Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 235790, 10 pages
http://dx.doi.org/10.1155/2015/235790

Research Article

A Partitioning Based Algorithm to Fuzzy Tricluster

Yongli Liu, Tengfei Yang, and Lili Fu

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, China

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

Received 17 October 2014; Revised 30 December 2014; Accepted 1 January 2015

Academic Editor: Zhan Shu

Fuzzy clustering allows an object to exist in multiple clusters and represents the affiliation of objects to clusters by memberships. It is extended to fuzzy coclustering by assigning both objects and features membership functions. In this paper we propose a new fuzzy triclustering (FTC) algorithm for automatic categorization of three-dimensional data collections. FTC specifies membership function for each dimension and is able to generate fuzzy clusters simultaneously on three dimensions. Thus FTC divides a three-dimensional cube into many little blocks which should be triclusters with strong coherent bonding among its members. The experimental studies on *MovieLens* demonstrate the strength of FTC in terms of accuracy compared to some recent popular fuzzy clustering and coclustering approaches.

1. Introduction

Clustering is one of the most important techniques of exploratory data mining, used in many applications such as automatic categorization of documents, grouping search engine results, detecting social communities, and others [1, 2]. Clustering algorithms try to detect intrinsic structures of data objects so that a set of clusters is generated where intercluster similarity is maximized and intracluster similarity is minimized.

There is already a large body of work that investigates approaches to clustering objects [3–5]. With different accumulation rules of data objects in clustering and methods employing these rules, clustering algorithms could be divided into four types: hierarchical clustering, partitional clustering, density and grid-based clustering, and others [6]. Hierarchical and partitional clustering methods are used in many applications, and their representative algorithms include agglomerative hierarchical clustering algorithm and K-Means algorithm, respectively. Generally, these algorithms put each object into a single cluster. However, one object may contain multiple subjects and characteristics, so it could belong to multiple clusters [7]. This suggests the appearance of soft clustering algorithms.

Different from hard clustering algorithms where each object belongs to exactly one cluster, soft clustering allows an object to exist in multiple clusters. Introducing fuzziness to clustering gives us the flexible solutions for soft clustering algorithms [8]. Fuzzy clustering extends traditional clustering and represents the affiliation of objects to clusters by memberships. Fuzzy C-Means (FCM) [9] is a representative fuzzy clustering algorithm. Based on this approach, there are many varieties [5, 8]. Nowadays, as most web datasets are known to be large and high dimensional, it becomes more and more challenging to develop satisfactory clustering process. Still, some new clustering algorithms have been designed such as fuzzy coclustering which combines fuzzy clustering with coclustering. Coclustering is a technique for extracting object-feature structures from cooccurrence information of objects and features and is very effective to handle high dimensional data by simultaneously grouping objects and features based on the cooccurrence information [10–12]. Several fuzzy coclustering algorithms have been previously proposed. Oh et al. [8] presented an algorithm of fuzzy clustering for categorical multivariate (FCCM) data. Tjhi and Chen [13] designed a new algorithm fuzzy coclustering with Ruspini's (FCR) condition. In 2013, Hanmandlu et al. [14] proposed a fuzzy coclustering algorithm for images (FCCI).

Besides FCR, Tjhi and Chen introduced two algorithms: one is possibilistic fuzzy coclustering (PFCC) [15]; the other is robust fuzzy coclustering (RFCC) [16]. In 2013, Yan et al. [17] proposed a new heuristic semisupervised fuzzy coclustering algorithm (SS-HFCR) for categorization of large web documents. Keeping the benefits of coclustering and fuzzy clustering, these algorithms improves the representation of overlapping clusters using fuzzy membership function and suitable to categorize documents, particularly web documents in Internet.

Although above clustering algorithms show good clustering quality, they are insufficient in such applications as time-location-type environmental sensor monitoring, time-author-keywords social network analysis, source-destination-text web graph mining. In these applications, data space has three dimensions. None of the existing coclustering algorithms is designed to work in this scenario.

Motivated by the three-dimensional clustering requirements, we propose a novel fuzzy triclustering algorithm, called FTC. The approach combines some popular algorithms such as fuzzy standard clustering and fuzzy coclustering and expands them to support three dimensions.

The remainder of this paper is organized as follows. In Section 2, we provide a literature review of fuzzy standard clustering and coclustering algorithms. Section 3 introduces in detail the proposed FTC. Section 4 presents our experimental results on the *MovieLens* dataset. Finally, we conclude our work.

2. Related Work

In this section, we briefly review two types of clustering algorithms that have been proposed earlier, fuzzy clustering and fuzzy coclustering. These algorithms could help to understand FTC introduced in next section. The explanations on the mathematical notations used in this paper are listed in the Mathematical Notations section.

2.1. Fuzzy Clustering. FCM is the most classical fuzzy clustering algorithm, which is known as the fuzzy version of the K-Means and has been studied by many researchers. FCM aims at minimizing the objective function in (1), subject to the membership constrain in (2). Consider

$$J_{\mathrm{FCM}} = \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci} d\left(x_i, v_c\right) + T_u \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci} \ln u_{ci}, \qquad (1)$$

$$\sum_{c=1}^{C} u_{ci} = 1, \quad \text{for } i = 1, 2, \ldots, N, \qquad (2)$$

where x_i is the ith object, v_c is centroid of the cth cluster, and $d(x_i, v_c)$ measures the distances between objects and clusters. The former item in (1) means that each object should be put into a close cluster, which achieves the purpose of clustering that intercluster similarity is maximized and intracluster similarity is minimized. The latter item controls the fuzziness of clustering, and T_u is just the weighting parameter that specifies the degree of fuzziness.

Lagrange's method of indeterminate multiplier is used to derive the objective function. The solution of this constrained optimization problem can be approximated by Picard iteration and alternating optimization through the following equations (3) and (4), which are the update equations for the object memberships and cluster centroids, respectively, in each iteration. Consider the following:

$$u_{ci} = \frac{\exp\left\{-d\left(x_i, v_c\right)/T_u\right\}}{\sum_{f=1}^{C} \exp\left\{-d\left(x_i, v_f\right)/T_u\right\}}, \qquad (3)$$

$$v_c = \frac{\sum_{i=1}^{N} u_{ci} x_i}{\sum_{i=1}^{N} u_{ci}}. \qquad (4)$$

In FCM, any point x_i has a set of coefficients, u_{ci}, giving the degree of being in the cth cluster, and v_c is the centroid of a cluster which is the mean of all points, weighted by their degrees of belonging to the cluster.

There have been hundreds of FCM-variant algorithms in the literatures that are designed based on the basic principle of FCM, which shows that FCM is very important and still popular in clustering analysis now.

