Treerattanapitak K, Jaruskulchai C. Exponential fuzzy C-means for collaborative filtering. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 27(3): 567–576 May 2012. DOI 10.1007/s11390-012-1244-x

Exponential Fuzzy C-Means for Collaborative Filtering

Kiatichai Treerattanapitak and Chuleerat Jaruskulchai

Department of Computer Science, Kasetsart University, 50 Ngamwongwan Rd., Jatuchak, Bangkok, Thailand

E-mail: kiat@smartzap.com; fscichj@ku.ac.th

Received September 1, 2011; revised March 1, 2012.

Abstract Collaborative filtering (CF) is one of the most popular techniques behind the success of recommendation system. It predicts the interest of users by collecting information from past users who have the same opinions. The most popular approaches used in CF research area are Matrix factorization methods such as SVD. However, many well-known recommendation systems do not use this method but still stick with Neighborhood models because of simplicity and explainability. There are some concerns that limit neighborhood models to achieve higher prediction accuracy. To address these concerns, we propose a new exponential fuzzy clustering (XFCM) algorithm by reformulating the clustering's objective function with an exponential equation in order to improve the method for membership assignment. The proposed method assigns data to the clusters by aggressively excluding irrelevant data, which is better than other fuzzy C-means (FCM) variants. The experiments show that XFCM-based CF improved 6.9% over item-based method and 3.0% over SVD in terms of mean absolute error for 100 K and 1 M MovieLens dataset.

Keywords fuzzy clustering, recommendation system, degree of membership, neighbor selection

1 Introduction

The recommendation system is a powerful tool to help guide online shoppers to choose the product that best fulfills their needs. The system is widely implemented in many online stores such as Netflix, eBay, Amazon, but the demand for a better system is endless.

Collaborative filtering (CF) is one of the reasons behind the success of the recommendation system and is one of the most popular techniques because of its simplicity and ease of use. CF predicts the interest of a user by collecting and using information from past users who have the same opinions. The most common approach for CF is based on neighborhood models that attempt to provide recommendation by either user-user approach[1] or item-item approach[2-4]. User-user based methods predict user rating from users with similar preferences and item-item based methods predict user rating from ratings made by the same user on similar items. Item-item based method is more favorable due to users are more familiar with items previously preferred by them rather than other users with the same preferences[5-6]. In general, matrix factorization methods[6-9] are the most popular in the research area. These methods try to map large users and items matrix in the lower dimensions using latent factor. They generate prediction more accurately than neighborhood models because these methods optimize prediction based on global ratings while neighborhood models compute ratings based on local neighbors[6]. However, many well-known recommendation systems like Amazon[4], TiVo[10] stick with neighborhood models due to their simplicity and ease of use. There are more important reasons why these methods are preferred. First, recommendations generated from neighborhood models are explainable which are often useful to enhance user experiences beyond the improvement of accuracy. Second, they can provide immediate recommendation when new ratings enter to the system[6]. However, the quality of recommendation for neighborhood models is relied on fundamental characteristics of the data which are:

Sparsity. Users do not always rate all products even if they are very active users. Ratings are made only to products previously used by them and therefore profile vectors consequently contain a lot of missing values.

Scalability. Users-items matrices are always large. The cost to find neighboring in a large matrix is expensive. It grows non-linearly to the number of users and items.

Cold Start. This happens when the system attempts to make recommendations for a new user or offer a new item to a user. In both cases, the new vector profiles cannot be paired with existing data in the database since they are empty.

Regular Paper

More importantly, the recommendation quality of neighborhood models can be improved according to some concerns raised by Koren *et al.*[6] First, the similarity functions used in neighborhood models is the score between a pair of users or items. Since neighborhood models generate recommendation based on subset of neighbors, similarity should represent the same degree of similarity among them. Second, neighborhood models do not account for interaction among neighbors. Similarity between an item and its neighbors is computed independently. Thus, some related items such as The Lord of the Rings 1∼3 may be triple counting. Third, overfitted ratings occur from the interpolation weights that sum to 1. In case of the useful neighbors the item has less than the specified k nearest neighbors, the neighborhood models force to find the first k nearest neighbors which may not be related to the item. This concern is fixed by adding penalty term into prediction equation[11]. Fourth, neighborhood models may not work well if ratings differ among neighbors. For example, two users rate three items by (1, 2, 3) and (3, 4, 5). The third item is obviously more preferred than the second and the first items by both users. However, it is difficult for neighborhood models to capture this kind of user preferences. The item ranking models are different approaches that model the recommendation directly to user preferences. Item ranking methods do not generate the ratings but rank the preferred items as output[12].

In order to achieve higher accuracy, Koren *et al.*[5-6] proposed jointly derived neighborhood interpolation weights to estimate interpolation weights based on least square method. The algorithm addresses all four concerns but the method is based on neighborhood model which uses predefined k nearest neighbors as an input parameter. In case data has less neighbors than k, other unrelated neighbors will be forced to be included in the prediction calculation.

In this paper we consider fuzzy clustering method to address all the concerns. First, fuzzy clustering methods generate ratings according to similarity between data and centroids[13-14]. Similarity in fuzzy clustering based CF is computed in term of degree of membership. These centroids and membership degree can be mapped to the set of neighbors and similarity respectively in the neighborhood models. Since centroids represent a set of similar users or items, the membership degrees also explain how similar of data to other members in the cluster. Second, fuzzy clustering based CF methods compute ratings from centroids not directly from data thus counting multiple times does not happen. Third, fuzzy clustering methods have a constraint that sums the membership degree to one, it forces on clustering to compute ratings from all centroids and falls into the same situation as neighborhood models. However, we propose a new fuzzy clustering that improves the quality of rating prediction by allocating data into the relevant clusters. So, only true related centroids will be used to compute the prediction. Fourth, fuzzy clustering methods can adapt other techniques such as normalization[15-16] to adjust the different ratings into the same scale.

