is based on diagonal co-cluster structure.

To get the final clustering result by solving the optimization problem in equation (11), as follows.

First, initialize F, G . Since these are indicator matrices, F and $G, F^T F$ and $G^T G$ are diagonal matrices. Calculate S with F and G fixed to solve S . Expand the objective function as follows and define it as

$$J = \|R - FSG^T\|_F^2$$

$$= Tr(R^T R) - 2Tr(G^T R^T FS) + Tr(SF^T FSG^T G)$$

$$= Tr(R^T R) - 2Tr(G^T R^T FS) + Tr(SF^T FG^T GS) \quad (14)$$

Then the constraints of S are relaxed. We calculate the partial derivative of J with respect to S and set it to zero. The objective function (14) is changed to:

$$\frac{\partial J}{\partial S} = -2F^T RG + 2(F^T FG^T G)^T S. \quad (15)$$

Thus we get $S = [(F^T FG^T G)^T]^{-1}(F^T RG)$.

Second, calculate G with F and S fixed. The optimization problem to solve G can be decomposed into $(c \times m)$ simple subproblems for each $i(1 \leq i \leq (c \times m))$. Simultaneously, the new data set matrix R also can be decomposed into $(c \times m)$ simple subproblems for each $i(1 \leq i \leq (c \times m))$:

$$\min_{G \in \phi^{(c \times m) \times c}} \|r_i - Wg_i^T\|_F^2, \quad (16)$$

where $W = FS$. Because there is only one element equal to 1 and the remaining elements are zero in the vector g_i , the solution of equation (16) can be obtained from:

$$g_{ij} = \begin{cases} 1 & j = \arg \min_k \|r_i - w_k\|^2 \\ 0 & otherwise, \end{cases} \quad (17)$$

where w_k is the k -th column of W .

Finally, we update G with F and S fixed. The optimization problem to solve F can be decomposed into n simple subproblems for each $i(1 \leq i \leq n)$. The new dataset matrix R also can be decomposed into n simple subproblems for each $i(1 \leq i \leq n)$,

$$\min_{F \in \phi^{n \times c}} \|r_i - f_i L\|_F^2, \quad (18)$$

where $L = SG^T$. There is only one element equal to 1, and the remaining elements are zero in the vector f_i , and the solution of equation (18) can be obtained from:

$$f_{ij} = \begin{cases} 1 & j = \arg \min_k \|r_i - l_k\|^2 \\ 0 & otherwise, \end{cases} \quad (19)$$

where l_k is the k -th row of L .

We summarize the main steps of our algorithm as follows

Algorithm 1 Co-Clustering Ensemble Based on Bilateral K-Means Algorithm

Require: The data matrix $X \in \mathbb{R}^{n \times d}$, and c is the number of clusters.

1. Run k-means algorithm to cluster the samples (n) into c clusters for m times.
2. The result of each clustering is changed to an indicator matrix. i.e., if the sample $x_i (1 \leq i \leq n)$ belongs to the cluster c then $x_{ic} = 1$, and the other numbers are zero, i.e., $x_{ij} = 0 (j \neq c, 1 \leq j \leq c)$.
3. Combine the clustering solutions to obtain a new matrix $R \in \mathbb{R}^{n \times (c \times m)}$.

Initialize F and G indicator matrices.

repeat

- Update S with F and G by using Eq. (15).
- Update G with F and S by using Eq. (17).
- Update F with G and S by using Eq. (19).

until Converge

Ensure: Indicator matrix F is the sample clustering.

IV. EXPERIMENTS

We conducted experiments on several datasets to evaluate of the effectiveness of the proposed method. There were two aspects to the experiments. The first was to compare NMI and AI of the proposed CEBKM with other clustering ensemble methods on the real datasets.

A. DATASETS

We compared the performance of our method with that of several state-of-the-art clustering techniques. We conducted experiments on twelve datasets. The characteristics of all these datasets are summarized in Table 2.

TABLE 2. The description of twelve datasets.