2.2. Fuzzy Coclustering. Coclustering is an important technique in data mining which simultaneously clusters objects and features. Compared to standard clustering, coclustering can offer several benefits [13, 16], including (1) dimensionality reduction, (2) interpretable document cluster, and (3) improvement in accuracy due to local model on clustering.

Fuzzy coclustering extends coclustering by adding fuzzy sets theory into coclusters. Besides the above three benefits, fuzzy coclustering could generate coclusters that are more realistic because one object may contain multiple subjects and one feature may contain multiple concepts.

Due to these advantages, fuzzy coclustering algorithms have been widely studied and developed. Prominent algorithms proposed earlier include FCCM, FCR, FCCI, and PFCC. We briefly review FCCM and FCCI in this paper which are theoretical principles of FTC.

FCCM tries to maximize an objective function defined in (5) to complete coclustering, subject to the membership constraints in (6) and (7). Consider the following:

$$\begin{aligned} J_{\mathrm{FCCM}} = {} & \sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{j=1}^{K} u_{ci} v_{cj} d_{ij} \\ & - T_u \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci} \ln u_{ci} - T_v \sum_{c=1}^{C}\sum_{j=1}^{K} v_{cj} \ln v_{cj}, \end{aligned} \qquad (5)$$

$$\sum_{c=1}^{C} u_{ci} = 1, \quad \text{for } i = 1, 2, \ldots, N, \qquad (6)$$

$$\sum_{j=1}^{K} v_{cj} = 1, \quad \text{for } c = 1, 2, \ldots, C. \qquad (7)$$

In (5), there are three terms. The first term is the degree of aggregation which tries to make highly interrelated objects and features. The u_{ci} and v_{cj} are two membership functions, indicating object membership and feature membership, respectively. The second and the third terms control the fuzziness of clustering, where T_u and T_v are two parameters that adjust the levels of fuzziness of object and feature memberships, respectively.

By applying Lagrange multiplier method, the solution of this constrained optimization problem can be approximated by iteratively updating the following equation in an alternating fashion:

$$u_{ci} = \frac{\exp\left\{\sum_{j=1}^{K} v_{cj} d_{ij}/T_u\right\}}{\sum_{f=1}^{C} \exp\left\{\sum_{j=1}^{K} v_{fj} d_{ij}/T_u\right\}},$$
$$v_{cj} = \frac{\exp\left\{\sum_{i=1}^{N} u_{ci} d_{ij}/T_v\right\}}{\sum_{q=1}^{K} \exp\left\{\sum_{i=1}^{N} u_{ci} d_{iq}/T_v\right\}}. \tag{8}$$

To understand fuzzy coclustering easily, Oh et al. gave a numerical example. They applied FCCM to literature retrieval data set which shows the cooccurrence relations among the literatures and the key words. The rows represent the literatures and columns are the key words. The results of FCCM were shown as two tables. The first table listed the memberships of literatures, that is, the values of u_{ci}. And the second table showed the memberships of key words (values of v_{cj}). Larger memberships mean literatures and key words are more likely to belong to a cluster; thus the final clusters could be generated based on the two tables.

Besides FCCM, FCCI is a novel color segmentation technique using fuzzy coclustering approach where both the objects and features are assigned membership functions. In FCCI, the following objective function equation (9) is minimized subject to the membership constraints in (10) and (11). Consider the following:

$$J_{\text{FCCI}} = \sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{j=1}^{K} u_{ci} v_{cj} D_{cij} + T_u \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci} \ln u_{ci} + T_v \sum_{c=1}^{C}\sum_{j=1}^{K} v_{cj} \ln v_{cj}, \tag{9}$$

$$\sum_{c=1}^{C} u_{ci} = 1, \quad \text{for } i = 1, 2, \ldots, N, \tag{10}$$

$$\sum_{j=1}^{K} v_{cj} = 1, \quad \text{for } c = 1, 2, \ldots, C. \tag{11}$$

The first term in (9) shows the distance relationship. In this term, $D_{cij} = (d_{ij} - p_{cj})^2$, which means the crucial distance equals the square of Euclidean distance between feature data point d_{ij} and the feature cluster centroid p_{cj}. The minimization of this term assigns to an object a higher membership value taking into account the feature cluster center it is closest to and which is more relevant than other features for that particular cluster. The second and third terms in (9) contribute to the fuzziness in the resulting clusters, where T_u and T_v are the weighting parameters specifying the degree of fuzziness.

The solution of this constrained optimization problem can be approximated by iteratively updating following equations in an alternating fashion:

$$u_{ci} = \frac{\exp\left\{-\sum_{j=1}^{K} v_{cj} D_{cij}/T_u\right\}}{\sum_{f=1}^{C} \exp\left\{-\sum_{j=1}^{K} v_{fj} D_{cij}/T_u\right\}},$$
$$v_{cj} = \frac{\exp\left\{-\sum_{i=1}^{N} u_{ci} D_{cij}/T_v\right\}}{\sum_{q=1}^{K} \exp\left\{-\sum_{i=1}^{N} u_{ci} D_{ciq}/T_v\right\}}, \tag{12}$$
$$p_{cj} = \frac{\sum_{i=1}^{N} u_{ci} x_{ij}}{\sum_{i=1}^{N} u_{ci}}.$$

3. Proposed FTC Algorithm

Although experimental results show that clustering quality of above clustering algorithms is encouraging, they are insufficient in some scenarios. Here there are two examples.

Scenario 1. Users submit queries to search engines and obtain search results. When grouping users, we generally employ the user-document matrix and neglect queries submitted. Sometimes the query information is actually useful. A user selects one document just because he submits a relative query before. Thus it becomes an issue of conditional probability and users, queries, and documents construct a three-dimensional space.

Scenario 2. We could model user interests by mining documents clicked. However, interests of a user may change over time. If we analyze all documents he browsed without regard to time, user interest drift could not be mined. Therefore, time is an important factor. When we take account of time factor, the data format becomes a user-document-time three-dimensional matrix from a user-document contingency table.

In these scenarios, we have data spaces with three dimensions, and the data format provided to us is not a contingency table with data object and feature, but a three-dimensional matrix. Current clustering algorithms primarily aim at contingency table, and few concern three-dimensional data. Gnatyshak et al. [18] proposed two novel methods for biclustering and triclustering data collected from online social networks, and they could reveal users' interests as tags and use them to describe Vkontakte groups with triclustering. In the context of two relational datasets that share labels among one of the dimensions, Mahiskar et al. [19] simultaneously process two datasets to unveil triclusters and presented a triclustering algorithm that searches for meaningful combinations of biclusters in two related datasets. Guigourès et al. [20] introduced a novel technique to track structures in time evolving graphs, based on a parameter free approach for three-dimensional coclustering of the source vertices, the target vertices, and the time. Above algorithms implement triclustering process without fuzzy sets theory, which means they concentrate on hard clustering.