In this paper, we focus on developing item-based fuzzy clustering since item-item approaches are more favorable as aforementioned. Basically, clustering-based CF methods calculate degree of membership by pairing profile vectors with the cluster centroids. Most clustering-based CF methods rely on k-means clustering but by doing so they assign data only to one cluster[17-20]. The reliance of CF methods on k-means clustering means they are limited due to their inability to group items into multiple clusters. For example an item such as The Lord of the Rings trilogy can be categorized to both action movies and fantasy movies. It is more reasonable to allow data to belong to multiple clusters by clustering data with uncertainty using fuzzy C-means (FCM)[21]. Nevertheless, this is not enough to make accurate recommendations because irrelevant data could be assigned to the clusters and overwhelm the rating predictions[22].

In general, the ratings should be computed using only ratings from relevant items. To overcome this issue, we propose a new clustering algorithm by reformulating the clustering's objective function with an exponential equation in order to improve the method for membership assignment. The paper is organized as follows. In Section 2, we review the background with some related work and indicate the problems with FCM. In Section 3, we present the new exponential fuzzy clustering. In Section 4, we perform experiments to validate our proposed algorithm by benchmarking against other approaches. In Section 5, we draw a conclusion and make recommendations for the future work.

2 Related Work

In data mining, the objective of clustering techniques is to separate unlabeled data into finite and discrete sets based on similarity or distance functions. Data in the same cluster or set are more similar than data in the other clusters. Clustering has a long history in the research studies such as Wang *et al.*[23] proposed pattern similarity for clustering and it can be applied to the CF domain. Wattanachon *et al.*[24] proposed a hybrid clustering algorithm to handle noisy data in nonlinear data analysis. George *et al.*[17] used co-clustering to handle dynamic real-time CF and cold start problems. Gong *et al.*[20] improved the accuracy of clustering-based CF by joining the result of user clustering and item clustering. Pham *et al.*[19] proposed a clustering model to improve accuracy that operates on social information rather than the user-item ratings matrix. Most of the clustering algorithms used in these studies are based on k-means which may not perform well for datasets with overlapping as aforementioned. A more reasonable approach is to use fuzzy clustering methods that allow data to be included in multiple clusters and is studied in this paper.

2.1 Fuzzy C-Means

FCM partitions dataset X $(x_1, x_2, x_3, \ldots, x_N)$ into k clusters. Partitioning involves uncertainty, i.e., the data (x_i) can be assigned to multiple clusters with different degrees of membership (μ_{ij}). The data is assigned to clusters by comparing its distance or dissimilarity d_{ij}^2 to the cluster centroids (v_j). The distances are usually mathematically computed using Euclidean function. The notations used in this paper are summarized in Table 1.

Table 1. Notations

X	A set of data
x_i	Data in dataset X
N	Total number of data in dataset X
k	Total number of clusters or number of nearest neighbors
d_{ij}^2	Distance from data x_i to centroid v_j
d_{iu}^2	Distance from data x_i to centroid v_u
v_j	Cluster centroid
μ_{ij}	Degree of membership for data x_i and cluster centroid v_j
J	Objective function
m, λ	Fuzzifier parameters
ϵ	Clustering termination coefficient
$P_{u,j}$	Prediction rating of user u on item j

FCM is the most classical method for fuzzy clustering proposed by Dunn[25] and Bezdek[26]. The objective function is formulated by sum of square distance and the degree of membership as (1).

$$J_{FCM} = \sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij}^m d_{ij}^2, \quad m \in (1, \infty), \quad \sum_{j=1}^{k} \mu_{ij} = 1. \tag{1}$$

The fuzzifier parameter (m) is used to control impact of the membership degree in the objective function. By minimizing (1) through the Lagrange multiplier, the solution for the values of membership degree and centroid are shown in (2) and (3) respectively.

$$\mu_{ij} = \frac{1}{\sum_{u=1}^{k} \left(\frac{d_{ij}^2}{d_{iu}^2}\right)^{\frac{1}{m-1}}}, \tag{2}$$

$$v_j = \frac{\sum_{i=1}^{N} \mu_{ij}^m x_i}{\sum_{i=1}^{N} \mu_{ij}^m}. \tag{3}$$

Although FCM assigns data to multiple clusters, allocation is more than enough in most clusters. This difficulty prevents FCM from generating predictions with high accuracy. For example, a dataset is clustered into four clusters (C_1, C_2, C_3, C_4) and the distance from a datum to each cluster's centroid is 15, 20, 30 and 500 respectively. If fuzzifier m is changed over a range of values, the impact of m is visualized in Fig.1.

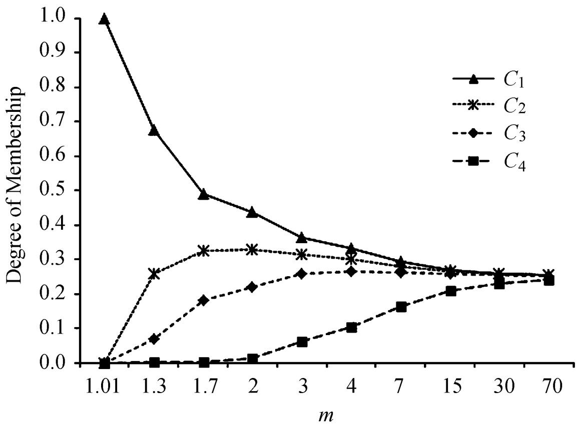

Fig.1. Impact of the value of fuzzifier on the degree of membership when using FCM.

For clustering-based CF, the quality of predictions is dependent on how centroids are influenced. For example the data shown in Fig.1 should not belong to C_4 but FCM begins to assign data to C_4 when m is around 1.7. The larger the value of m, the greater contribution of irrelevant data to the cluster assignment of the data (x_i). The appropriate value for the fuzzifier m that causes minimal distortion of the data should be between [1.01, 1.7] when the data belongs only to C_1, C_2 and C_3 (see Fig.1).