Datasets	Number of Samples	Features	Classes
Twomoon	200	2	2
Gaosi	200	2	2
Zoo	101	16	7
Yeast	1484	1470	10
Pima	768	8	2
Hear	270	13	2
Ecoli	366	343	8
Diabetes	768	8	2
Crx	690	15	2
Australian	690	14	2
corel-5k	5000	423	50
MnistData-10	6996	784	10

The real datasets (Zoo, Yeast, Pima, Heart, Ecoli, Diabetes, Crx, Australian, corel-5k, MnistData-10) are from benchmark datasets. MnistData-10 is 10% sampled from the original Mnist dataset. The number of classes ranges from 2 to 50, the number of samples from 101 to 6996, and the number of features from 8 to 1470.

The toy datasets (Twomoon, Gaosi) are shown in Fig. 3. Toy dataset 1 (Twomoon) consists of two classes of data distributed in a moon shape. Each cluster has 100 samples and the noise percentage is set to be 0.1. Toy dataset 2 (Gaosi2)

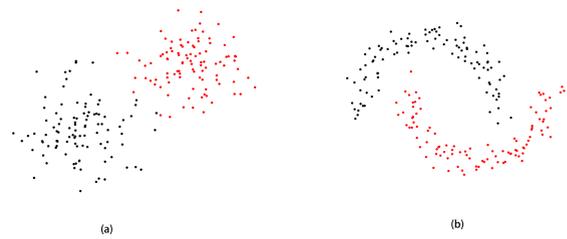


FIGURE 3. (a) is Gaosi dataset. (b) is Twomoon dataset.

consists of two classes with 100 data points each. The two classes are generated according to multivariate Gaussian distributions, and all share the same covariance matrix $[1 \ 0; 0 \ 2]$. The mean vectors are $(7, 8)$ $(10, 11)$.

1) EVALUATION CRITERIA

Since the labels of the datasets in Table 2 are known, we can use the normalized mutual information (NMI) [17] and the Rand index (RI) [45] to quantitatively validate our model.

Given a clustering result α (the clustering result by running the algorithm) and the true clustering labels β , The number of clusters in α and classes in β are both c (the number of clusters). Suppose n is the number of samples, n_i is the number of samples in the i -th cluster, n_j is the number of samples in the j -th class, and n_{ij} is the number of samples in the i -th cluster and j -th class. An NMI between α and β is calculated as follows:

$$NMI(\alpha, \beta) = \frac{\sum_{i=1}^c \sum_{j=1}^c n_{ij} \log \frac{n \times n_{ij}}{n_i n_j}}{\sqrt{\sum_{i=1}^c n_i \log \frac{n_i}{n} \sum_{j=1}^c n_j \log \frac{n_j}{n}}} \tag{20}$$

Denote by a_{11} the number of pairs of samples that are both partitioned in the same cluster in α and are also partitioned in the same cluster in β . Let a_{00} be the number of pairs of samples that are partitioned in different clusters in α and are also partitioned in different clusters in β . a_{01} is the number of pairs of samples that are partitioned in different classes in α and in the same class in β . a_{10} is the number of pairs of samples that are placed in the same class in α and in different classes in β . The Rand index is:

$$RI(\alpha, \beta) = \frac{a_{00} + a_{11}}{a_{00} + a_{01} + a_{10} + a_{11}} \tag{21}$$

Both NMI and RI range from 0 to 1, and the higher values of RI or NMI indicate better clustering performance.

2) COMPARISON SETTINGS

We compared CEBKM with the state-of-the-art clustering ensemble algorithms including HBGF, HGPA, MCLA, CSPA, WOHB, WOSP, WOMC, and with the k-means algorithm. Source codes for the compared methods were obtained from the authors' websites. The experimental setup is described as following. To generate input to the clustering algorithm, we set c as the number of ground-truth classes.

TABLE 3. Experimental results in the normalized mutual information (where the best values are highlighted in bold).

