Research Article

A Fuzzy Co-Clustering Algorithm via Modularity Maximization

Yongli Liu , Jingli Chen , and Hao Chao

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, Henan, China

Correspondence should be addressed to Yongli Liu; yongli.buaa@gmail.com

Received 3 September 2018; Accepted 21 October 2018; Published 29 October 2018

Academic Editor: Francesco Lolli

Copyright © 2018 Yongli Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper we propose a fuzzy co-clustering algorithm via modularity maximization, named MMFCC. In its objective function, we use the modularity measure as the criterion for co-clustering object-feature matrices. After converting into a constrained optimization problem, it is solved by an iterative alternative optimization procedure via modularity maximization. This algorithm offers some advantages such as directly producing a block diagonal matrix and interpretable description of resulting co-clusters, automatically determining the appropriate number of final co-clusters. The experimental studies on several benchmark datasets demonstrate that this algorithm can yield higher quality co-clusters than such competitors as some fuzzy co-clustering algorithms and crisp block-diagonal co-clustering algorithms, in terms of accuracy.

1. Introduction

Co-clustering, also called biclustering or block-clustering, aiming to group a set of objects and a set of features simultaneously, has found an increasingly wide utilization in many fields. In the bioinformatics domain, co-clustering method has been used for simultaneously clustering genes and conditions [1]. Hussain and Ramazan [2] referred the theory of co-clustering text data and proposed a new algorithm based on weighted similarity, which generated two co-similarity matrices between genes and between conditions. In collaborative filtering, co-clustering technique is commonly used to simultaneously cluster users and items [3]. Honda et al. [4] utilized the sequential weighted co-clustering approach to find user-item co-clusters and recommended the proper items to the users according to the membership of users and items. In text mining, co-clustering algorithms are devoted to grouping documents and words simultaneously [5]. Tjhi and Chen [6] proposed a new fuzzy co-clustering algorithm for documents and words based on Ruspini's condition. The proposal improved the drawback of some previous co-clustering algorithms about the constraint for word memberships, which caused the failure to obtain practical and valuable natural fuzzy word clusters.

Fuzzy co-clustering further extends co-clustering by adding fuzzy membership function. It therefore simultaneously keeps advantages of co-clustering and fuzzy clustering, including dimensionality reduction, interpretable clusters, and improvement in accuracy [7]. In a fuzzy co-clustering framework, the fuzzy relationship is presented by the captured degree to which objects belong to each other, as well as the features. Based on the fuzzy relationship, the objective function is designed, which will be minimized or maximized through multiple iterations until the convergence is realized. Oh et al. [8] proposed the famous fuzzy clustering algorithm for categorical multivariate data (FCCM), which maximizes the co-occurrence of categorical attributes and the individual patterns in clusters. Kummamuru et al. [9] modified the FCCM algorithm so that it can be used to cluster large text corpora. Hanmandlu et al. [10] presented a novel fuzzy co-clustering algorithm for images color segmentation, named FCCI. The FCCI formulates an objective function which contains a multidimensional distance function as the dissimilarity measure and entropy as the regularization term. Forsati et al. [11] introduced a new fuzzy co-clustering approach based on genetic algorithm, computed the similarity between web pages and users, and proposed recommendations for hybrid recommender systems. These works show that fuzzy co-clustering can always achieve higher clustering accuracy,

because of its fuzzy mathematical method which determines the fuzzy relationship among samples.

Although fuzzy co-clustering algorithms could be simultaneously carried out on both the object and feature dimensions, the co-clusters generated cannot be intuitively illustrated via graphs. In order to address this issue, some block-diagonal co-clustering techniques have been proposed that can generate a block-diagonal matrix. It has the advantage of directly producing interpretable descriptions of the resulting document clusters [12].

Aykanat et al. [13] proposed bipartite graph and hyper-graph models, which transform Linear Programming problems into block diagonal forms and obtain good clustering performance on sparse datasets. Laclau and Nadif [14] proposed a hard and a fuzzy diagonal co-clustering algorithms based on double K-means, which can process sparse binary and continuous data effectively and generate a block diagonal data matrix by minimizing a criterion based on the intraclass variance and the centroid effect. Dhillon [15] described the co-clustering problem as a bipartite graph partitioning problem and proposed a new spectral co-clustering algorithm Spec, aiming to find minimum cut vertex partitions in a bipartite graph between documents and words.

Some researchers proposed block diagonal clustering approaches based on modularity. Compared with some nondiagonal co-clustering algorithms such as ITCC [16], which is a new information theoretic divisive algorithm, these algorithms can generate more descriptive and significant co-clusters. For example, Labiod and Nadif [17] proposed a generalized modularity measure and a spectral approximation of the modularity matrix. Aliem et al. [18] proposed a novel block-diagonal co-clustering algorithm named CoClus. This method maximizes the modularity using an iterative alternating procedure, which has been proved effective. Experimental results of these existing algorithms show that algorithms using the maximization of modularity as criterion performed well in the field of document clustering. Furthermore, modularity based clustering algorithms can help to easily determine the appropriate number of final (co-)clusters, which is one of the troublesome issues of clustering.