2.2 Fuzzy C-Means with Entropy Regularization

In order to improve fuzzy degree assignment of membership values, fuzzy C-means with entropy regularization (FCME) is proposed. The entropy term from the information theory is applied to the objective function as shown in (4)[27-28].

$$J_{FCME} = \sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij} d_{ij}^2 + \lambda \sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij} \log \mu_{ij}, \quad \lambda > 0, \quad \sum_{j=1}^{k} \mu_{ij} = 1. \tag{4}$$

The Lagrange multiplier is used to find solutions for μ_{ij} and v_j as shown in (5) and (6) respectively.

$$\mu_{ij} = \frac{\exp(-\lambda \times d_{ij}^2)}{\sum_{u=1}^{k}(\exp(-\lambda \times d_{ij}^2))}, \quad (5)$$

$$v_j = \frac{\sum_{i=1}^{N}\mu_{ij}x_i}{\sum_{i=1}^{N}\mu_{ij}}. \quad (6)$$

The algorithm tries to minimize the objective function by minimizing within-cluster dispersion and maximizing the negative entropy. Parameter λ is used to control the effect of the entropy term. From the previous example, when λ is changed, its impact on the membership values μ_{ij} can be visualized in Fig.2.

Fig.2 looks like a mirror image of Fig.1 reflected across the Y-axis of the graph. At the point where the value of λ is very large, i.e., beyond 0.9, only the nearest data will be assigned to the cluster or similar to k-means clustering. At the other end of the graph, FCME initially contributes data to the cluster and stops when λ is around 0.009. Thus, the range of λ should be [0.009, 0.9] for FCME clustering.

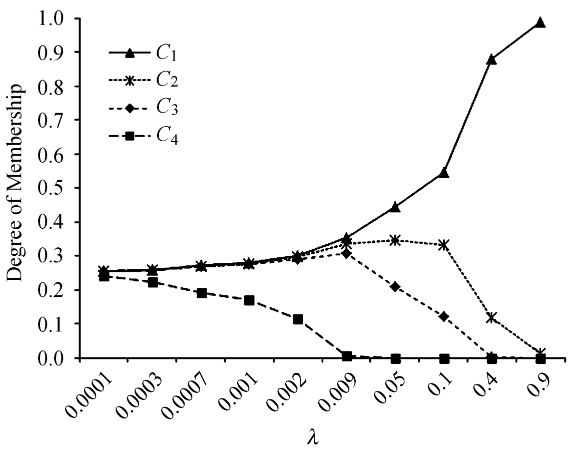

Fig.2. Impact of the value of fuzzifier on the degree of membership when using FCME.

2.3 Membership Distribution

Fuzzy clustering allocates data to all clusters but the degrees of membership that are assigned by each algorithm are different. For clustering-based CF, clusters that include only relevant data are crucial for achieving a high accuracy for rating predictions. In previous examples, FCME has a higher flexibility than FCM with respect to its ability of having its algorithm adjusted since FCME can use a wider range of fuzzifier values. As shown in Figs. 1 and 2, FCME has more options for the allocation of the degrees of membership of data for $C_1 \sim C_3$. Besides, membership degree for C_2 should be closer to C_1 than C_3. For FCM the difference between the membership values of C_1 and C_2 is the smallest when $m = 1.7$ and the difference is 0.2. When λ is around 0.009, the FCME algorithm assigns much closer membership values for C_1 and C_2 than FCM can. This is the limitation of FCM since it produces the degree of membership using a polynomial function whereas FCME uses an exponential function.

In Fig.3, the value of the fuzzifier is fixed, but the distance between the data and the centroid is varied. The algorithm generated degree of membership using FCM from (2) linearly decreases as distance increases. For FCME, the degree of membership slightly decreases during the initial period, i.e., the data gets a very high degree of membership when the distance is changed between 10~160. Thus, most data within this distance range of the cluster are potentially included in the cluster by default. This membership behavior is called *conservative membership distribution*. In our perspective, a cluster should collect only the most relevant data especially since the high prediction accuracy is very important for CF. If data are allocated to unrelated clusters, the ratings are eventually computed by overwhelmed centroids. The membership behavior should follow aggressive membership distribution by rapidly changing even if the data is located near the centroid. To achieve this, the membership function should be represented in a logarithmic function, and the objective function must be developed in an exponential equation in order to get the Logarithmic membership function.

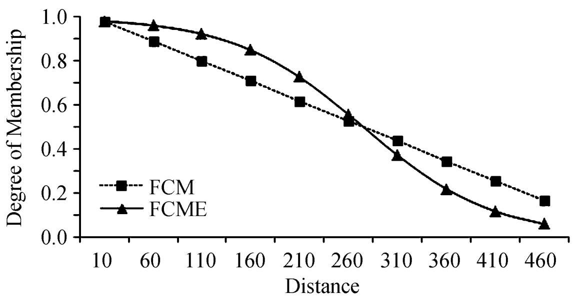

Fig.3. Comparison of the relationship between the distance to the degree of membership for FCM and FCME.

3 Exponential Fuzzy Clustering

The objective function for exponential fuzzy clustering (XFCM)[22] that fulfills the above requirements can be formulated based on two conditions. First, the condition sets $\mu_{ij} = 0$ when data does not belong to any cluster. In this case, the objective function must be 0. Second, the condition sets $\mu_{ij} = 1$ when data belongs to the cluster. The objective function must be equal to the distance function the same as FCM and FCME.