We propose a novel fuzzy triclustering algorithm, FTC, in this paper. In FTC, we target to keep the benefits of coclustering/triclustering and fuzzy clustering simultaneously. As indicated in Figure 1, the new algorithm aims at essentially achieving fuzzy three-dimensional clustering (as Figure 1(e)). Figures 1(a) and 1(b) refer to hard clustering and hard coclustering, respectively. Figures 1(c) and 1(d) both indicate fuzzy coclustering. The latter is more flexible than the former because each cluster can contains part elements of some rows and columns in the later mode. We enlarge the issue of Figure 1(d) from two dimensions to three and in turn extend fuzzy coclustering algorithms to fuzzy triclustering (FTC) algorithm.

The goal of FTC is to maximize the objective function in (13), subject to the following constraints in (14), (15), and (16). Consider the following:

$$\begin{aligned} J_{\mathrm{FTC}} = & \sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{k=1}^{M} u_{ci}v_{cj}w_{ck}d_{ijk} \\ & - T_u\sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci}\ln u_{ci} - T_v\sum_{c=1}^{C}\sum_{j=1}^{K} v_{cj}\ln v_{cj} \\ & - T_w\sum_{c=1}^{C}\sum_{k=1}^{M} w_{ck}\ln w_{ck}, \end{aligned} \tag{13}$$

$$\sum_{c=1}^{C} u_{ci} = 1, \quad \text{for } i = 1, 2, \dots, N, \tag{14}$$

$$\sum_{j=1}^{K} v_{cj} = 1, \quad \text{for } c = 1, 2, \dots, C, \tag{15}$$

$$\sum_{k=1}^{M} w_{ck} = 1, \quad \text{for } c = 1, 2, \dots, C. \tag{16}$$

The first term in (13) is the degree of aggregation that should be maximized among the triclustering, which intends to make highly related u-q-d in Figure 1(e) to be triclustered together. In other words, the cube in Figure 1(e) will be divided into many little blocks which should be triclusters with strong coherent bonding among its members (i.e., u, q, and d). u_{ci}, v_{cj}, and w_{ck} are three membership functions, indicating memberships of three dimensions, respectively. This term is similar to the first term in objective function of FCCM which is denoted by the component-wise inner product of two matrices. The second, third, and fourth terms are entropy regularization factors that combines all u_{ci}'s, v_{cj}'s, and w_{ck}'s separately. They control the degree of fuzziness in final clusters, where T_u, T_v, and T_w are weighting parameters.

The constrained optimization of FTC can be solved by applying the Lagrange multipliers α, β, and γ to constraints in (14), (15), and (16), respectively. Consider the following:

$$\begin{aligned} J'_{\mathrm{FTC}} = & \sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{k=1}^{M} u_{ci}v_{cj}w_{ck}d_{ijk} \\ & - T_u\sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci}\ln u_{ci} - T_v\sum_{c=1}^{C}\sum_{j=1}^{K} v_{cj}\ln v_{cj} \\ & - T_w\sum_{c=1}^{C}\sum_{k=1}^{M} w_{ck}\ln w_{ck} + \sum_{i=1}^{N}\alpha_i\left(\sum_{c=1}^{C}u_{ci} - 1\right) \\ & + \sum_{c=1}^{C}\beta_c\left(\sum_{j=1}^{K}v_{cj} - 1\right) + \sum_{c=1}^{C}\gamma_c\left(\sum_{k=1}^{M}w_{ck} - 1\right). \end{aligned} \tag{17}$$

Taking the partial derivative of J'_{FTC} in (17) with respect to U, V, and W, respectively, and setting the gradient to zero we have

$$\begin{aligned} \frac{\partial J'_{\mathrm{FTC}}}{\partial U} &= \sum_{j=1}^{K}\sum_{k=1}^{M} v_{cj}w_{ck}d_{ijk} - T_u\left(1 + \ln u_{ci}\right) + \alpha_i = 0, \\ \frac{\partial J'_{\mathrm{FTC}}}{\partial V} &= \sum_{i=1}^{N}\sum_{k=1}^{M} u_{ci}w_{ck}d_{ijk} - T_v\left(1 + \ln v_{cj}\right) + \beta_c = 0, \\ \frac{\partial J'_{\mathrm{FTC}}}{\partial W} &= \sum_{i=1}^{N}\sum_{j=1}^{K} u_{ci}v_{cj}d_{ijk} - T_w\left(1 + \ln w_{ck}\right) + \gamma_c = 0. \end{aligned} \tag{18}$$

Solving above equations yield the formulae for u_{ci}, v_{cj}, and w_{ck} as

$$u_{ci} = \frac{\exp\left\{\sum_{j=1}^{K}\sum_{k=1}^{M} v_{cj}w_{ck}d_{ijk}/T_u\right\}}{\sum_{c=1}^{C}\exp\left\{\sum_{j=1}^{K}\sum_{k=1}^{M} v_{cj}w_{ck}d_{ijk}/T_u\right\}}, \tag{19}$$

$$v_{cj} = \frac{\exp\left\{\sum_{i=1}^{N}\sum_{k=1}^{M} u_{ci}w_{ck}d_{ijk}/T_v\right\}}{\sum_{j=1}^{K}\exp\left\{\sum_{i=1}^{N}\sum_{k=1}^{M} u_{ci}w_{ck}d_{ijk}/T_v\right\}}, \tag{20}$$