Data sets	Ensemble size	HGBF	HGPA	MCLA	CSPA	WOHB	WOSP	WOMC	Our	K-means
Zoo	20	0.748221	0.724082	0.781882	0.690736	0.693053	0.673069	0.786015	0.786371	0.718590
	40	0.751295	0.714350	0.782330	0.693016	0.748149	0.675121	0.789974	0.801096	
	60	0.753402	0.716766	0.797162	0.696721	0.787074	0.673567	0.794643	0.804803	
	80	0.752586	0.724130	0.796039	0.696072	0.804100	0.674242	0.797473	0.807757	
	100	0.763032	0.726516	0.795777	0.696230	0.800885	0.673976	0.798043	0.804281	
Yeast	20	0.047652	0.044952	0.022669	0.190598	0.079345	0.025149	0.024105	0.246224	0.153503
	40	0.046419	0.072427	0.007407	0.205700	0.095042	0.024024	0.005583	0.260566	
	60	0.047469	0.093933	0.008120	0.219090	0.105372	0.021382	0.005966	0.265851	
	80	0.044197	0.117001	0.006368	0.216693	0.096424	0.022183	0.005495	0.267836	
	100	0.047535	0.133575	0.005312	0.217313	0.116906	0.021572	0.005969	0.268814	
Pima	20	0.047694	0.000796	0.093117	0.103069	0.106324	0.097122	0.089397	0.108693	0.029700
	40	0.058811	0.000869	0.091989	0.108427	0.105781	0.106318	0.086368	0.112832	
	60	0.051244	0.001022	0.097997	0.111722	0.103359	0.107219	0.088712	0.112460	
	80	0.058362	0.000885	0.099738	0.112664	0.100981	0.114209	0.086114	0.114244	
	100	0.062826	0.000583	0.103267	0.114237	0.106582	0.114051	0.091741	0.114107	
Heart	20	0.263094	0.000675	0.327174	0.293215	0.284441	0.281799	0.323722	0.323947	0.018989
	40	0.279003	0.000834	0.328237	0.294695	0.283542	0.282360	0.327061	0.329644	
	60	0.274693	0.001029	0.328206	0.295364	0.287399	0.284041	0.328225	0.328465	
	80	0.273265	0.001247	0.328581	0.296889	0.287987	0.281409	0.327451	0.329438	
	100	0.275117	0.001336	0.328951	0.297559	0.289189	0.282534	0.327333	0.327695	
Ecoli	20	0.132224	0.391969	0.296021	0.478791	0.379573	0.372882	0.291313	0.570667	0.546702
	40	0.136215	0.472642	0.226100	0.489041	0.38791	0.372409	0.216874	0.584163	
	60	0.176418	0.497472	0.408434	0.493271	0.396637	0.385469	0.449280	0.584200	
	80	0.117302	0.507149	0.290740	0.494712	0.431214	0.381746	0.37014	0.587288	
	100	0.144818	0.486600	0.440949	0.498324	0.437705	0.384278	0.462610	0.588200	
Diabetes	20	0.057171	0.000938	0.0948	0.107921	0.098339	0.091688	0.057171	0.102846	0.029700
	40	0.052434	0.001066	0.08822	0.108944	0.093831	0.102828	0.089127	0.114088	
	60	0.052812	0.001049	0.094029	0.112249	0.090447	0.112863	0.112863	0.112868	
	80	0.058326	0.000805	0.099650	0.110827	0.078536	0.113816	0.090435	0.113606	
	100	0.057338	0.000644	0.101791	0.114224	0.078438	0.113957	0.097176	0.120200	
Crx	20	0.125639	0.000168	0.221900	0.280550	0.204292	0.230646	0.226513	0.303583	0.005580
	40	0.159207	0.000129	0.268665	0.288352	0.231342	0.224709	0.240996	0.313647	
	60	0.120064	0.000129	0.290413	0.273898	0.2305	0.225837	0.266292	0.317316	
	80	0.120280	0.000079	0.276856	0.273993	0.229979	0.227147	0.250527	0.315534	
	100	0.122244	0.000135	0.282666	0.298934	0.232662	0.223538	0.266233	0.319711	
Australian	20	0.274962	0.000971	0.375923	0.351996	0.347856	0.347269	0.374503	0.374999	0.033111
	40	0.256233	0.000863	0.375949	0.356717	0.350175	0.342248	0.375225	0.376745	
	60	0.260276	0.000870	0.376939	0.370592	0.349407	0.34713	0.376466	0.376776	
	80	0.251024	0.000851	0.376999	0.371766	0.349622	0.350664	0.376543	0.376838	
	100	0.265395	0.001015	0.377284	0.375598	0.349987	0.340967	0.376912	0.376807	

TABLE 4. Experimental results in the normalized mutual information (where the best values are highlighted in bold).

Data sets	HGPA	MCLA	CSPA	WOHB	WOSP	WOMC	SCCE	KCC	Our
Zoo	0.736118	0.80224	0.69242	0.752881	0.674462	0.801434	0.794830	0.801546	0.804546
Yeast	0.093096	0.010408	0.201926	0.062168	0.021044	0.016184	0.192376	0.237690	0.265851
Pima	0.001182	0.098516	0.113896	0.103434	0.11605	0.111256	0.110348	0.112470	0.116664
Heart	0.013232	0.315414	0.290436	0.299412	0.29682	0.309034	0.311140	0.304051	0.320465
Ecoli	0.498136	0.396816	0.48995	0.254466	0.373634	0.402158	0.498376	0.510372	0.598522
Diabetes	0.001724	0.106344	0.115626	0.089468	0.113156	0.087318	0.112034	0.105763	0.117562
Crx	0.001314	0.29148	0.252664	0.223552	0.232188	0.279948	0.293204	0.296582	0.317316
Australian	0.00077	0.377608	0.361038	0.344914	0.348754	0.376402	0.364472	0.357904	0.376776
corel_5k	0.238550	0.226610	0.242891	0.235425	0.262530	0.231390	0.234047	0.243182	0.263102
MnistData_10	0.239430	0.392681	0.381532	0.270460	0.113012	0.392900	0.35167	0.394710	0.405030

Simultaneously, in the generation step, we ran k-means 20, 40, 60, 80, and 100 times with random initiation with selected centers to divide the dataset into *c* disjoint parts. We independently ran the clustering ensemble algorithms 100 times.

B. EXPERIMENTAL RESULTS ON TOY DATASETS

We compared the performance of k-means and CEBKM on the toy datasets. We set the ensemble size to 40. RI was used to evaluate the performance of the algorithms.

TABLE 5. Averages value of RI((where the best values are highlighted in bold)).

Datesets	k-means	CEBKM
Twomoon	0.6211	0.8164
Gaosi	0.9510	0.9801

We independently ran the two algorithms 50 times, and we recorded the average values of RI. The results are shown in Table 5. Best performances that are statistically significant

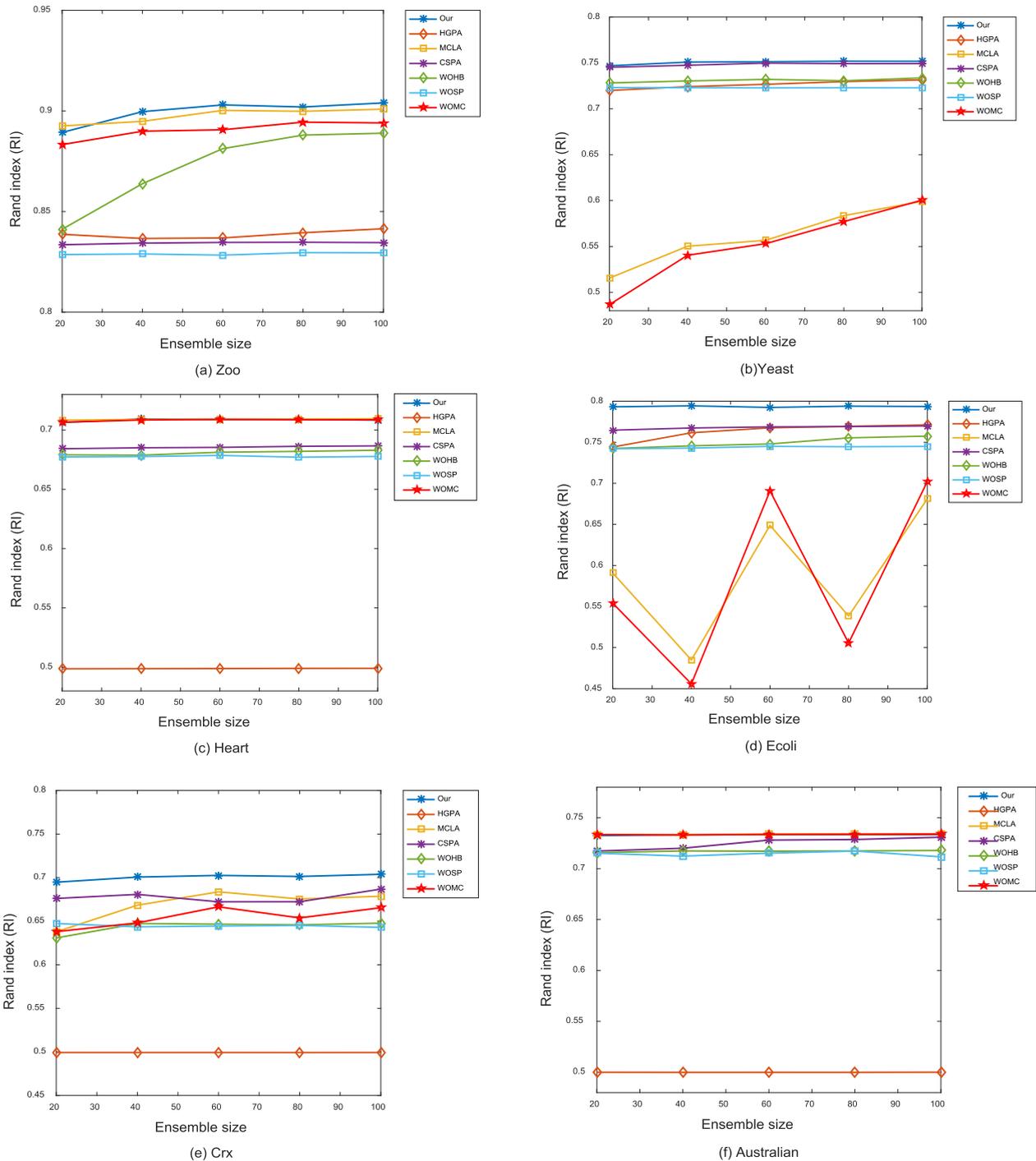


FIGURE 4. Experimental results of the Rand index on Zoo, Yeast, Heart, Ecoli, Crx and Australian, where the number of ensemble size varies from 20 to 100.

are highlighted in boldface. Results in Table 5 show that CEBKM significantly outperformed k-means on toy datasets. were still larger than those of k-means. This demonstrates the effectiveness of ensemble techniques.

C. EXPERIMENTAL RESULTS ON REAL DATASETS

Table 3 shows the normalized mutual information (NMI) of all of the algorithms on the eight real datasets. In the

experiments, each method had 100 independent runs, and the average results were computed. As seen from the Table 3, we can find that the proposed method CEBKM achieved better results than the other clustering ensemble methods, except for the clustering result on Australian dataset, where our value was still close to the best value.

To determine the relationship between the clustering result and the number of the ensemble size, we performed the other

experiment on the six datasets (Zoo, Yeast, Heart, Ecoli, Crx, Australian). Each clustering result for each dataset was generated by combining 20, 40, 60, 80, and 100 ensemble size (base clusterings). The average clustering performance of the six datasets with the ensemble size ascending from 20 to 100 is shown in Fig. 4, from which it can be observed that the clustering result was almost constant or increased when the ensemble size increased from 20 to 100, except that for Ecoli dataset, the clustering result of HGPA and WOMC had a large fluctuations. The overall trend was to increase with increasing ensemble size. Moreover, the clustering results remained almost constant, since there were no significant differences between these base clusterings. Fig. 4 shows that the clustering result of the CEBKM was the best among the six results. This reveals that the number of the ensemble size is beneficial. We can also find that the value of CEBKM increased with the ensemble size.

Another experiment to explore the relationship between the clustering result and the diversity of the ensemble size. Since the experimental effect of HGPA is not ideal, we add SCCE and KCC. Each of the base clusterings is produced by the k-means algorithm with the number of clusters k randomly selected in the interval of $[2, \sqrt{N}]$, where N is the number of objects in the dataset. In this paper, 40 base clusterings are randomly generated for the experiment on the datasets (Zoo, Yeast, Pima, Heart, Ecoli, Diabetes, Crx, Australian, corel_5k, MnistData_10). As seen from the Table 4, we can find that the proposed method CEBKM achieved better results than the other clustering ensemble methods.

V. CONCLUSION

We have proposed a clustering ensemble model named CEBKM. Unlike the traditional ensemble methods that only work on the sample or base clusterings of the datasets. CEBKM can simultaneously cluster the sample and the base clusterings of a dataset, and it can make full use of the relationship between the sample and the base clusterings of the datasets to get better clustering result. What is more, our algorithm can efficiently obtain the final clustering result without post-processing. When the clustering algorithm converges, the result stored in the indicator matrix is the final clustering result. Experimental results show that our method outperforms other advanced clustering algorithms. Moreover, we studied the relationship between the clustering results and the ensemble size. Our analysis shows that diversity of the base clusterings is beneficial.