The objective function that meets these conditions is shown in (7).

$$J_{XFCM} = \sum_{j=1}^{k}\sum_{i=1}^{N} \frac{m^{\mu_{ij}}-1}{m-1} d_{ij}^2,$$

$$m \in (1, \infty), \quad \sum_{j=1}^{k} \mu_{ij} = 1. \tag{7}$$

The optimal solution for (7) can be resolved by minimizing the objective function using Lagrange multipliers, which is the same method used for FCM and FCME. We first introduce the Lagrange multiplier λ_i, $i = 1, 2, \ldots, N$ to yield the Lagrange function in (8).

$$L = \sum_{j=1}^{k}\sum_{i=1}^{N} \frac{m^{\mu_{ij}}-1}{m-1} d_{ij}^2 + \sum_{i=1}^{N} \lambda_i \Big(\sum_{j=1}^{k} \mu_{ij} - 1\Big), \tag{8}$$

$$\mu_{ij} = \frac{1}{k}\Big(1 + k\log_m \frac{1}{d_{ij}^2} - \sum_{u=1}^{k} \log_m \frac{1}{d_{iu}^2}\Big)$$

$$= \frac{1}{k}\Big(1 + \log_m \frac{\prod_{u=1}^{k} d_{iu}^2}{(d_{ij}^2)^k}\Big), \tag{9}$$

$$v_j = \frac{\sum_{i=1}^{N}(m^{\mu_{ij}}-1)x_i}{\sum_{i=1}^{N} m^{\mu_{ij}} - 1}. \tag{10}$$

Differentiate (8) by μ_{ij} to get the optimality of the degree of membership in (9). Differentiate (8) again by v_j to get the optimality of the centroid function in (10).

The degree of membership calculated by (9) could go beyond 1 or below 0 to be negative. It happens if and only if condition (11) is true.

$$(d_{ij}^2)^k > \prod_{u=1}^{k} d_{iu}^2. \tag{11}$$

This means that data located very far from the cluster are potentially assigned negative degree of membership. From our perspective, the negative membership degree indicates very low correlation of data x_i and the cluster j. On the other hand, data x_i truly belongs to cluster j if the membership degree goes beyond 1. Thus, we use these properties to filter out irrelevant data when the degree of membership is negative. However, the negative degree of membership is not a fuzzy compliance in which μ_{ij} is in the range of [0, 1].

In order to resolve compliance issues, a new condition is introduced to verify cluster membership. The data will be included in the cluster if and only if the fuzzifier parameter m satisfies the condition (12) otherwise data are not included by assigning 0 to the degree of membership for a particular cluster.

$$m \geqslant \frac{(d_{ij}^2)^k}{\prod_{u=1}^{k} d_{iu}^2}. \tag{12}$$

The same example is reproduced by XFCM as illustrated in Fig.4. Data starts contributing to C_2 and C_3 after the fuzzifier value is over 1 and 2 respectively. It starts contributing to C_4 after the fuzzifier value is over 100. Thus, the usable range is (1, 100], which is larger than the range for FCM and FCME. In addition, the best gap between C_1 and C_2 is at $m = 100$ with a difference of 0.15.

FCME produces a closer gap when $\lambda = 0.009$ but it is not reasonable in practice because the membership values are distributed almost equally in all clusters while distance length to C_3 is the double of the length to C_1. In another example, if the fuzzifier value is fixed and distance changes over the usable range as illustrated in Fig.5 the algorithm produces conservative membership distribution as described by rapidly decreasing from 0.98 to 0.72 in the early stage when the distance changes from 10 to 60.

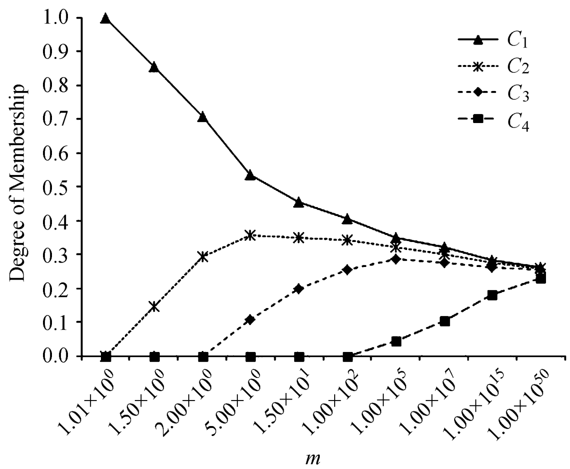

Fig.4. Impact of the value of the fuzzifier on the degree of membership using XFCM.

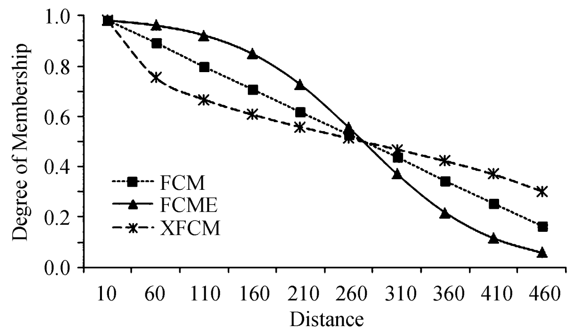

Fig.5. Impact of distance to degree of membership for FCM, FCME and XFCM.

To compare the neighborhood models and clustering-based CF, both methods are similar by trying to find the nearest data. The difference is clustering introduces centroid to represent the set of the nearest data by grouping them together. With this property, it leads clustering method to solve the problem that similarity does not represent similarity among neighbors (the first concern) and accounting the interaction among neighbors (the second concern) by default. For the third concern, it does not well address when using clustering-based CF. There is a condition that sums the membership degree to 1 in the objective functions (1) and (4). This condition forces data to belong to unrelated clusters. For XFCM, it does not allocate data to all clusters because there is a constraint (12) that prevents this situation to happen. Hence, the computed ratings using XFCM are different from other fuzzy clustering whereby overfitting does not occur.