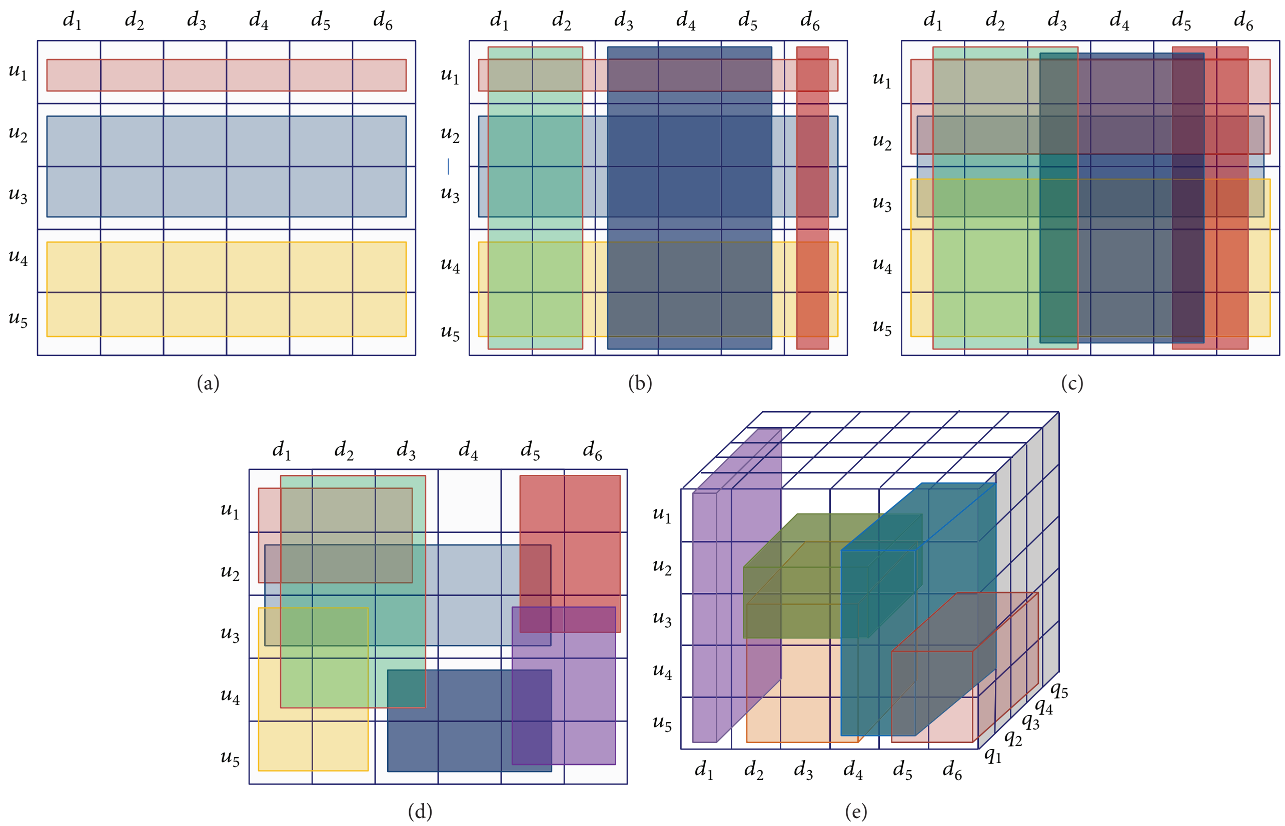
$$w_{ck} = \frac{\exp\left\{\sum_{i=1}^{N}\sum_{j=1}^{K} u_{ci}v_{cj}d_{ijk}/T_w\right\}}{\sum_{k=1}^{M}\exp\left\{\sum_{i=1}^{N}\sum_{j=1}^{K} u_{ci}v_{cj}d_{ijk}/T_w\right\}}. \tag{21}$$

Equations (19), (20), and (21) are the update equations for the three dimensions memberships, through which the solution of the constrained optimization problem in (17) can be approximated by Picard iteration. Therefore FTC can be written as Pseudocode 1.

The pseudocode of FTC in Pseudocode 1 shows that the time complexity of FTC is $O(CNKM\tau)$, where τ denotes the number of iterations. Its time complexity is higher than such fuzzy coclustering algorithms as FCM, FCCM with $O(CNK\tau)$. The main reason is that FTC generates fuzzy clusters simultaneously on three dimensions while other algorithms on two dimensions.

4. Experiments

In order to test the effectiveness of FTC, we carried out a set of experiments in this paper. The results are also compared with *five* well received approaches in the literature, K-Means, FCM, FCCM, RFCC, and FCCI. Of the five algorithms, K-Means is a traditional standard clustering algorithm, FCM is a fuzzy standard clustering algorithm, and other three are fuzzy coclustering algorithms.



(a) (b) (c) (d) (e)

Figure 1: Different clustering strategies.

```
Pseudocode of FTC algorithm
Set values of parameters $C$, $T_u$, $T_v$, $T_w$, maximum error limit $\xi$ and the maximum number of iterations parameter $\tau_{\max}$
Set $\tau = 1$
Initialize memberships $u_{ci}$ and $v_{cj}$ randomly
REPEAT
    Update membership $w_{ck}$ using (21)
    Update membership $v_{cj}$ using (20)
    Update membership $u_{ci}$ using (19)
    Set $\tau + +$
UNTIL $\max(|u_{ci}(\tau) - u_{ci}(\tau - 1)|) \le \xi$ or $\tau = \tau_{\max}$
```

Pseudocode 1: FTC pseudocode.

4.1. Experimental Setup. We employ the *MovieLens* dataset [21] to evaluate the performance of FTC in categorizing real-world data. This dataset contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 *MovieLens* users who joined *MovieLens* in 2000. The *user-movie* matrix is very sparse, so we construct two dense subsets from *MovieLens*. The first subset contains top 100 users who give the most times of ratings, and top 100 movies that are evaluated most often. In addition, by counting genres of all movies and the top 100 ones, we observe that some genres of movies are popular and evaluated most. The concise movie genre information (as Table 1) is analyzed, and *drama*, *comedy*, *action*, *thriller*, and *romance* five genres are added into the first subset to help accomplish experiments. Therefore, this subset is a 100 × 100 × 5 matrix. The second subset is larger, which has top 500 users, top 500 movies, and all 18 genres. Based on the *user-movie-genre* three-dimensional matrices, we implemented our experiments.

Although we could get *movie* clusters based on coclustering or even *genre* clusters based on triclustering, the absence of the ground-truth *genre* categorization makes it difficult to evaluate the accuracies of the resulting *genre* clusters. We mainly intend to get the *user* and *movie* clusters by above six algorithms, whose performance will be evaluated using the following evaluation measures.

4.2. Evaluation Criteria. There are several ways for numerically scoring the cluster quality, such as Entropy, *F*-Measure, and Overall Similarity. We choose *F*-Measure and Entropy as the criteria to evaluate the performance of FTC.

F-Measure is the weighted harmonic mean of precision and recall, and it is often used to measure clustering quality. The higher the value of *F*-Measure is, the better the clustering quality is. For standard datasets, class information is known. We get clusters after clustering. When objects in class i is grouped into cluster j, the *F*-Measure value is given by

$$F(i,j) = \frac{2 * \text{precision}(i,j) * \text{recall}(i,j)}{\text{precision}(i,j) + \text{recall}(i,j)}, \qquad (22)$$

Table 1: Movie genres information.