However, these existing clustering algorithms, including CoClus, are all hard clustering algorithms. In the field of clustering, we know that fuzzy (co-)clustering, allowing each object to belong to more than one cluster, is thought to be more consistent with human thinking than hard (co-)clustering. Therefore, in this paper we propose a block-diagonal fuzzy co-clustering algorithm via modularity maximization, named MMFCC. This algorithm not only evaluates the clustering quality by applying modularity to the fuzzy co-clustering technique, but also enhances the validity and accuracy of the final clustering result by introducing the fuzzy membership degrees.

The main contributions of this paper include the following:

- (i) We propose a novel fuzzy co-clustering algorithm which uses modularity measure as a useful criterion.
- (ii) MMFCC can easily produce a block diagonal matrix, which can be intuitively illustrated via graphs.

- (iii) We implement MMFCC on some real benchmark datasets and experimental results show that this algorithm can achieve higher accuracy than comparative algorithms.

- (iv) MMFCC can help to determine the appropriate number of final co-clusters.

The rest of this paper is organized as follows. Section 2 reviews the application of modularity in the fields of community detection and co-clustering. Section 3 presents our MMFCC approach. In Section 4, we demonstrate the effectiveness of MMFCC by carrying out experiments on 6 benchmark datasets. Finally, the conclusion and future work are given in Section 5.

2. Modularity

The maximization of modularity is known as an efficient and meaningful method in the aspect of indicating and evaluating the partitioned community structure in a network [19], whereas the sizes of the communities are not usually pre-established, which is similar to the mechanism of clustering in unsupervised learning. Given this, the modified modularity has been currently extended to text mining considered as a criterion for clustering text data.

2.1. Modularity for Community Detection. Newman and Girvan [20] proposed an approach that measures the community division results by using modularity which is defined as

$$Q = \sum_i (e_{ij} - a_i^2) \quad (1)$$

where Q denotes the value of modularity, e_{ij} represents the ratio of the number of edges connecting the community i and community j to the total number of edges, and $a_i = \sum_j e_{ij}$ denotes the ratio of the vertices connected to the sides in the community i to the total number of vertices.

In order to study the community structure in more complex and larger networks, Clauset et al. [21] defined the modularity as

$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w) \quad (2)$$

where $2m$ is the total edges in a network and A_{vw} is an element of the adjacency matrix for the network. The value of A_{vw} will be 1 if there is an edge between nodes v and w and 0 otherwise. k_v and k_w represent the degrees of nodes v and w , respectively, and c_v and c_w denote the two communities where nodes v and w are located, respectively. The value of $\delta(c_v, c_w)$ will equal 1 if $c_v = c_w$ meaning that nodes v and w are in the same community and 0 otherwise.

Let us define S_{vr} whose value will be 1 if node v belongs to group r and 0 otherwise. Meanwhile, we define S_{wr} whose value will be 1 if node w belongs to group r and 0 otherwise. Then the matrix form of modularity can be rewritten as

$$\begin{aligned} Q &= \frac{1}{2m} \sum_{vw} \sum_r \left[A_{vw} - \frac{k_v k_w}{2m} \right] S_{vr} S_{wr} \\ &= \frac{1}{2m} \text{Trace}(S^T B S) \end{aligned} \quad (3)$$

where $\text{Trace}(S^T BS)$ represents the sum of the elements on the main diagonal of $S^T BS$. The element of matrix B can be expressed as

$$B_{vw} = A_{vw} - \frac{k_v k_w}{2m} \quad (4)$$

Nowadays, modularity has become one popular measure of the structure of networks or graphs. The higher the value of modularity indicates more desirable and accurate network division results.

2.2. Modularity for Co-Clustering. As mentioned above, in the CoClus algorithm, Aliem et al. defined a block seriation relation $S = ZW^t$, where $C_{ij} = \sum_{k=1}^C z_{ik} w_{jk}$, and Z and W are $N \times C$ and $M \times C$ index matrices, respectively (C , N , and M represent the number of co-clusters, documents, and terms, respectively). And then for data matrix A with size $N \times C$, the modularity can be rewritten as

$$Q = \frac{1}{a_{..}} \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^C \left(a_{ij} - \frac{a_{i.} a_{.j}}{a_{..}} \right) z_{ik} w_{jk} \quad (5)$$

where $a_{..} = \sum_{i,j} a_{ij} = |E|$ represents the total number of edges, $a_{i.} = \sum_j a_{ij}$ and $a_{.j} = \sum_i a_{ij}$ express the degrees of i and j , respectively, and z_{ik} and w_{jk} represent the membership degrees of objects and features, respectively, which are different from the corresponding parameters in traditional co-clustering. The difference lies in that z_{ik} and w_{jk} have the same significance for the simultaneous clustering of documents and terms. If the object i belongs to the cluster Z_k , z_{ik} takes the value of 1 and 0 otherwise. Similarly, if attribute j belongs to the cluster W_k , the value of w_{jk} is 1, and 0 otherwise.

3. Fuzzy Block-Diagonal Co-Clustering Based on Modularity Maximization

In this paper, we propose an innovative and effective block-diagonal fuzzy co-clustering algorithm MMFCC, which combines the mentioned improved modularity with the idea of fuzzy co-clustering. The algorithm is described in detail as follows.