To use XFCM, the procedure is slightly different from other fuzzy clustering since it requires an additional step to recompute membership degree for fuzzy compliance. The procedure is illustrated in Fig.6.

```
Step 1. Predefined Parameters:
- number of cluster (k)
- fuzzifier parameter (m)
- termination coefficient (ε)
- initialize k centroids for each cluster j.
Step 2. Allocation of Data:
For each data
    For each cluster centroid
        Compute d_ij^2
    End For each
End For each
For each data
    For each d_ij^2
        If condition (12) is true then
            Set μ_ij to 0
        Else
            Compute μ_ij according to (9)
        End If
    End For each
End For each
Step 3. Update Centroids:
For each data
    For each cluster
        Update centroid according to (10)
    End For each
End For each
Step 4. Validate Stop Condition:
Calculate objective function according to (7)
If termination condition is met then
    Stop clustering process
Else
    Repeat Step 2
End If
```

Fig.6. Procedure of XFCM.

4 Experiments

Two real datasets were used in experiments to validate the performance of XFCM: the 100 K MovieLens dataset and 1 M MovieLens dataset. We performed an item-item fuzzy clustering method by replacing the distance function in clustering with the inverse of adjusted cosine similarity[2]. We evaluated XFCM's performance by measuring the prediction error using mean absolute error (MAE) as in (13) for each pair of predicted and actual ratings.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n}, \tag{13}$$

where p_i represents the prediction values, q_i represents the actual ratings made by users, n is total number of prediction values.

The predicted ratings in (14) were calculated by withdrawing the rating from the centroid based on the membership function. We used MAE as a cluster validation to validate clustering parameters since it was used for evaluation measurement[13-14,22].

$$P_{u,j} = \sum_{j=1}^{k} \sum_{i=1}^{N} \mu_{ij} v_j, \tag{14}$$

where $P_{u,j}$ is a prediction value for a pair of user u and the movie j.

All algorithms had the same initial seeds. Miss ratings in initial centroids were filled with average item ratings. All algorithms stopped when the objective function changed less than 1 ($\epsilon = 1$). From our test, the optimum results did not improve after 10 interations so the algorithms stopped when the iteration reached the 10th step. At the end of each iteration, MAE was computed and the best MAE was selected from running by loop to compare among testing algorithms.

XFCM clustering-based CF was benchmarked with item-based CF[2,6] algorithms, SVD[6] for 100 K MovieLens dataset and 1 M MovieLens dataset. Implementation of clustering-based CF methods was developed using C# with SQL Server 2005 for back-end database on a Core-i5 computer with ram 4 GB①.

4.1 100 K MovieLens Dataset

100 K MovieLens dataset was collected user rating during the 7-month period from September 19th, 1997 through April 22nd, 1998 for CF research by the GroupLens Research Project at the University of Minnesota. The dataset consists of 100 000 ratings made by 943 users on 1 682 movies. The data is very sparse with a 0.939 6 sparsity level. The sparsity is calculated by

①Implementation of FCM, FCME and XFCM are available for download at: http://iahcitaik.blogspot.com.

1−(nonzero entries/total entries). We separated 80% of dataset for training and 20% for the prediction. An appropriate fuzzifier parameter for each clustering algorithm was selected from experiments as illustrated in Figs. 7∼9. The relationship between fuzzifier m and MAE is represented by U-shaped curves with the minimum MAE at fuzzifier values of 2, 20 and 10 for FCM, FCME and XFCM respectively. These parameters were used going forward in our experiments.

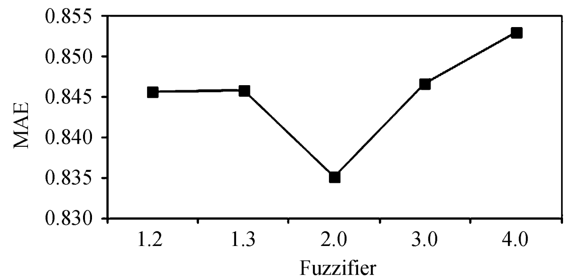

Fig.7. FCM fuzzifier experiment on 100 K MovieLens dataset.

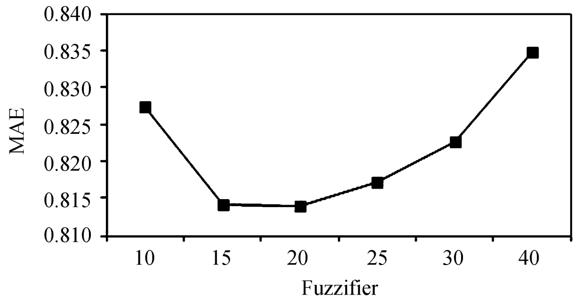

Fig.8. FCME fuzzifier experiment on 100 K MovieLens dataset.

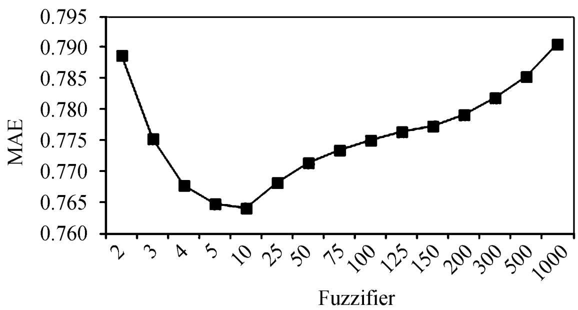

Fig.9. XFCM fuzzifier experiment on 100 K MovieLens dataset.

4.2 1 M MovieLens Dataset

1 M MovieLens dataset contains 1 000 209 anonymous ratings of approximately 3 900 movies made by 6 040 MovieLens users who joined MovieLens in the year 2000. The dataset has a 0.964 0 sparsity. This is calculated using the same method as 100 K MovieLens dataset. We performed experiments using the same strategy as used for 100 K MovieLens dataset. The dataset was separated 80% for training and 20% for the prediction. Accuracy was evaluated by MAE. In this dataset, we implemented a compression technique[29] in order to speed up the similarity calculation. Fuzzifier parameters for each clustering algorithm were selected from experiments as illustrated in Figs. 10∼12. Optimum fuzzifier values are 2, 0.9 and 2 for FCM, FCME and XFCM respectively.