Movie genres	# movies	# movies top 100
Film-Noir	44	1
Crime	211	4
Romance	471	12
Animation	105	4
Horror	343	9
Comedy	1200	28
Mystery	106	2
Western	68	3
Action	503	9
Adventure	283	5
War	143	4
Fantasy	68	1
Children's	251	7
Musical	114	2
Sci-Fi	276	6
Thriller	492	17
Drama	1603	36
Documentary	127	7

where precision(i, j) and recall(i, j) are computed using the following equations, respectively:

$$\text{recall}\left(i, j\right) = \frac{n_{ij}}{n_i}, \quad (23)$$

$$\text{precision}\left(i, j\right) = \frac{n_{ij}}{n_j}, \quad (24)$$

where n_{ij} is the number of members of class i in cluster j, n_j is the number of members of cluster j, and n_i is the number of members of class i. The overall F-Measure value is given by the following:

$$F = \sum_i \frac{n_i}{n} * \max\left\{F\left(i, j\right)\right\}, \quad (25)$$

where n is the total number of documents.

Essentially, Entropy is designed for measuring the randomness of molecules in a thermodynamics system. It can also be used to evaluate cluster distribution during clustering in information theory. If documents are distributed uniformly and there are little differences between clusters, the value of Entropy will be high. On the contrary, if there are great differences between clusters, the value of Entropy will be low. The purpose of clustering is to enlarge the differences between the clusters, so the lower the value of Entropy is, the higher the clustering quality will be. The expression for Entropy of the whole clustering result is listed as follows:

$$F_{cs} = \sum_{j=1}^{m} \frac{n_j E_j}{n}, \quad (26)$$

where E_{cs} is the whole Entropy value, n_j is the number of documents in cluster j, n is the number of all the documents, m is the number of clusters, and E_j is the Entropy value of cluster j, which is calculated using the following formula:

$$E_j = -\sum_i p_{ij} \log p_{ij}, \quad (27)$$

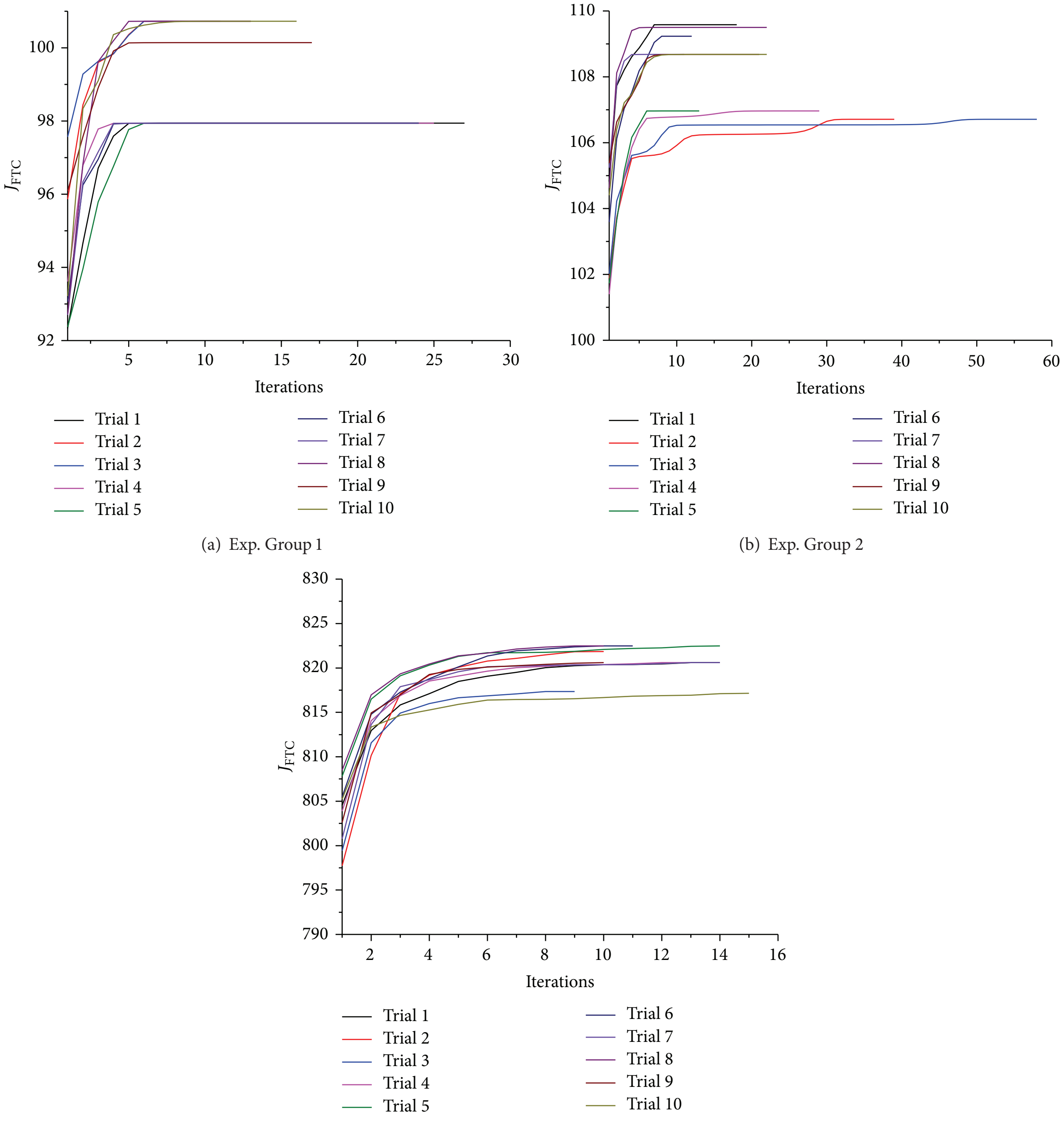
where p_{ij} is the probability that one document belonging to class i could be put into cluster j during the partition.

4.3. Experimental Results. We divided the first subset into 2 classes in terms of user gender and 3 classes in terms of user age where Class 1 includes users younger than 24, Class 2 includes users whose age ranges from 25 to 49, and Class 3 includes users older than 50. The second subset is divided into 7 classes according to user age that correspond to 7 age ranges tagged with 1, 18, 25, 35, 45, 50, and 56, respectively, in *MovieLens*. We conducted three groups of experiments on the subsets of which two groups of experiments divide the first subset into 2 and 3 clusters, respectively, and the third group classifies the second subset into 7 clusters.

Before discussing the performance comparisons, we inspect the value of objective function. Since FTC tries to maximize the objective function in (13), we run a 10-trial simulation on the two subsets and choose the one that resulted in the best performance. For all the experiments, we set $T_v = 1$, $T_w = 1$, $T_u = 0.5$, maximum number of iterations $\tau_{\max} = 50$, and convergence indicator $\xi = 0.0001$. Figure 2 displays the changing trend of objective function with updating iteratively.