Let A be an $N \times M$ original data matrix, u_{ic} and v_{jc} be the fuzzy membership degrees that object i and feature j belong to cluster c , respectively, and $U = \{u_{ic}\}$ indicates the $N \times C$ document membership degree matrix and $V = \{v_{jc}\}$ represents the $M \times C$ document membership degree matrix. The MMFCC algorithm defines the modularity as

$$Q = \frac{1}{a_{..}} \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^C \left[\left(a_{ij} - \frac{a_{i.} a_{.j}}{a_{..}} \right)^t u_{ic} v_{jc}^t \right] \quad (6)$$

According to the definition, the objective function of MMFCC is given as

$$J_{MMFCC}(A, U, V) = \frac{1}{a_{..}} \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^C \left[\left(a_{ij} - \frac{a_{i.} a_{.j}}{a_{..}} \right)^t u_{ic} v_{jc}^t \right] \quad (7)$$

$$- T_u \sum_{i=1}^N \sum_{c=1}^C u_{ic} \log u_{ic} - T_v \sum_{j=1}^M \sum_{c=1}^C v_{jc} \log v_{jc}$$

where $\sum_{i=1}^N \sum_{c=1}^C u_{ic} \log u_{ic}$ and $\sum_{j=1}^M \sum_{c=1}^C v_{jc} \log v_{jc}$ denote the separate entropy regularizing terms of document and term membership degree functions separately, and minimizing these two items corresponds to maximizing the fuzzy entropies $-\sum_{i=1}^N \sum_{c=1}^C u_{ic} \log u_{ic}$ and $-\sum_{j=1}^M \sum_{c=1}^C v_{jc} \log v_{jc}$. Introducing T_u and T_v , two weighted fuzzy parameters that specify the ambiguity, will help contribute to increasing the clustering accuracy.

The objective function is subject to the following constraints:

$$\sum_{c=1}^C u_{ic} = 1, \quad u_{ic} \in [0, 1], \quad \forall i = 1, \dots, N \quad (8)$$

$$\sum_{c=1}^C v_{jc} = 1, \quad v_{jc} \in [0, 1], \quad \forall j = 1, \dots, M \quad (9)$$

The above constrained optimization problem of MMFCC can be solved by Lagrange multiplier method by applying the Lagrange multipliers λ_i and γ_j to constraints as (8) and (9), respectively, as follows:

$$\begin{aligned} Q(A, U, V) = & \frac{1}{a_{..}} \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^C \left[\left(a_{ij} - \frac{a_{i.} a_{.j}}{a_{..}} \right)^t u_{ic} v_{jc}^t \right] \\ & - T_u \sum_{i=1}^N \sum_{c=1}^C u_{ic} \log u_{ic} - T_v \sum_{j=1}^M \sum_{c=1}^C v_{jc} \log v_{jc} \\ & + \sum_{i=1}^N \lambda_i \left(\sum_{c=1}^C u_{ic} - 1 \right) + \sum_{j=1}^M \gamma_j \left(\sum_{c=1}^C v_{jc} - 1 \right) \end{aligned} \quad (10)$$

Deriving (10) for U and setting the gradient to zero, we have

$$\frac{\partial Q}{\partial U} = \sum_{j=1}^M \left(a_{ij} - \frac{a_{i.} a_{.j}}{a_{..}} \right)^t v_{jc}^t - T_u (1 + \log u_{ic}) + \lambda_i = 0 \quad (11)$$

Since the transposition does not change the values and size of a matrix, that is, the derivative value of a matrix transposition is the same as the matrix, thus in order to calculate conveniently, (11) can be written as

$$\frac{\partial Q}{\partial U} = \sum_{j=1}^M \left(a_{ij} - \frac{a_{i.} a_{.j}}{a_{..}} \right) v_{jc} - T_u (1 + \log u_{ic}) + \lambda_i = 0 \quad (12)$$

Applying u_{ic} obtained from (12) to the constraint as (8), we get

$$u_{ic} = \frac{e^{(\sum_{j=1}^M (a_{ij} - a_{i.} a_{.j} / a_{..}) v_{jc} / T_u)}}{\sum_{c=1}^C e^{(\sum_{j=1}^M (a_{ij} - a_{i.} a_{.j} / a_{..}) v_{jc} / T_u)}} \quad (13)$$

Derive (10) for V , set the gradient to zero, and we have

$$\frac{\partial Q}{\partial V} = \sum_{i=1}^N \left(a_{ij} - \frac{a_{i.} a_{.j}}{a_{..}} \right)^t u_{ic} - T_v (1 + \log v_{jc}) + \gamma_j = 0 \quad (14)$$

Apply v_{jc} obtained from (14) to the constraint as (9), and we have

$$v_{jc} = \frac{e^{(\sum_{i=1}^N (a_{ij} - a_{i.} a_{.j} / a_{..})^t u_{ic} / T_v)}}{\sum_{c=1}^C e^{(\sum_{i=1}^N (a_{ij} - a_{i.} a_{.j} / a_{..})^t u_{ic} / T_v)}} \quad (15)$$

1. Input: the original data matrix A and the number of co-clusters C, T_u, T_v
2. Output: fuzzy partitioning membership matrices u_{ic} and v_{jc}
3. Randomized initialization of u_{ic} and v_{jc}
4. Alternating iterations:
 5. Compute u_{ic} according to Eq(7)
 6. Compute v_{jc} according to Eq(9)
 7. Compute $Q(A, U, V)$ according to the obtained u_{ic} and v_{jc}
8. Stop the iterations until the number of $Q(A, U, V)$ keeps unchangeable
9. Gain the two sets of indices of maximum from each row according to the final u_{ic} and v_{jc} respectively
10. Acquire the reorganized matrix according to the above two sets of indices

ALGORITHM 1: MMFCC algorithm.

Put the final u_{ic} and v_{jc} obtained after the iteration into (7), and we can obtain the maximized modularity degree.

The MMFCC is outlined as shown in Algorithm 1.

4. Numerical Experiments

In this section, the MMFCC is compared with some other well-known block-diagonal and non-block-diagonal algorithms to validate its favourable clustering performance.

4.1. Datasets. Eleven text datasets with different sizes and sources are selected in the following experiments, and their detailed information is summarized in Table 1.

Given the reliability, integrity, and accuracy of our experiments, each dataset is divided into two parts: train set and test set. The train set contains 75% samples of the corresponding dataset, and its data is selected randomly. And the test data contains the remaining 25% data.

4.2. Evaluation Metrics. There already exist many common evaluation criteria for assessing the clustering equality [22] such as entropy, *Acc* (accuracy), and *NMI* (normalized mutual information). In this paper, *Acc* and *NMI* are chosen as two evaluated indices for the MMFCC algorithm.

Acc is used to describe how close the result values are to the true values of the series of experiments. It is defined as $Acc = (TP + TN) / (TP + FN + FP + TN)$, where a *TP* (True Positive) decision assigns two similar documents to the same cluster, a *TN* (True Negative) decision assigns two dissimilar documents to different clusters, a *FP* (False Positive) decision assigns two dissimilar documents to the same cluster, and a *FN* (False Negative) decision assigns two similar documents to different clusters. High value of *Acc* illustrates the excellent clustering results.

The other assessment index is *NMI*. Assume that S and T are the distributions of N sample labels; the entropy of the two distributions is

$$H(S) = \sum_{i=1}^{|S|} P(i) \log(P(i)) \quad (16)$$

where $P(i) = |S_i|/N$.

$$H(T) = \sum_{j=1}^{|T|} P'(j) \log(P'(j)) \quad (17)$$

where $P'(j) = |T_j|/N$.

The mutual information between S and T is defined as

$$MI(S, T) = \sum_{i=1}^{|S|} \sum_{j=1}^{|T|} P(i, j) \log\left(\frac{P(i, j)}{P(i)P'(j)}\right) \quad (18)$$

where $P(i, j) = |S_i \cap T_j|/N$.

The mutual information after normalization becomes

$$NMI(S, T) = \frac{MI(S, T)}{\sqrt{H(S)H(T)}} \quad (19)$$

High value of *NMI* indicates the excellent clustering results.

4.3. Fuzzy Parameters Settings. In fuzzy co-clustering algorithms, including our MMFCC, two fuzzy parameters, T_u and T_v , are introduced. These two parameters control the fuzzy degree of the algorithm. Often, in many other algorithms, T_u and T_v are assigned manually directly.

In this paper, in order to ensure the precision, we determine the values of these two parameters according to the experimental instances on train sets. Given amount of calculation, T_u and T_v take values in the range of $[1e-8, 1e+8]$ and increase by an integer multiple of 10. Figure 1 depicts the influence of different pairs of T_u and T_v on the values of modularity.

Figure 1 involves eleven graphs for the eleven datasets, respectively. It displays the three-dimensional curved surfaces distribution among modularity values and $\log T_u$, $\log T_v$. The three-dimensional surface fitting technique is adopted to fit the mean modularity values.

It can be observed from Figure 1 that the value range of $\log T_u$ and $\log T_v$ is located in $[-2, 4]$ and $[-3, 3]$, which means that T_u and T_v take values from $[0.01, 10000]$ and $[0.001, 1000]$ by an integer multiple of 10 for all datasets, respectively. Generally, the valid modularity is located in the range of $[0, 1]$ and Figures 1(a)–1(k) illustrate the point well.

It is easy to observe that there exists a highest position representing the maximum value of modularity in Figure 1. Table 2 summarizes the maximum modularity values and their corresponding pairs of T_u and T_v .

4.4. Evaluation of Clustering Performance. As mentioned above, we select *Acc* and *NMI* as the evaluation standards of our experiments. Table 3 lists all the *Acc* and *NMI* values

TABLE 1: The details of datasets.

Datasets	Documents	Words	Classes	Description
cstr	475	1000	4	The dataset contains the abstracts of technical reports published in the Department of Computer Science at the Rochester University between 1991 and 2002. It can be divided into 4 research areas: Natural Language Processing, Robotics/Vision, Systems and Theory.
tr23	204	5832	6	It is derived from TREC-5 collection of TREC collections and divided into 6 classes.
tr41	878	7454	10	It is derived from TREC-6 collection of TREC collections and divided into 10 classes.
tr45	690	8261	10	It is derived from TREC-7 collection of TREC collections and divided into 10 classes.
k1b	2340	21839	6	The dataset is from the WebACE project. Each document corresponds to a web page listed in the subject hierarchy of Yahoo. It includes 6 categories: health, entertainment, sports, politics, tech and business.
la2	3075	31472	6	This dataset is obtained from articles of the Los Angeles Times in the following 6 categories: entertainment, financial, foreign, metro, national and sports.
classic3	3891	4303	3	This dataset belongs to the Classic Collection and is widely used in text mining. It includes 3 different categories: CISI, CRAN and MED.
classic4	7095	5896	4	The dataset also belongs to the Classic Collection as well as the classic3, including 4 different categories: CACM, CRAN, CISI and MED.
RCV1	9625	29992	4	It is a subset of the RCV1-v2 corpus and contains the information of topics, regions and industries for each document, including 4 categories: C15, ECAT, GCAT and MCAT.
NG5	4702	26214	5	It is a subset of 20Newsgroups corpus and contains 5 categories, each with 799, 973, 985, 982 and 963 documents respectively.
ohscal	11162	11465	10	It is from the OHSUMED collection and contains 10 categories: antibodies, carcinoma, DNA, in-vitro, molecular sequence data, pregnancy, prognosis, receptors, risk factors and tomography.

TABLE 2: The values of T_u and T_v when the modularity of these datasets is maximized, respectively.

Datasets	T_u	T_v	Modularity
cstr	10	1	0.4136
tr23	0.1	1	0.4442
tr41	10	1	0.2754
tr45	0.1	10	0.2120
k1b	10	1	0.3755
la2	10	1	0.4084
classic3	1	10	0.3490
classic4	10	1	0.3259
RCV1	0.1	0.1	0.3380
NG5	10	10	0.1969
ohscal	1	10	0.3053

obtained by running MMFCC and other five comparative algorithms on eleven datasets for fifty times.

As shown in Table 3, our MMFCC algorithm achieves the highest *NMI* values on all the eleven datasets and the highest *Acc* values on ten datasets. The average *Acc* values of the six algorithms (Spec, ITCC, FCCM, FCCI, CoClus, and MMFCC) are 0.49, 0.43, 0.63, 0.68, 0.76, and 0.80, respectively, and the average *NMI* values are 0.36, 0.19, 0.21, 0.26, 0.35, and 0.50, respectively. Even on the cstr dataset, which is the only dataset in the experiments that MMFCC cannot achieve the best clustering accuracy in terms of *Acc*, the *Acc* value of MMFCC is comparable to that of Spec, the most outstanding algorithm on the dataset. In brief, this group of experiments shows that the MMFCC is better than or comparable to such competitors as Spec, ITCC, FCCM, FCCI, and CoClus in terms of *Acc* and *NMI*.

FIGURE 1: The distribution of values of modularity according to different T_u and T_v on eleven datasets. (a) cstr. (b) tr23. (c) tr41. (d) tr45. (e) k1b. (f) la2. (g) classic3. (h) classic4. (i) RCV1. (j) NG5. (k) ohscal.

4.5. Validation of Classification Correctness. One of the disadvantages of clustering analysis is that it is very difficult to determine the classification number of datasets in advance. In this condition, users need to give empirical values. It will reduce the automation level of clustering.

In MMFCC, this problem is solved well by using the modularity measure. This measure can help to find the optimal number of final clusters. In our experiments, we implemented MMFCC on the eleven datasets. Because the numbers of classes of these datasets are different (3, 4, 5, 6, and 10 clusters, respectively), we let $C = 2, 3, \dots, 8$, $C = 2, 3, \dots, 10$, $C = 6, 7, \dots, 14$, and $C = 7, 8, \dots, 13$ as the variation range of classification number, respectively. In MMFCC, there exists a pair of T_u and T_v that can maximize the modularity when each dataset takes a certain value of C . The relationship between the value of C and maximum modularity obtained from different C for all datasets is shown in Figure 2.

Figure 2 illustrates this group of experiments on the eleven datasets, cstr, tr23, tr41, tr45, k1b, la2, classic3, classic4, RCV1, NG5, and ohscal. It can be seen from Figure 2 that the optimal classification numbers for these eleven datasets are 4, 6, 10, 10, 6, 6, 3, 4, 4, 5, and 10, respectively. The results exactly match the original definition of these benchmark datasets.

As mentioned above, for each group of co-clusters of each dataset, there is a pair of T_u and T_v to maximize the result modularity. These pairs of T_u and T_v for different co-clusters of the eleven datasets are summarized in Tables 4–14, respectively.

According to the data shown from Tables 4–14, it is easy to see that the value range of T_u and T_v is $[0.1, 10.0]$. However, in terms of the values of T_u and T_v , the optimal number of co-clusters is obviously different from other compared ones.

Due to logarithm and exponent operations that are involved in the objective function, generally, the change of order of T_u and T_v may lead to different results. It can be seen from Figure 2 that the trends of curves on cstr, tr23, la2, and ohscal datasets are flatter when the corresponding fuzzy parameters stay invariant. But for the other datasets, the fluctuation of modularity values is relatively apparent even if T_u and T_v do not change. Moreover, it is easily observed that the optimal numbers of co-clusters for eleven datasets are, respectively, 3 for classic3, 4 for cstr, classic4, and RCV1, 5 for NG5, 6 for tr23, k1b, and la2, and 10 for tr41, tr45, and ohscal.

Figure 3 displays the clustering result graphs of MMFCC and five comparative algorithms. Given the limited space, we choose only the cstr dataset as a representative to intuitively

TABLE 3: Comparison of the several co-clustering algorithms in terms of *Acc* and *NMI* on eleven datasets. The bold values represent the best results.

Datasets	Acc					NMI						
	Spec	ITCC	FCCM	FCCI	CoClus	MMFCC	Spec	ITCC	FCCM	FCCI	CoClus	MMFCC
cstr	0.8596	0.5365	0.7862	0.7627	0.8220	0.8332	0.5318	0.3722	0.2491	0.4597	0.4705	0.5502
tr23	0.3585	0.4151	0.6508	0.6198	0.6507	0.7217	0.1329	0.1432	0.2240	0.2018	0.2677	0.4539
tr41	0.3750	0.3125	0.7112	0.7574	0.7203	0.7729	0.1913	0.0504	0.1734	0.1068	0.1238	0.4896
tr45	0.3371	0.2514	0.4836	0.6924	0.7416	0.7872	0.3112	0.0361	0.1384	0.1127	0.1510	0.4607
klb	0.4813	0.4524	0.6639	0.6423	0.6338	0.6800	0.3703	0.0124	0.2746	0.3085	0.3701	0.4478
la2	0.2724	0.3061	0.5968	0.6305	0.7454	0.7863	0.3875	0.0115	0.2136	0.3116	0.3113	0.5342
classic3	0.9794	0.7296	0.5213	0.6254	0.9528	0.9807	0.9137	0.7124	0.2706	0.3147	0.8763	0.9251
classic4	0.5710	0.5755	0.4953	0.7169	0.7523	0.8689	0.3979	0.3746	0.2170	0.3024	0.4532	0.6773
RCV1	0.3063	0.4210	0.6081	0.6870	0.7448	0.7589	0.0021	0.1044	0.1430	0.1569	0.2737	0.3026
NG5	0.4813	0.4524	0.6639	0.6423	0.7462	0.7562	0.3703	0.0124	0.2746	0.3085	0.3048	0.3204
ohscal	0.4059	0.2469	0.7145	0.6502	0.8237	0.8617	0.3613	0.2577	0.1413	0.2485	0.2666	0.3438

FIGURE 2: The number of co-clusters confirmed by the optimal modularity which are obtained by giving appropriate T_u and T_v acquired in Figure 1 in MMFCC algorithm on eleven datasets. (a) cstr. (b) tr23. (c) tr41. (d) tr45. (e) k1b. (f) la2. (g) classic3. (h) classic4. (i) RCV1. (j) NG5. (k) ohscal.

TABLE 4: Values of T_u and T_v that maximize modularity for different co-clusters of the cstr dataset.

Dataset	cstr						
#co-clusters	2	3	4	5	6	7	8
T_u	10	10	10	10	10	10	1
T_v	10	10	1	10	10	10	10
Modularity	0.21	0.21	0.41	0.17	0.15	0.15	0.14

TABLE 5: Values of T_u and T_v that maximize modularity for different co-clusters of the tr23 dataset.

Dataset	tr23								
#co-clusters	2	3	4	5	6	7	8	9	10
T_u	10	0.1	10	0.1	0.1	0.1	0.1	0.1	0.1
T_v	1	10	1	10	1	10	10	10	10
Modularity	0.23	0.17	0.20	0.17	0.44	0.15	0.14	0.14	0.13

demonstrate clustering performance and effect of these algorithms.

Now, we know that the optimal number of co-clusters on cstr dataset is four, and MMFCC and CoClus algorithms find the four co-clusters in Figures 3(a) and 3(b) clearly,

TABLE 6: Values of T_u and T_v that maximize modularity for different co-clusters of the tr41 dataset.

Dataset	tr41								
#co-clusters	6	7	8	9	10	11	12	13	14
T_u	1	1	1	1	10	1	1	1	0.1
T_v	10	10	10	10	1	10	10	10	1
Modularity	0.11	0.20	0.18	0.19	0.28	0.15	0.13	0.18	0.14

TABLE 7: Values of T_u and T_v that maximize modularity for different co-clusters of the tr45 dataset.

Dataset	tr45								
#co-clusters	6	7	8	9	10	11	12	13	14
T_u	1	1	1	1	0.1	0.1	0.1	0.1	0.1
T_v	10	10	10	10	10	10	10	10	10
Modularity	0.11	0.19	0.12	0.11	0.21	0.10	0.10	0.10	0.16

TABLE 8: Values of T_u and T_v that maximize modularity for different co-clusters of the k1b dataset.

Dataset	k1b								
#co-clusters	2	3	4	5	6	7	8	9	10
T_u	10	10	10	1	10	1	10	10	10
T_v	10	10	10	10	1	10	1	1	1
Modularity	0.14	0.12	0.17	0.11	0.38	0.10	0.21	0.28	0.20

TABLE 9: Values of T_u and T_v that maximize modularity for different co-clusters of the la2 dataset.

Dataset	la2								
#co-clusters	2	3	4	5	6	7	8	9	10
T_u	1	10	10	10	10	1	1	1	1
T_v	10	10	10	10	1	10	10	10	10
Modularity	0.14	0.12	0.16	0.15	0.41	0.13	0.13	0.12	0.12

TABLE 10: Values of T_u and T_v that maximize modularity for different co-clusters of the classic3 dataset.

Dataset	classic3						
#co-clusters	2	3	4	5	6	7	8
T_u	10	1	10	10	10	10	10
T_v	10	10	10	1	1	1	1
Modularity	0.27	0.35	0.25	0.33	0.32	0.31	0.29

TABLE 11: Values of T_u and T_v that maximize modularity for different co-clusters of the classic4 dataset.

Dataset	classic4						
#co-clusters	2	3	4	5	6	7	8
T_u	1	1	10	1	1	1	10
T_v	10	10	1	10	10	10	1
Modularity	0.23	0.27	0.33	0.26	0.25	0.24	0.26

respectively. However, the CoClus algorithm cannot group many documents and terms synchronously, and moreover there exist lots of scattered points compared with MMFCC. The Spec algorithm also generates block-diagonal co-clusters

TABLE 12: Values of T_u and T_v that maximize modularity for different co-clusters of the RCV1 dataset.

Dataset	RCV1						
#co-clusters	2	3	4	5	6	7	8
T_u	0.1	0.1	0.1	1	1	1	1
T_v	1	1	0.1	0.1	0.1	0.1	0.1
Modularity	0.19	0.20	0.34	0.28	0.25	0.23	0.21

TABLE 13: Values of T_u and T_v that maximize modularity for different co-clusters of the NG5 dataset.

Dataset	NG5						
#co-clusters	2	3	4	5	6	7	8
T_u	10	10	10	10	10	10	10
T_v	10	10	10	10	10	10	10
Modularity	0.16	0.19	0.18	0.20	0.15	0.15	0.13

TABLE 14: Values of T_u and T_v that maximize modularity for different co-clusters of the ohscal dataset.

Dataset	ohscal						
#co-clusters	7	8	9	10	11	12	13
T_u	1	1	1	1	1	1	1
T_v	10	10	10	10	10	10	10
Modularity	0.22	0.22	0.21	0.30	0.20	0.20	0.20

but the numbers of co-clusters are incorrect, and the ITCC algorithm does not perform well and makes it hard to identify the true meaning of each co-cluster, which indicates the classification quality of these two algorithms is somewhat unsatisfactory.

The two classical fuzzy co-clustering algorithms, FCCI and FCCM, even cannot produce intuitive diagonal co-clusters. However, the *Acc* and *NMI* values in Table 3 reveal that their clustering performances are relatively good on some datasets.

Time complexity is another important index for measuring clustering performance of a clustering algorithm. Theoretically, the time complexity of the MMFCC algorithm is $O(nz \cdot t \cdot C \cdot \log(nz \cdot t \cdot C))$, and ones of other five comparative algorithms, CoClus, Spec, ITCC, FCCM, and FCCI, can be, respectively, expressed as $O(nz \cdot t \cdot C)$, $O(nz \cdot t \cdot N \cdot M \cdot (k+l))$, $O(nz \cdot t \cdot (k+l))$, $O(N \cdot M \cdot t \cdot g)$, and $O(N \cdot M \cdot (N+M) \cdot t \cdot C)$, where nz , t , C , N , M , k , and l , respectively, represent the number of nonzero values, iterations, result clusters, objects, features, row clusters, and column clusters. From the above expressions, we can make a preliminary conclusion that the computation effort of MMFCC is a bit more than CoClus, ITCC, and FCCM, but less than Spec and FCCI algorithms.

In our experiments, we recorded the clustering time of these algorithms running on test sets, and the results are listed in Table 15. We can see that the MMFCC algorithm takes less time than the Spec and FCCI algorithms except on cstr dataset, but slightly more time than the other three algorithms: CoClus, ITCC, and FCCM. The experimental results are consistent with the above theoretical analysis of time complexity.

TABLE 15: Comparison of the time (s) taken by these six algorithms to cluster eleven datasets.

Datasets	Algorithms					
	MMFCC	CoClus	Spec	ITCC	FCCM	FCCI
cstr	0.887	0.5472	0.47	0.3639	0.11	1.324
tr23	1.896	1.1684	4.6157	1.1914	1.432	8.649
tr41	4.992	3.0444	8.9829	3.4159	3.903	16.344
tr45	2.818	2.624	7.8082	1.0474	1.249	41.943
k1b	14.324	6.6895	72.0599	5.7043	11.629	102.234
la2	49.872	12.5328	63.4153	11.3077	33.472	151.459
classic3	5.015	3.8963	6.0345	3.4509	4.774	19.247
classic4	10.467	9.2918	13.6791	7.3142	7.723	47.345
RCV1	60.023	39.2304	263.9268	29.8268	48.798	322.651
NG5	79.294	33.0918	164.108	31.7077	56.984	282.934
ohscal	50.388	30.7594	57.0194	25.6669	44.214	166.357

FIGURE 3: The result graphs obtained by MMFCC and other five comparative algorithms on cstr dataset. (a) MMFCC algorithm. (b) CoClus algorithm. (c) Spec algorithm. (d) ITCC algorithm. (e) FCCI algorithm. (f) FCCM algorithm.

5. Conclusions

In this paper, we propose a novel and efficient fuzzy co-clustering method, named MMFCC, which groups data by maximizing the modularity. This algorithm not only greatly improves the accuracy of the block-diagonal algorithm but also makes up for such shortcomings of the traditional fuzzy co-clustering algorithm as determining the number of final co-clusters.

In addition, the modularity is no longer limited to the application in the field of graph or network. When processing text data, it can also verify the correctness of the original classification numbers of multiple datasets and determine the optimal classification number. It also helps the MMFCC improve the clustering accuracy.

The network and graph datasets are not applied in our experiments, but it is one of the prospective research points to employ our MMFCC algorithm in the field of network and graph. Besides, there are also some significant future research directions, such as combining other clustering methods with fuzzy co-clustering algorithm, adding weights into the fuzzy co-clustering algorithm.

Data Availability

For cstr dataset, please visit <http://www.cs.rochester.edu/trs>. For Classic3 and classic4 datasets, please visit <http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets>. For tr23, tr41, tr45, k1b, la2, and ohscal datasets, please visit <http://glaros.dtc.umn.edu/gkhome/views/cluto>. For RCV1

and NG5 datasets, please visit <http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html>.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the support of Foundation for University Key Teacher by Henan Province under Grant [2015GGJS-068], Fundamental Research Funds for the Universities of Henan Province under Grant [NSFRF1616], Foundation for Scientific and Technological Project of Henan Province under Grant [172102210279], and Foundation for National Natural Science of China under Grants [61872126] and [61433012]. The authors would like to thank members of the IR&DM Research Group from Henan Polytechnic University for their invaluable advice that makes this paper successfully completed.

References

- [1] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, "Spectral biclustering of microarray data: Coclustering genes and conditions," *Genome Research*, vol. 13, no. 4, pp. 703–716, 2003.
- [2] S. F. Hussain and M. Ramazan, "Biclustering of human cancer microarray data using co-similarity based co-clustering," *Expert Systems with Applications*, vol. 55, pp. 520–531, 2016.
- [3] K. Honda, "Collaborative filtering by sequential user-item co-cluster extraction from rectangular relational data," *International Journal of Knowledge Engineering & Soft Data Paradigms*, vol. 2, no. 4, pp. 312–327, 2010.
- [4] K. Honda, A. Notsu, and H. Ichihashi, "Sequential user-item weighted-cluster extraction for collaborative filtering," in *Proceedings of the 2010 World Automation Congress, WAC 2010*, Japan, September 2010.
- [5] F. O. Franca, "Scalable Overlapping Co-clustering of Word-Document Data," in *Proceedings of the 2012 Eleventh International Conference on Machine Learning and Applications (ICMLA)*, pp. 464–467, Boca Raton, FL, USA, December 2012.
- [6] W.-C. Tjhi and L. Chen, "A partitioning based algorithm to fuzzy co-cluster documents and words," *Pattern Recognition Letters*, vol. 27, no. 3, pp. 151–159, 2006.
- [7] Y. Liu, S. Wu, Z. Liu, H. Chao, and Z. Deng, "A fuzzy co-clustering algorithm for biomedical data," *PLoS ONE*, vol. 12, no. 4, p. e0176536, 2017.
- [8] C.-H. Oh, K. Honda, and H. Ichihashi, "Fuzzy clustering for categorical multivariate data," in *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, vol. 4, pp. 2154–2159, July 2001.
- [9] K. Kummamuru, A. Dhawale, and R. Krishnapuram, "Fuzzy co-clustering of documents and keywords," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 772–777, May 2003.
- [10] M. Hanmandlu, O. P. Verma, S. Susan, and V. K. Madasu, "Color segmentation by fuzzy co-clustering of chrominance color features," *Neurocomputing*, vol. 120, pp. 235–249, 2013.
- [11] R. Forsati, H. M. Doustdar, M. Shamsfard, A. Keikha, and M. R. Meybodi, "A fuzzy co-clustering approach for hybrid recommender systems," *International Journal of Hybrid Intelligent Systems*, vol. 10, no. 2, pp. 71–81, 2013.
- [12] M. Rege, M. Dong, and F. Fotouhi, "Co-clustering documents and words using Bipartite Isoperimetric Graph Partitioning," in *Proceedings of the 6th International Conference on Data Mining, ICDM 2006*, pp. 532–541, China, December 2006.
- [13] C. Aykanat, A. Pinar, and V. Ü. Çatalyürek, "Permuting sparse rectangular matrices into block-diagonal form," *SIAM Journal on Scientific Computing*, vol. 25, no. 6, pp. 1860–1879, 2004.
- [14] C. Laclau and M. Nadif, "Diagonal Co-clustering Algorithm for Document-Word Partitioning," in *Advances in Intelligent Data Analysis XIV*, vol. 9385 of *Lecture Notes in Computer Science*, pp. 170–180, Springer International Publishing, Cham, 2015.
- [15] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)*, pp. 269–274, August 2001.
- [16] I. S. Dhillon, S. Mallela, and R. Kumar, "A divisive information-theoretic feature clustering algorithm for text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1265–1287, 2003.
- [17] L. Labiod and M. Nadif, "Co-clustering for binary and categorical data with maximum modularity," in *Proceedings of the 11th IEEE International Conference on Data Mining, ICDM 2011*, pp. 1140–1145, Canada, December 2011.
- [18] M. Ailem, F. Role, and M. Nadif, "Graph modularity maximization as an effective method for co-clustering text data," *Knowledge-Based Systems*, vol. 109, pp. 160–173, 2016.
- [19] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [20] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 2, Article ID 026113, 2004.
- [21] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 70, no. 6, Article ID 066111, 2004.
- [22] S. Karol and V. Mangat, "Evaluation of text document clustering approach based on particle swarm optimization," *Open Computer Science*, vol. 3, no. 2, 2013.