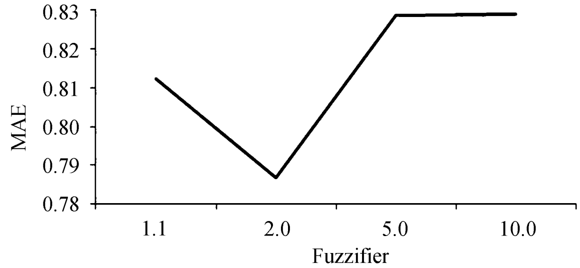

Fig.10. FCM fuzzifier experiment on 1 M MovieLens dataset.

In Figs. 7∼12, the curves are U-shaped. This behavior could be explained by the different levels of fuzziness in correspondence to fuzzifier parameter. First, when the fuzzifier is nearly maximized, FCM and XFCM produce membership for the data by $1/k$ while FCME produces membership to $1/k$ when the fuzzifier is minimized. Second, FCM and XFCM assign the data with the shortest distance to the cluster while FCME behaves the same when the fuzzifier is maximized as displayed in Figs. 1, 2 and 4.

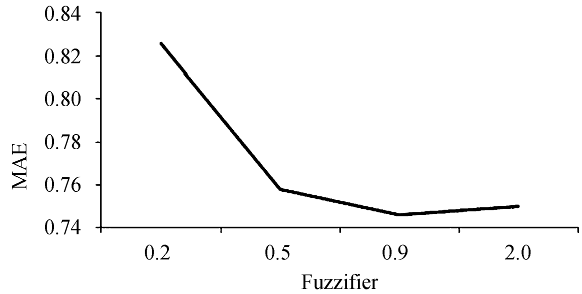

Fig.11. FCME fuzzifier experiment on 1 M MovieLens dataset.

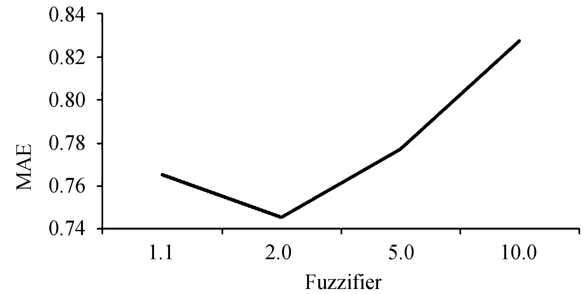

Fig.12. XFCM fuzzifier experiment on 1 M MovieLens dataset.

The U-shaped curves are the result of the trade-off between these two behaviors thus the saddle point is in the middle of each curve. As mentioned earlier, the generated ratings depend on centroids, and the process to update the centroids relies on the quality of the members in the cluster. Hence, the ratings are overwhelmed by irrelevant data if the fuzzifier is maximized and vice versa if the fuzzifier is minimized. FCME behaves the opposite way since the ratings are overwhelmed by irrelevant data when the fuzzifier is minimized and vice versa if the fuzzifier is maximized. Given these circumstances, there is an incentive to optimize the fuzzifier by narrowing the scope to the saddle point of the curve.

4.3 Benchmarking Results

Optimum fuzzifier values from previous experiments will be used to perform clustering-based CF at a variety of clusters.

From Fig.13 and Fig.14, XFCM outperforms FCM by 6.4~9.1% and FCME by 4.6~6.1% for 100 K MovieLens dataset, and outperforms FCM by 5.2~9.8% and FCME by 1.0~2.5% for 1 M MovieLens dataset. For benchmarking against other algorithms, XFCM performs the best for both datasets by outperforming item-based CF by 6.9%, SVD by 3% for 100 K MovieLens dataset as illustrated in Fig.15, and outperforms item-based CF by 2.7%, SVD by 1%, for 1 M MovieLens dataset as illustrated in Fig.16.

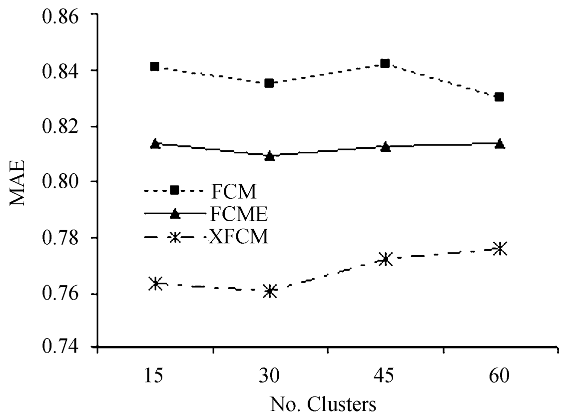

Fig.13. Benchmarking result of each fuzzy clustering by different clusters for 100 K MovieLens dataset.

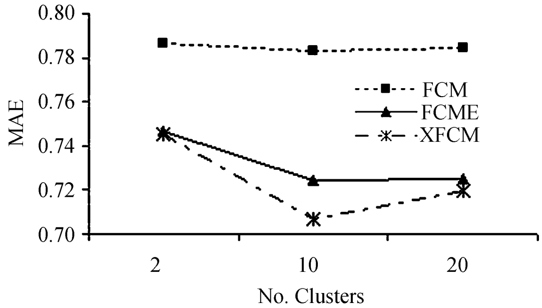

Fig.14. Benchmarking result of each fuzzy clustering by different clusters for 1 M MovieLens dataset.

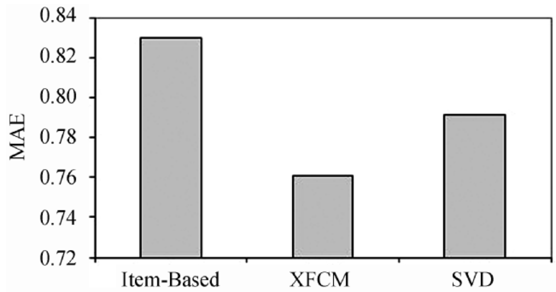

Fig.15. Benchmarking result between XFCM-based CF with item-based CF[2] and SVD[6] for 100 K MovieLens dataset.