Figure 2(a) shows value of objective function in the first group of experiments that divide the first subset into 2 clusters, Figure 2(b) describes the second group dividing the first subset into 3 clusters, and Figure 2(c) illustrates the third group of experiments dividing the second subset into 7 categories. In Figure 2, different trials converge at different local maxima. The second, third, eighth, and tenth trials result in the best performance in Figure 2(a), the first trial results in the best performance in Figure 2(b), and fifth, sixth, and eighth trials result in the best performance in Figure 2(c). It can also be seen from Figure 2 that each trial almost reaches local maxima when τ is less than 10, which shows that FTC has the rapid constringency speed of approaching local maxima.

The clustering performance comparisons of K-Means, FCM, FCCM, RFCC, FCCI, and FTC are illustrated in Figure 3. On the three groups of experiments, FTC achieves either better or comparable performance to its partner algorithms, which strengthen our argument that the proposed fuzzy triclustering algorithm, FTC, can generally perform clustering better than such existing formulations as fuzzy clustering and fuzzy coclustering. K-Means is a hard clustering algorithm that just implements clustering on one dimension. In the first group of experiments where the values of N and C are both low, K-Means achieve higher clustering quality. When the number of clusters is greater, the performance of K-Means drops dramatically and is gradually poorer than fuzzy algorithms. Of the five fuzzy algorithms, except in first group of experiments, FCM is shown to consistently result in poor clustering quality, which further



(a) Exp. Group 1

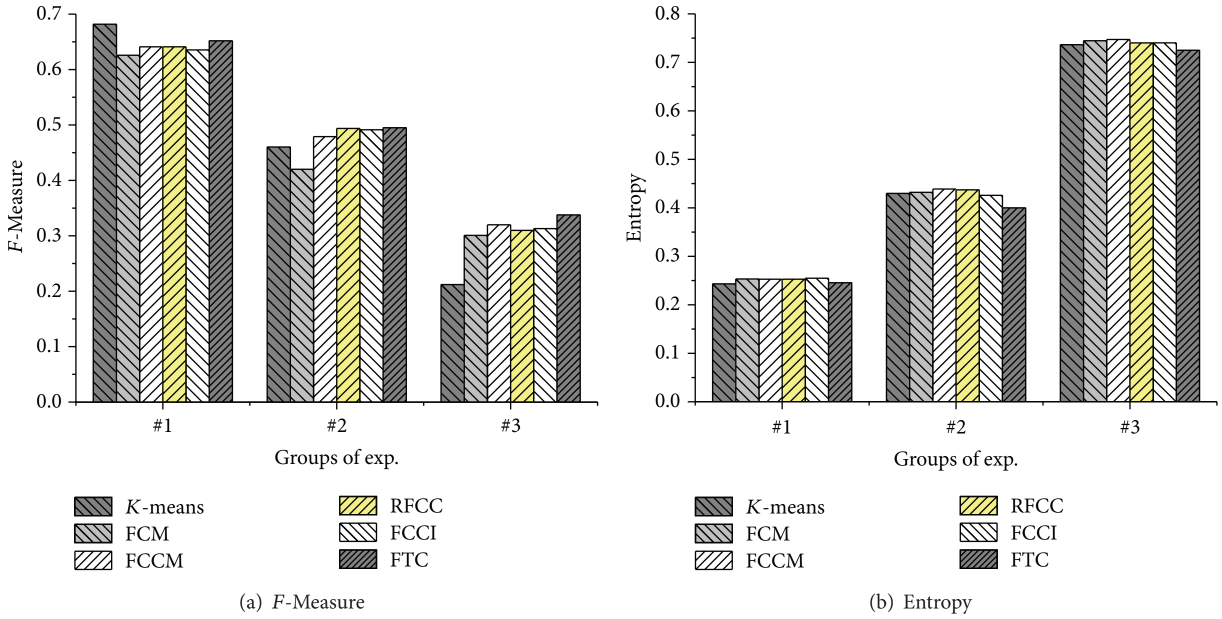
(b) Exp. Group 2

(c) Exp. Group 3

Figure 2: Objective function values during optimization.

confirms that coclustering could achieve better performance than standard clustering. Another thing to highlight from Figure 3 is that FCCM, RFCC, and FCCI are competitive, and they can generate clusters with relatively high quality.

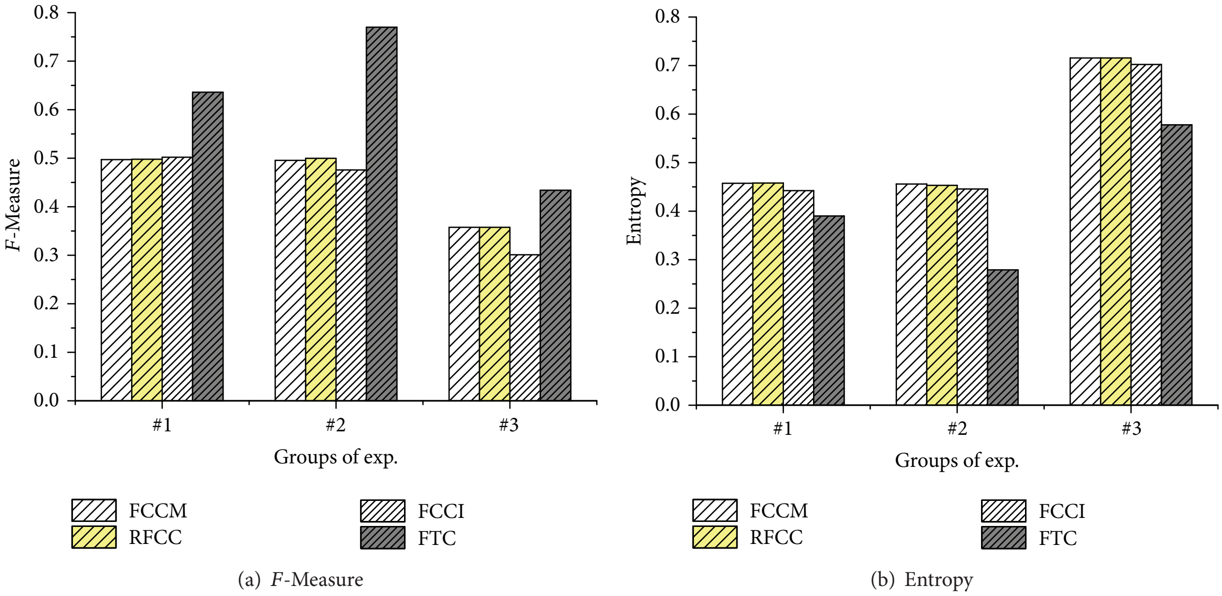
Besides clustering *users*, coclustering algorithms and FTC also simultaneously generate fuzzy *movie* and even *genre* clusters. We have *genre* information of movies as conditions classifying movies and focus on evaluating quality of fuzzy movie clusters. The above experiments are investigated and experimental results about *movie* clusters are illustrated in Figure 4. K-Means and FCM cannot generate cluster in *movie* dimension; thus they do not participate in this comparison. Figure 4 shows that FTC achieves the best performance among the four algorithms, and moreover, the improvement in terms of F-Measure is significant. Particularly in the second group of experiments, the F-Measure value of FTC equals 0.77, and the values of FCCM, RFCC, and FCCI are 0.50, 0.50, and 0.48, respectively. The Entropy values of the FTC, FCCM, RFCC, and FCCI in this group experiments are 0.28, 0.46, 0.45, and 0.44, respectively.