REFERENCES

- [1] R. Sharan and R. Shamir, "CLICK: A clustering algorithm with applications to gene expression analysis," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 2000, p. 16.
- [2] 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.
- [3] S. K. Bhatia and J. S. Deogun, "Conceptual clustering in information retrieval," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 427–436, Jun. 1998.
- [4] J. A. F. Costa and M. L. de A. Netto, "Cluster analysis using self-organizing maps and image processing techniques," in *Proc. IEEE Conf. Int. Conf. Syst., Man, Cybern. (SMC)*, vol. 5, Oct. 1999, pp. 367–372.
- [5] C. Domeniconi and M. Al-Razgan, "Weighted cluster ensembles: Methods and analysis," *ACM Trans. Knowl. Discovery Data*, vol. 2, no. 4, 2009, Art. no. 17.
- [6] H. Parvin, I. Jafari, and F. Rad, "A clustering ensemble learning method based on the ant colony clustering algorithm," *Int. J. Innov. Comput. Appl.*, vol. 8, no. 3, p. 172, 2017.
- [7] H. Parvin, B. Minaei-Bidgoli, H. Alinejad-Rokny, and W. F. Punch, "Data weighing mechanisms for clustering ensembles," *Comput. Electr. Eng.*, vol. 39, no. 5, pp. 1433–1450, Jul. 2013.
- [8] M. Mojarad, S. Nejatian, H. Parvin, and M. Mohammadpoor, "A fuzzy clustering ensemble based on cluster clustering and iterative fusion of base clusters," *Int. J. Speech Technol.*, vol. 49, no. 7, pp. 2567–2581, Jul. 2019.
- [9] B. Minaei-Bidgoli, H. Parvin, H. Alinejad-Rokny, H. Alizadeh, and W. F. Punch, "Effects of resampling method and adaptation on clustering ensemble efficacy," *Artif. Intell. Rev.*, vol. 41, no. 1, pp. 27–48, Jan. 2014.
- [10] H. Alizadeh, M. Yousefnezhad, and B. M. Bidgoli, "Wisdom of crowds cluster ensemble," *Intell. Data Anal.*, vol. 19, no. 3, pp. 485–503, Jun. 2015.
- [11] D. Huang, J.-H. Lai, and C.-D. Wang, "Robust ensemble clustering using probability trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1312–1326, May 2016.
- [12] M. Yousefnezhad, S.-J. Huang, and D. Zhang, "WoCE: A framework for clustering ensemble by exploiting the wisdom of crowds theory," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 486–499, Feb. 2018.
- [13] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm," *Pattern Anal. Appl.*, vol. 18, no. 1, pp. 87–112, Feb. 2015.
- [14] D. Huang, C.-D. Wang, H. Peng, J. Lai, and C.-K. Kwok, "Enhanced ensemble clustering via fast propagation of cluster-wise similarities," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.
- [15] A. L. N. Fred and A. K. Jain, "Data clustering using evidence accumulation," in *Proc. 16th Int. Conf. Pattern Recognit.*, Aug. 2002, pp. 276–280.
- [16] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on elite selection of weighted clusters," *Adv. Data Anal. Classification*, vol. 7, no. 2, pp. 181–208, Jun. 2013.
- [17] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, 2003, pp. 583–617.
- [18] A. Topchy, A. K. Jain, and W. Punch, "Combining multiple weak clusterings," in *Proc. 3rd IEEE Int. Conf. Data Mining*, Nov. 2003, pp. 331–338.
- [19] X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," in *Proc. 20th Int. Conf. Int. Conf. Mach. Learn.*, 2003, pp. 186–193.
- [20] B. Minaei-Bidgoli, A. Topchy, and W. F. Punch, "Ensembles of partitions via data resampling," in *Proc. Int. Conf. Inf. Technol., Coding Comput. (ITCC)*, 2004, pp. 188–192.
- [21] B. Fischer and J. M. Buhmann, "Bagging for path-based clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 11, pp. 1411–1415, Nov. 2003.
- [22] H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin, "To improve the quality of cluster ensembles by selecting a subset of base clusters," *J. Experim. Theor. Artif. Intell.*, vol. 26, no. 1, pp. 127–150, Jan. 2014.
- [23] D. Huang, C.-D. Wang, J. Wu, J.-H. Lai, and C. K. Kwok, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Trans. Knowl. Data Eng.*, to be published.
- [24] D. Huang, C.-D. Wang, and J.-H. Lai, "Locally weighted ensemble clustering," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1460–1473, May 2018.
- [25] H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin, "Cluster ensemble selection based on a new cluster stability measure," *Intell. Data Anal.*, vol. 18, no. 3, pp. 389–408, 2014.
- [26] D. Huang, J. Lai, and C.-D. Wang, "Ensemble clustering using factor graph," *Pattern Recognit.*, vol. 50, pp. 131–142, Feb. 2016.
- [27] A. Nazari, A. Dehghan, S. Nejatian, V. Rezaie, and H. Parvin, "A comprehensive study of clustering ensemble weighting based on cluster quality and diversity," *Pattern Anal. Appl.*, vol. 22, no. 1, pp. 133–145, Feb. 2019.
- [28] A. Sadr-Olah, N. Samad, P. Hamid, R. Vahideh, and B. Karamolah, "Clustering ensemble selection considering quality and diversity," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1311–1340, 2019.
- [29] S. Huang, H. Wang, D. Li, Y. Yang, and T. Li, "Spectral co-clustering ensemble," *Knowl.-Based Syst.*, vol. 84, pp. 46–55, Aug. 2015.

- [30] H. G. Ayad and M. S. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 160–173, Jan. 2008.
- [31] H. Shinnou and M. Sasaki, "Ensemble document clustering using weighted hypergraph generated by NMF," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics Companion*, 2007, pp. 77–80.
- [32] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1866–1881, Dec. 2005.
- [33] S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper, "Weighted partition consensus via kernels," *Pattern Recognit.*, vol. 43, no. 8, pp. 2712–2724, 2010.
- [34] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, Jun. 2005.
- [35] X. Wang, C. Yang, and J. Zhou, "Clustering aggregation by probability accumulation," *Pattern Recognit.*, vol. 42, no. 5, pp. 668–675, May 2009.
- [36] F. Rashidi, S. Nejatian, H. Parvin, and V. Rezaie, "Diversity based cluster weighting in cluster ensemble: An information theory approach," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1341–1368, Aug. 2019.
- [37] S. Xu, Z. Lu, and G. Gu, "An efficient spectral method for document cluster ensemble," in *Proc. 9th Int. Conf. Young Comput. Scientists*, Nov. 2008, pp. 808–813.
- [38] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 36.
- [39] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst., Natural Synth.*, 2001, pp. 849–856.
- [40] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, Jan. 1998.
- [41] J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen, "K-means-based consensus clustering: A unified view," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 155–169, Jan. 2015.
- [42] Y. Ren, C. Domeniconi, G. Zhang, and G. Yu, "Weighted-object ensemble clustering: Methods and analysis," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 661–689, May 2017.
- [43] X. Y. Stella and J. Shi, "Multiclass spectral clustering," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, p. 313.
- [44] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2001, pp. 269–274.
- [45] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.



HUI YANG received the M.S. and Ph.D. degrees from Northeastern University, Shenyang, China, in 1988 and 2004, respectively. He is currently a Professor with the School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang, China. His current research interests include intelligent transportation systems, complex system modeling, control and optimization, and process industry integrated automation technology and applications.



HAN PENG is currently pursuing the M.S. degree with the School of Key Laboratory of Advanced Control and Optimization of Jiangxi Province, East China Jiaotong University, Nanchang, China. His research interests include machine learning and data mining.



JIANYONG ZHU received the Ph.D. degree from Central South University, China, in 2014. He is currently an Associate Professor with the School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang, China. His research interests include data analysis, random distribution control, and predictive control.



FEIPING NIE received the Ph.D. degree in computer science from Tsinghua University, China, in 2009. He has published more than 100 articles in the following top journals and conferences, such as TPAMI, IJCV, TIP, TNNLS/TNN, TKDE, TKDD, *Bioinformatics*, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, and ACM MM. His articles have been cited more than 5000 times (Google scholar). His research interests include machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval. He is currently serving as an Associate Editor or a PC Member for several prestigious journals and conferences in the related fields.

• • •