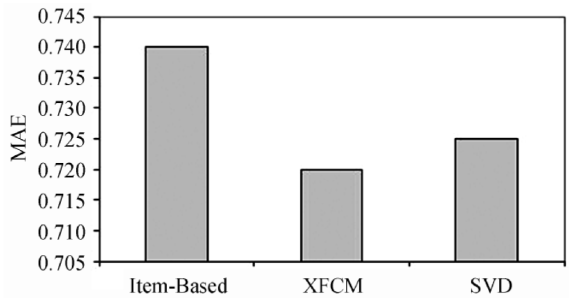

Fig.16. Benchmarking result between XFCM-based CF with item-based CF[2] and SVD[6] for 1 M MovieLens dataset.

We further performed statistics analysis on the results of each clustering algorithm as in Table 2.

Table 2. Average Standard Deviation (S.D.) of Membership Degree

	FCM	FCME	XFCM
100 K MovieLens Dataset			
Total Number	24 750	24 750	24 750
S.D.	0.102 9	0.115 0	0.091 1
$\mu_{ij}=0$	0.02%	1.50%	50.98%
1 M MovieLens Dataset			
Total Number	36 820	36 820	36 820
S.D.	0.049 3	0.130 6	0.214 3
$\mu_{ij}=0$	0.00%	5.89%	71.95%

The standard deviation (S.D.) of each algorithm is not very different from each other while a large number of $\mu_{ij}=0$ are generated by XFCM as a result of aggressive membership distribution. Both FCME and FCM generate ratings from influenced centroids. Thus, it is difficult to achieve high accuracy using these methods. Moreover, FCM spreads the membership of all data to every cluster as illustrated in Table 3 and Table 4. The ratings calculated from (14) are eventually overwhelmed by every rating in the dataset as mentioned earlier. For FCME, the membership distribution looks similar to k-means clustering since only one data is highly correlated to cluster 7 for 100 K MovieLens dataset. For 1 M MovieLens dataset, the data strongly correlates to clusters 1, 3, 7 and 8. Although the contribution to irrelevant clusters of this data is low, the prediction could be deviated by other data in the dataset. For XFCM, only five and four relevant clusters are used to compute ratings for movie ID #1000 in 100 K MovieLens dataset and 1 M MovieLens dataset respectively. Thus, the centroids are computed only using relevant data. In CF perspective, not all centroids are used to predict ratings of a user for movie ID #1000 as indicated in Table 3 and Table 4. Thus, the ratings computed using XFCM-based CF do not overfit.

Table 3. Degree of Membership of Movie ID #1000 from 100 K MovieLens Dataset

Cluster No.	FCM	FCME	XFCM
1	0.1236	7.2E−07	0.0000
2	0.0712	0.0002	0.1872
3	0.2538	0.0001	0.0543
4	0.1097	8.16E−06	0.0000
5	0.0145	5.22E−05	0.2301
6	0.0050	2.26E−05	0.0000
7	0.1063	0.9316	0.1529
8	0.0068	1.46E−05	0.0000
9	0.0500	0.0543	0.0000
10	0.0272	3.4E−06	0.3754
11	0.0924	0.0028	0.0000
12	0.0015	2E−08	0.0000
13	0.0206	0.0097	0.0000
14	0.0116	0.0004	0.0000
15	0.1050	0.0006	0.0000

Table 4. Degree of Membership of Movie ID #1000 from 1 M MovieLens Dataset

Cluster No.	FCM	FCME	XFCM
1	0.1167	0.1557	0.3779
2	0.0961	0.0933	0.0000
3	0.1222	0.2202	0.4804
4	0.0580	0.0091	0.0000
5	0.1014	0.0195	0.0000
6	0.0922	0.0619	0.0000
7	0.1170	0.1712	0.1378
8	0.1039	0.1395	0.0039
9	0.0797	0.0443	0.0000
10	0.1128	0.0852	0.0000

By using clustering-based CF especially with XFCM, all four concerns are addressed as mentioned earlier. First, fuzzy clustering produces similarity in term of membership degree which explains the relation between data and their neighbors in the clusters. Second, fuzzy clustering accounts interaction among neighbors by grouping them into the cluster regarding to degree of membership. Third, XFCM provides an incentive condition to prevent overfitting through the selection of members into the clusters. Fourth, preprocessing data such as normalization can be adapted to clustering in order to adjust the ratings into the same scale. These reasons lead XFCM to improve prediction accuracy over other methods.

5 Conclusions

Although fuzzy clustering has many success stories for many applications, FCM produces too much fuzziness for the assignment of degree of membership values when it is used for CF. FCME is an improved membership assignment method, but it is conservative membership distribution and the inclusion of irrelevant data can happen. We propose a new clustering method formulated from the Exponential function, which leads to aggressive membership distribution. Various experiments were performed on real life datasets to validate XFCM. The MAE results show that XFCM outperforms FCM by 5.2∼9.8%, FCME by 1.0∼6.1%, the item-based method by 2.7∼6.9% and SVD by 1.0∼3.0% for both MovieLens datasets. Further experiments indicate that XFCM works very well by discarding irrelevant data when updating its centroid and eventually predictions are not overwhelmed by irrelevant data.

In summary, clustering-based CF has the potential to produce inaccurate results as a result of irrelevant data that are included in the clusters. Although some fuzzy clustering algorithms produce good membership assignment to prevent the influence from irrelevant data, they are eventually included in the cluster. XFCM proposes to change membership distribution to be aggressive. This approach is a promising algorithm for the allocation of data to the clusters in clustering-based CF method. In the future, we plan to integrate the possibilistic approach in order to create a more concrete and robust algorithm.