(a) *F*-Measure

(b) Entropy

Figure 3: Clustering performance comparisons of FCM, FCCM, RFCC, FCCI, and FTC.



(a) *F*-Measure

(b) Entropy

Figure 4: Clustering performance comparisons of FCCM, RFCC, FCCI, and FTC.

In our experiments, we observe that some algorithms are sensitive to the value of parameter T_u. Different values of T_u may generate different results. We extend the above three groups of experiments and investigate the effects of T_u. In the experiments, the parameter T_u ranges from 0.1 to 1.5, the clustering performance is illustrated in Figure 5. It indicates that FTC is not so sensitive to the value of parameter T_u. The *F*-Measure value varies only slightly with the value of T_u changing.

We have analyzed the computation complexity of FTC above. The time complexity of FTC is $O(CNKM\tau)$, where τ denotes the number of iterations, and the time complexities of such fuzzy coclustering algorithms as FCM and FCCM are all $O(CNK\tau)$. FTC is more time-consuming because FCCM, RFCC, and FCCI group objects simultaneously on two dimensions; however FTC objects on three dimensions whose clustering process takes into account one more dimension. Table 2 shows the actual runtime of the six algorithms in above experiments. It can be seen that sometimes FTC is more timesaving than other fuzzy coclustering algorithms; though its time complexity is higher theoretically. In the first and second groups of experiments, FTC generates clusters

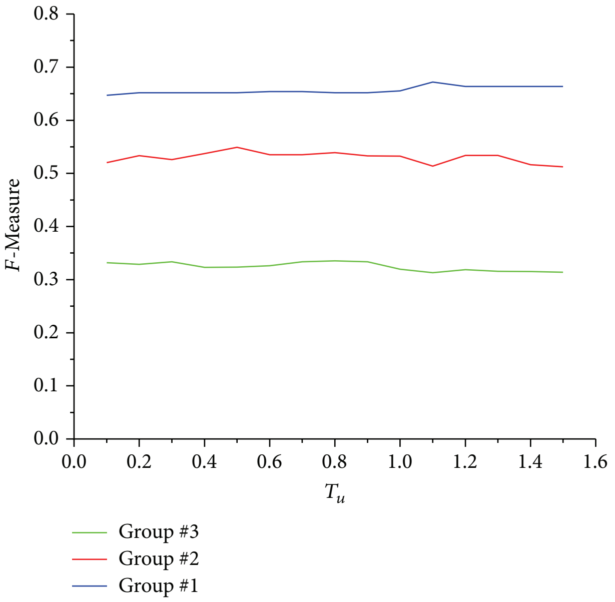


FIGURE 5: The effects of T_u change.

TABLE 2: The comparison of clustering time (ms).

Exp. groups	K-Means	FCM	FCCM	RFCC	FCCI	FTC
# 1	54	11	154	151	89	80
# 2	90	32	287	350	112	235
# 3	1979	997	2630	3534	940	16239

in 80 and 235 ms, respectively; however FCCM generates clusters in 154 and 287 ms and RFCC generates clusters in 151 and 350 ms. Even in the third group of experiments, clustering time of FTC is less than M times of such coclustering algorithms as FCCM and FCCI. In addition, clustering time of FTC is manageable by adjusting the value of τ. Therefore, FTC may complete clustering process more rapidly in practice than in theory.

5. Conclusion

Recently, some new clustering algorithms have been proposed, such as fuzzy clustering and fuzzy coclustering. They still focus primarily on contingency tables with two dimensions and are insufficient in three-dimensional scenarios.

In this paper, a novel fuzzy triclustering approach, FTC, based on simultaneous clustering of three-dimensional memberships is proposed. Under the proposed framework, a new objective function and the update rules are formulated. FTC offers numerous benefits, such as dimensionality reduction, interpretable clusters, and overlapping clusters, deriving from (fuzzy) coclustering. We implement three groups of experiments to evaluate the performance of FTC on *MovieLens* subsets, and it is also compared with some popular fuzzy (co)clustering algorithms and proves to outperform them.

It is challenging to determine the number of clusters in the literature. In our study, the value of C is still specified by users manually, which determines that FTC is not unsupervised absolutely. In the future, we intend to incorporate techniques evaluating the number of clusters to extend our approach.

Mathematical Notations

C, N, K, and M:	Numbers of (co)clusters, objects, and features (K and M)
u_{ci}:	Fuzzy object partitioning membership
v_{cj}, w_{ck}:	Fuzzy feature partitioning membership
d_{ij}:	Relatedness measure between an object and a feature
T_u, T_v, and T_w:	Coclustering user-defined parameters
τ:	Number of iterations
$\tau_{\max}$:	Maximum number of iterations parameter
ξ:	Convergence indicator.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was financially supported by the NSFC Fund (no. 61202286) and Special Fund for Fast Sharing of Science Paper in Net Era by CSTD (no. 2013117).

References

[1] Y. Liu, Y. Ouyang, and Z. Xiong, "Incremental clustering using information bottleneck theory," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 5, pp. 695–712, 2011.

[2] K. M. Hammouda and M. S. Kamel, "Efficient phrase-based document indexing for web document clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 10, pp. 1279–1296, 2004.

[3] M. A. Rahman and M. Z. Islam, "A hybrid clustering technique combining a novel genetic algorithm with K-means," *Knowledge-Based Systems*, vol. 71, pp. 345–365, 2014.

[4] E. Rashedi, A. Mirzaei, and M. Rahmati, "An information theoretic approach to hierarchical clustering combination," *Neurocomputing*, vol. 148, pp. 487–497, 2015.

[5] S. Wikaisuksakul, "A multi-objective genetic algorithm with fuzzy c-means for automatic data clustering," *Applied Soft Computing*, vol. 24, pp. 679–691, 2014.

[6] Y. Liu, Q. Guo, L. Yang, and Y. Li, "Research on incremental clustering," in *Proceedings of the 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet '12)*, pp. 2803–2806, Yichang, China, April 2012.