Acknowledgements Special thanks to Tirapon Premchitt for his proofreading.

References

[1] Herlocker J L, Konstan J A, Borchers A, Riedl J. An algorithmic framework for performing collaborative filtering. In *Proc. the 22nd ACM SIGIR Conf. Research and Development in Information Retrieval*, Aug. 1999, pp.230-237.

[2] Sarwar B, Karypis G, Konstan J, Riedl, J. Item-based collaborative filtering recommendation algorithms. In *Proc. the 10th Int. Conf. World Wide Web*, May 2001, pp.285-295.

[3] Karypis G. Evaluation of item-based top-N recommendation algorithms. In *Proc. the 10th Conf. Information and Knowledge Management*, Nov. 2001, pp.247-254.

[4] Linden G, Smith B, York J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 2003, 7(1): 76-80.

[5] Bell R, Koren Y. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proc. the 7th Int. Conf. Data Mining*, Oct. 2007, pp.43-52.

[6] Koren Y, Bell R. Advanced in collaborative filtering. In *Recommender Systems Handbook* (1st edition), Springer, 2011,

pp.145-186.

[7] Sarwar B M, Karypis G, Konstan J A, Riedl J T. Application of dimensionality reduction in recommender system — A case study. In *ACM WebKDD Web Mining for ECommence Workshop*, Aug. 2000.

[8] Vozalis M, Markos A, Margaritis K G. Evaluation of standard SVD-based techniques for collaborative filtering. In *Proc. the 9th Hellenic European Research on Computer Mathematics and its Applications*, Sept. 2009.

[9] Rendle S. Factorization machines. In *Proc. the 10th Int. Conf. Data Mining*, Dec. 2010, pp.995-1000.

[10] Ali K, van Stam W. TiVo: Making show recommendations using a distributed collaborative filtering architecture. In *Proc. the 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Aug. 2004, pp.394-401.

[11] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proc. the 14th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Aug. 2008, pp.426-434.

[12] Liu N N, Yang Q. EigenRank: A ranking-oriented approach to collaborative filtering. In *Proc. the 31st Conf. ACM SIGIR on Information Retrieval*, Jul. 2008, pp.83-90.

[13] Treerattnapitak K, Jaruskulchai C. Entropy based fuzzy *C*-mean for item-based collaborative filtering. In *Proc. the 9th Int. Symposium on Communication and Information Technology*, Sept. 2009, pp.881-886.

[14] Treerattnapitak K, Jaruskulchai C. Items based fuzzy C-mean clustering for collaborative filtering. *Information Technology Journal*, 2009, 5(10): 30-34.

[15] Jin R, Si L. A study of methods for normalizing user ratings in collaborative filtering. In *Proc. the 27th Conf. ACM SIGIR on Research and Development in Information Retrieval*, Jul. 2004, pp.568-569.

[16] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. the 14th Conf. Uncertainty in Artificial Intelligence*, Jul. 1998, pp.43-52.

[17] George T, Merugu S. A scalable collaborative filtering framework based on co-clustering. In *Proc. the 5th IEEE Int. Conf. Data Mining*, Nov. 2005, pp.625-628.

[18] Ungar L H, Foster D P. Clustering methods for collaborative filtering. In *Proc. AAAI Workshop on Recommendation System*, Jul. 1998.

[19] Pham M C, Cao Y, Klamma R, Jarke M. A clustering approach for collaborative filtering recommendation using social network analysis. *Journal of Universal Computer Science*, 2011, 17(4): 583-604.

[20] Gong S. A collaborative filtering recommendation algorithm based on user clustering and item clustering. *Journal of Software*, 2010, 5(7): 745-752.

[21] Wu J, Li T. A modified fuzzy C-means algorithm for collaborative filtering. In *Proc. the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, Aug. 2008, Article No. 2.

[22] Treerattnapitak K, Jaruskulchai C. Membership enhancement with exponential fuzzy clustering for collaborative filtering. In *Proc. the 17th Int. Conf. Neural Information Processing*, Nov. 2010, pp.559-566.

[23] Wang H, Pei J. Clustering by pattern similarity. *Journal of Computer Science and Technology*, 2008, 23(4): 481-496.

[24] Wattanachon U, Suksawatchon J, Lursinsap C. Nonlinear data analysis using a new hybrid data clustering algorithm. In *Lecture Notes in Computer Science* 5476, Theeramunkong T *et al.* (eds.), Springer-Verlag, 2009, pp.160-171.

[25] Dunn J C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics.*, 1973, 3(3): 32-57.

[26] Bezdek J C. Pattern Recognition with Fuzzy Objective Function Algoritms. New York: Plenum Press, 1981.

[27] Miyamoto S, Mukaidono M. Fuzzy C-means as a regularization and maximum entropy approach. In *Proc. the 7th Int. Fuzzy System Association World Congress* (*IFSA 1997*), Jun. 1997, 2: 86-92.

[28] Miyamoto S, Ichihashi H, Katsuhiro H. Algorithms for Fuzzy Clustering: Methods in *c*-Means Clustering with Applications. Springer-Verlag Berlin Heidelberg, 2008.

[29] Goharian N, El-Ghazawi T A, Grossman D A, Chowdhury A. On the enhancements of a sparse matrix information retrieval approach. In *Proc. the Int. Conf. Parallel and Distributed Processing Technology and Application*, Jun. 2000.

Kiatichai Treerattanapitak is currently a Ph.D. candidate in Kasetsart University, Thailand. He is also a senior consultant of environmental resources management in Thailand. His research interests include artificial inteligence, data mining, and clustering.

Chuleerat Jaruskulchai received her D.Sc. degree in computer science from School of Engineering and Applied Science, George Washington University, USA, in 1998. She is currently an associate professor in the Department of Computer Science, Kasetsart University, Thailand. Her interest and research areas include information retrieval, clustering, text classification, and statistic modeling.