[7] K. Honda, C.-H. Oh, Y. Matsumoto, A. Notsu, and H. Ichihashi, "Exclusive partition in FCM-type co-clustering and its application to collaborative filtering," *International Journal of Computer Science and Network Security*, vol. 12, no. 12, pp. 52–58, 2012.

[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] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, NY, USA, 1981.

[10] N. Gupta and S. Aggarwal, "MIB: using mutual information for biclustering gene expression data," *Pattern Recognition*, vol. 43, no. 8, pp. 2692–2697, 2010.

[11] B. Hanczar and M. Nadif, "Ensemble methods for biclustering tasks," *Pattern Recognition*, vol. 45, no. 11, pp. 3938–3949, 2012.

[12] L. Han and H. Yan, "BSN: an automatic generation algorithm of social network data," *Journal of Systems and Software*, vol. 84, no. 8, pp. 1261–1269, 2011.

[13] 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.

[14] 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.

[15] W.-C. Tjhi and L. Chen, "Possibilistic fuzzy co-clustering of large document collections," *Pattern Recognition*, vol. 40, no. 12, pp. 3452–3466, 2007.

[16] W.-C. Tjhi and L. Chen, "Robust fuzzy co-clustering algorithm," in *Proceedings of the 6th International Conference on Information, Communications and Signal Processing (ICICS '07)*, pp. 1–5, December 2007.

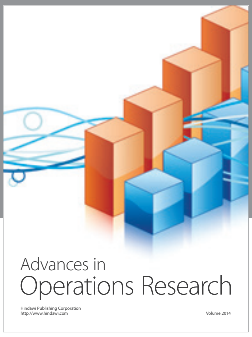
[17] Y. Yan, L. Chen, and W.-C. Tjhi, "Fuzzy semi-supervised co-clustering for text documents," *Fuzzy Sets and Systems*, vol. 215, pp. 74–89, 2013.

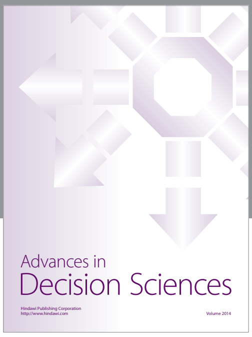
[18] D. Gnatyshak, D. I. Ignatov, A. Semenov, and J. Poelmans, "Analysing online social network data with biclustering and triclustering," in *Proceedings of the 2nd International Workshop on Concept Discovery in Unstructured Data (CDUD '12)*, pp. 30–39, May 2012.

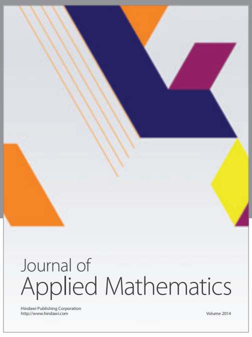
[19] P. S. Mahiskar, A. W. Bhade, and P. N. Chatur, "An effective triclustering algorithm for mining real datasets: review paper," *International Journal Computer Technology & Applications*, vol. 3, no. 1, pp. 352–355, 2012.

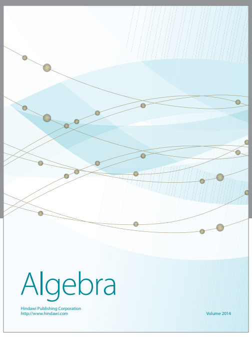
[20] R. Guigourès, M. Boullé, and F. Rossi, "A triclustering approach for time evolving graphs," in *Proceedings of the IEEE 12th International Conference on Data Mining Workshops (ICDMW '12)*, pp. 115–122, Brussels, Belgium, December 2012.

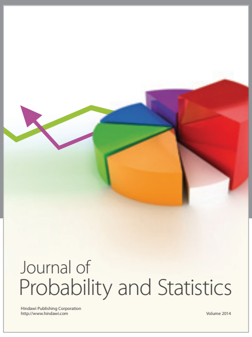
[21] MovieLens, http://grouplens.org/datasets/movielens/.


Advances in
Operations Research

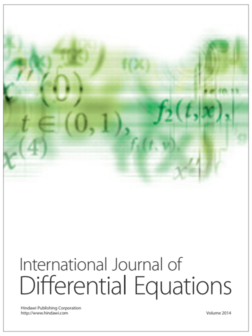

Advances in
Decision Sciences


Journal of
Applied Mathematics

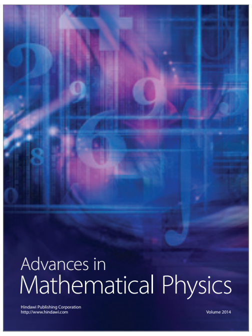

Algebra

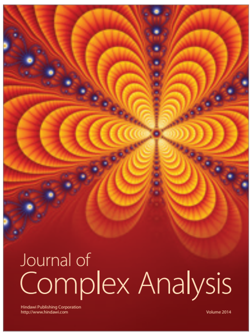

Journal of
Probability and Statistics

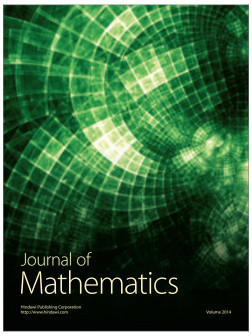
The Scientific
World Journal

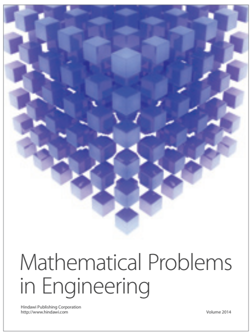

International Journal of
Differential Equations

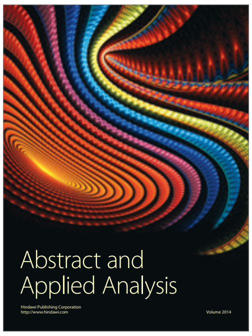
International Journal of
Combinatorics

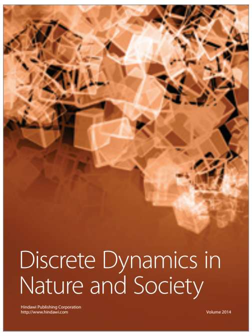

Advances in
Mathematical Physics

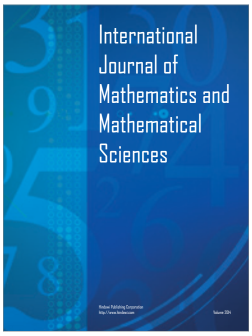

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

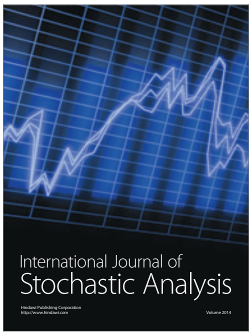

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization