Dynamic Fuzzy Cluster based Collaborative Filtering

Sung-Hwan Min* and Ingoo Han

Graduate School of Management, Korea Advanced Institute of Science and Technology, 207-43 Cheongrangri-dong, Dongdaemun-gu, Seoul 130-722, Korea

Email: shmin@kgsm.kaist.ac.kr

Abstract. Due to the explosion of e-commerce, recommender systems are rapidly becoming a core tool to accelerate cross-selling and strengthen customer loyalty. There are two prevalent approaches for building recommender systems - content-based recommending and collaborative filtering. Collaborative filtering recommender systems have been very successful in both information filtering domains and e-commerce domains, and many researchers have presented variations of collaborative filtering to increase its performance. However, the current research on recommendation has paid little attention to the use of time related data in the recommendation process. Up to now there has not been any study on collaborative filtering to reflect changes in user interest. This paper proposes dynamic fuzzy clustering algorithm and apply it to collaborative filtering algorithm for dynamic recommendations. The proposed methodology detects changes in customer behavior using the customer data at different periods of time and improves the performance of recommendations using information on changes. The results of the evaluation experiment show the proposed model's improvement in making recommendations.

1 Introduction

Recommender systems are the information filtering process to supply personalized information by predicting user's preferences to specific items. In a world where the number of choices can be overwhelming, recommender systems help users find and evaluate items of interest [5]. To date, a variety of techniques for building recommender systems have been developed. These techniques can be classified into two main categories: content-based filtering and collaborative filtering (CF). CF is the most successful recommendation technique, which has been used in a number of different applications such as recommending movies, articles, products, Web pages CF is built on the assumption that a good way to predict the preference of the active consumer for a target product is to find other consumers who have similar preferences, and then use those similar consumer's preferences for that product to make a prediction [3].

Currently, CF algorithms can be classified into memory-based and model-based algorithms [2]. Memory-based algorithms repeatedly scan the preference (or people) database to locate the peer groups for the active users. A prediction is then computed by weighting the votes of users in the peer groups. The people in the peer groups are identified based on their similarity or nearness in tastes to the active user. Model-based algorithms infer a user model from the database of rating histories. The user model is then consulted for predictions. Model-based algorithms require more time to train but can provide predictions in a shorter time in comparison to nearest-neighbor algorithms [5].

Finding neighbors in memory-based CF is crucial for accurate recommendations because recommendations are based on the ratings of an active user's neighbors. For example, if an active user has preference for a small-sized car, his or her neighbors should be selected among the people who also have preference for a small-sized car. But, if his or her preference has moved from a small-sized car to a large-sized car, then his or her neighbors should be changed immediately. But the current CF algorithms are not adaptive to these situations dynamically, which results in the false recommendations. Our research focuses on these situations.

The purpose of this research is to develop a model which is adaptive to users' changing patterns in order to improve recommendations. In this paper we present a new approach to collaborative filtering based on dynamic fuzzy cluster. The proposed model is expected to find the active user's neighbors dynamically according to his or her changing pattern by using dynamic fuzzy cluster and improve recommendations.

The rest of this paper is organized as follows: The next section describes fuzzy clustering. Section 3 presents the proposed model. Section 4 describes change measure. Section 5 explains the results of the evaluation experiment. The final section presents the summary and future research issue.

2 Fuzzy Clustering

The most widely used fuzzy clustering algorithm is the fuzzy c-means(FCM) algorithm proposed by Bezdek [1,6].FCM is a clustering method in which an object can be a member of different classes at the same time. This method is an unsupervised clustering algorithm that has been applied successfully to a number of problems involving feature analysis, clustering and classifier design. FCM aims to determine cluster centers v_i (i=1, 2,...,c) and the fuzzy partition matrix U by minimizing the objective function J defined as follows:

$$J_m(U,V;X)=\sum_{j=1}^{n}\sum_{i=1}^{c}u_{ij}{}^{m}\left\|x_j-v_i\right\|^2 \qquad (1)$$

where n is the number of individuals to be clustered, c is the number of clusters and u_{ij} is degree of membership of individual j in cluster i. The exponent m is used to control the fuzziness of membership of each datum. $\left\|x_j-v_i\right\|$ is the Euclidean norm between x_j and v_i. The FCM algorithm is as follows:

Step 1. Initialize u_{ij} by generating random numbers in the interval [0, 1] such that

$$\sum_{i=1}^{c}u_{ij}=1\ \forall j=1,\ldots,\mathrm{n} \qquad (2)$$

Step 2. Compute the fuzzy cluster centroid v_i for I = 1, 2, ..., c according to the following Eq. (3)

$$v_i=\frac{\sum_{j=1}^{n}(u_{ij})^m x_j}{\sum_{j=1}^{n}(u_{ij})^m} \qquad (3)$$

Step 3. Update the degree of membership u_{ij} using

$$u_{ij}=\frac{1}{\sum_{k=1}^{c}(\frac{d_{ij}}{d_{kj}})^{2/m-1}}=\frac{1}{\sum_{k=1}^{c}(\frac{\left\|x_j-v_i\right\|}{\left\|x_j-v_k\right\|})^{2/m-1}} \qquad (4)$$

Step 4. If the improvement in $J_m(U,V;X)$ is less than the given threshold ε, then stop. Otherwise go to step 2.

In this paper, the FCM algorithm is used in order to cluster users. We modify the FCM algorithm and apply it to CF for dynamic recommendations.

3 Dynamic Fuzzy Cluster based CF

Most recommender systems do little to detect or predict the changes in preference, although the customer interest does vary from time to time in the world. Especially for an internet-based company, it is of crucial importance knowing what is changing and how it has been changed because it allows the management to provide the right products and services to suit the changing market needs.

This paper suggests a dynamic fuzzy cluster based collaborative filtering which is adaptive to users' changing patterns in order to improve recommendations. The procedure for the proposed model is as follows:

Step 1. Data Preparation: Add a time dimension to the original input data and reduce item dimension by using hierarchy information.
Step 2. User Clustering: Apply the FCM algorithm to produce p partitions of users. This is used as a cluster base for finding time-variant fuzzy cluster.
Step 3. Dynamic fuzzy cluster: Find fuzzy cluster at different timeframes for a given active user and compute the dynamic degree of membership.

Step 4. Neighbor Selection: Determine the neighborhood for a given user based on the dynamic fuzzy cluster.
Step 5. Recommendation: Predict the active users' rating unanswered based on neighborhood ratings.

In the following section, we will explain each step in detail.

3.1 Data Preparation

Input data of a CF problem is usually a user-to-rating matrix in Table 1. In order to detect the dynamic cluster change for an active user, we need to add a time dimension to the original input data.

Table 1. Original CF input data (User-to-Item Rating Matrix)

	Item1	Item2	Item3	Item4	...	Item n
User1		2				
User2	3		1		...	
...						1
Active User	1	5	2	5		3
User n						

Table 2 shows item ratings for an active user at different timeframes. As shown in Table 2, each row is too sparse. To solve this problem, input data reduction (item dimension reduction) methods are needed. In this paper, we use a hierarchy of items whose leaf nodes represent items and non-leaf nodes represent a higher-level category to reduce the dimension of input data space.

Table 2. Time-to-Item Rating Matrix of an active user

Timeframe	Item1	Item2	Item3	Item4	...	Item n
T1	1					
T2		5	2			
T3				5		3
...					...	
Tn						

The rating for a category is defined as the average ratings for the items in that category as follows [7].

$$cr_{a,k} = \sum_{i \in category\ k} \frac{1}{RN_k} r_{a,i} \quad (5)$$

In above equation, $CR_{a,k}$ is the derived rating of category k of user A, RN_k is the number of rated items that belong to that category, and $r_{a,i}$ is the rating of item i of user A. These derived ratings of non-leaf level nodes are incorporated in computing the fuzzy cluster at different timeframes. Table 3 shows an example of category ratings. In section 4, we describe the experiments in which we use genre data as category information.

Table 3. Time-to-Category Rating Matrix

Timeframe	Category 1						Category n	
	Item 1	Item 4	Item 7				Item 2	Item i
T1	1		3					
	2							
T2			1				5	
	1							
T3		5						1
	5						1	
...								

3.2 User Clustering

Customers with similar interests are clustered by the FCM algorithms and this output is used as a base for detecting dynamic cluster change. In the FCM process, category ratings, calculated using item hierarchy information, are used as input data in order to extend the FCM into the dynamic FCM. Fuzzy cluster, defined as the degree of

membership u_{ij}, is computed in this step. Crisp cluster of a given user is also determined. Crisp cluster of a given user j (CC_j) is defined as the cluster with the largest degree of membership for a given user as follows.

$$CC_a = k, \text{ if } u_{ka} = \max_i \{u_{ia}\} \text{ for user a} \quad (6)$$

Information on the crisp cluster of all users is used in neighbor selection step.

3.3 Dynamic Fuzzy Cluster

We propose the dynamic fuzzy cluster which is used to detect the time-variant cluster of an active user and to find an active user's neighbors dynamically. Input data shown in Table 3 is used to find the fuzzy cluster at different timeframes for a given active user. One user may belong to the same fuzzy cluster at different time-frames, but another user may belong to the different fuzzy cluster at different timeframes as shown in Table 4.

Table 4. Dynamic Fuzzy Cluster

Cluster / Timeframe	C1	C2	C3	C4	C5	C6	C7	C8
T1 (t=1)		0.1	0.7		0.2			
T2 (t=2)		0.2	0.6		0.2			
T3 (t=3)		0.1			0.1	0.7		
T4 (t=4)					0.1	0.8	0.1	
su_{ij} (w(t)=1+t)	0	0.8	1.9	0	1.3	5.3	0.4	0
du_{ij}	0	0.08	0.19	0	0.13	0.54	0.04	0

Different clusters at different timeframes means that the user may have a time-variant pattern. Sum of degree of membership (su_{ij}) is computed as follows.

$$su_{ij} = \sum_{t=1}^{T} w(t) \bullet u(t)_{ij} \quad (7)$$

where $u(t)_{ij}$ is the degree of membership of individual j in cluster i at timeframe t and w(t) is the weighting function which is used to weight $u(t)_{ij}$ differently according to timeframe. Dynamic degree of membership of individual j in cluster i (du_{ij}) is defined as follows.

$$du_{ij} = \frac{su_{ij}}{\sum_{i=1}^{c} su_{ij}} \quad (8)$$

3.4 Neighbor Selection

We select neighbors among users in the cluster i where the dynamic degree of membership of active user (du_{ia}) is larger than zero. Number of neighbors selected in each cluster is proportional to du_{ij}.

The procedure for selecting nearest neighbors is as follows.

Step 1. Sort du_{ij} in ascending order and initialize $n_i = 0$ for i, ..., c (c=number of cluster)

Step 2. If $du_{ka} = \max_i \{du_{ia}\}$ for i where $n_i = 0$, $n_k = \text{int}(du_{ka} \cdot N)$

Step 3. If $N \geq \sum_{i \in D} n_k$, update n_k ($n_k = n_k - (\sum_{i \in D} n_i - N)$) and stop.

Otherwise go to step 2.

Step 4. Select the nearest n_i users in cluster i as neighbors. (for i = 1, ..., c)

In the above explanation, N is the total number of neighbors and n_k is the number of neighbors selected in cluster k. D is the cluster set which consists of clusters that n_i is larger than zero. Int(m) means the smallest integer value more than m.

3.5 Recommendation

Once the neighborhood is selected, traditional collaborative filtering algorithm is used to generate recommendation from that. After $S_{a,v}$ is obtained, the predicted numerical rating $r_{a,x}$ of the active user for a target item x is calculated as follows.

$$P_{a,x} = \bar{r}_a + \frac{\sum_{v \in Raters} (r_{v,x} - \bar{r}_v) \cdot S_{a,v}}{\sum_{v \in Raters} |S_{a,v}|} \qquad (9)$$

where raters are the selected neighbors who rate the target item, $\bar{r}_a$ is the average rating of user A, and $S_{a,v}$ it the similarity weight between the active user and neighbor v.

4 Change Measure

In this paper, we propose using Auto-Similarity measure to detect a change by the active user. Auto-Similarity means the similarity weight between an active user's ratings at different timeframes and it is also calculated by Eq. (10) while similarity in the traditional CF approach means the similarity weight between an active user and neighbors

This measure is used for detecting the degree and the characteristics of changes.

Auto-Similarity $AS(a)_{t1,t2}$ is defined as the similarity between the active user's category ratings at timeframe T1 and the same active user's category ratings at timeframe T2.

$$AS(a)_{T1,T2} = \frac{\sum_k (cr(T1)_{a,k} - \overline{r(t1)_a}) \cdot (cr(T2)_{a,k} - \overline{r(T2)_a})}{\sqrt{\sum ((cr(T1)_{a,k} - \overline{r(T1)_a})^2} \cdot \sqrt{\sum ((cr(T2)_{a,k} - \overline{r(T2)_u})^2}} \qquad (10)$$

where k is the index of each category that user A has rated at both timeframe T1 and timeframe T2, $cr(T1)_{a,k}$ is user A's category rating for category k at timeframe T1, $cr(T2)_{a,k}$ is the same user A's rating for category k at timeframe T2, $\overline{r(T1)_a}$ denotes the average category rating of active user A at timeframe T1 and $\overline{r(T2)_a}$ is the average of the same user's category ratings at timeframe T2.

$AS(a)_{T1,T2}$ is 1 if the active user has exactly the same preference for the category rating at timeframe T1 and timeframe T2, and is -1 if he or she has a completely different preference between T1 and T2. When $AS(a)_{T1,T2}$ is -1, we can conclude that the active user's interest changed after timeframe T1. If there is no correlation between the preferences of the category rating at timeframe T1 and the category rating at timeframe T2 of the active user, $AS(a)_{T1,T2} = 0$. If user A tends to have a similar category rating for each timeframe, $AS(a)_{T1,T2}$ becomes a positive number. The absolute value of $AS(a)_{T1,T2}$ indicates how much user A at timeframe T1 tends to agree with the same user at timeframe T2 on the category rating that he or she rated at each timeframe. If they tend to have opposite ratings for each category, $AS(a)_{T1,T2}$ becomes a negative number. The absolute value of $AS(a)_{T1,T2}$ indicates how much user A at timeframe T1 tends to

disagree with the same user at timeframe T2 on the category rating that he or her rated at each timeframe. $AS(a)_{T1,T2}$ can be between -1 and 1. If $AS(a)_{t1,t2}$ is less than some negative value, we can conclude that the active user's behavior changed after timeframe T1.

5 Experimental Evaluation

5.1 Data Set

For experiments we used the EachMovie database, provided by Comaq Systems Research Center [8]. The data-set contains explicit rating data provided by each user for various movies. The EachMovie dataset has rating information on 1,628 movies by 72,916 users during an 18 month period from 1996. Users rated various numbers of movies by the number of stars 0 to 5. In these experiments, we normalized the rating values to values between 0 and 1. Other than the rating information, the database contains the movie title, genre, each user's age, and gender.

We assumed that the items are classified into a multi-level (hierarchical) category, and we used the category information to compute the similarity between an active user's ratings at different timeframes. Even when an active user does not have rating information on the same item at different timeframes, the same user's cluster change can be computed if some of the rated items belong to the same category in higher level. In this experiment we used genre data as category information.

5.2 Evaluation Metrics

We used MAE as our choice of evaluation metric to report prediction experiments because it is commonly used and easy to interpret. MAE is a measure of the deviation of recommendations from their true user-specified values. For each ratings-prediction pair $<p_i,q_i>$ this metric treats the absolute error between them i.e., $|p_i-q_i|$ equally [4]. The MAE is computed as follows.

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{11}$$

where N is the number of ratings in the test data set.

5.3 Experimental Method

First we selected 1200 users with more than 100 rated items. We divided the data set into a training set and a test portion. Before starting full experimental evaluation of different algorithms we determined the sensitivity of different parameters. We fixed the optimum values of these parameters from the sensitivity plots and used them for the rest of the experiments.

To compare the performance of the proposed dynamic fuzzy cluster based CF (DFCF) algorithm we used the traditional CF (TCF) algorithm as the benchmark model. The traditional CF recommendation employs the Pearson nearest neighbor algorithm. We also experimented using both fuzzy CF (FCF) algorithm and crisp cluster based CF (CCF) algorithm. In CCF algorithm, crisp cluster is determined by using Eq.(6) and the degree of membership in the crisp cluster is defined as 1 while the degree of membership in other clusters is zero.

5.4 Experimental Results

Table 5 presents the performance of the competing models according to the metric of MAE of recommendation. It can be observed that the proposed dynamic fuzzy cluster based CF algorithm outperforms the traditional CF algorithm. When the number of cluster is 7, the performance of the proposed model is best. Table 6 present the results of FCM when the number of clusters is 7. Prediction quality of the FCF is worse than TCF but the difference is small while prediction quality of CCF is worst. It can also be observed from the table that as the number of clusters increase the quality tends to be inferior in case of CCF. In addition, a set of pairwise t-tests in Table 7 indicates that the differences were statistically significant. DFCF reflects better user preference than other

models at the 5% significance level. These results show that the proposed DFCF algorithm is more accurate than the traditional CF algorithm.

To experimentally compare the quality of DFCF with TCF according to the change threshold value, we selectively varied the value of threshold to be used for auto-similarity computation from -.01 to -1 in increments of -0.1. A threshold value of -0.1 means that we only considered -0.1 as the best threshold values for detecting change. If the auto-similarity value of the active user is smaller than -0.1, we conclude that the active user's pattern changed.

Fig. 1 shows the plots at different change threshold(TS) values. TCF-All means recommendations of all users' ratings in the test data set by TCF while TFCF-All describes recommendations of all users' ratings in the test data set by the proposed TFCF. TCF-Changed User means recommendations of ratings of users who are detected as changed users, in the test data set by TCF while DFCF-All describes recommendations of changed users' ratings in the test data set by the proposed DFCF.

As we can see from the results, MAE values of TCF-All and DFCF-All have no relation to the change threshold value and MAE values of both TCF-Changed User while DFCF-Changed User increase as the change threshold value increases. MAE values of TCF-Changed User are larger than those of DFCF-Changed User at all change threshold sizes. This indicates that the proposed model can improve the performance of the recommendation.

Table 5. Performance Results (MAE)

No. of cluster	Model			
	FCF	CCF	DFCF	TCF
c=2	0.19921	0.19921	0.19921	0.19921
c=3	0.19924	0.19935	0.19915	
c=4	0.19926	0.19945	0.19908	
c=5	0.19926	0.19964	0.19883	
c=6	0.19927	0.19991	0.19859	
c=7	**0.19926**	**0.20033**	**0.19831**	
c=8	0.19927	0.20103	0.19834	
c=9	0.19927	0.20187	0.19841	
c=10	0.19927	0.20271	0.19863	
c=15	0.19929	0.20621	0.19899	
c=20	0.19943	0.22301	0.19918	
c=30	0.1995	0.24821	0.19927	

Table 6. Cluster Centroid

genre	Cluster						
	1	2	3	4	5	6	7
Action	**0.75**	0.49	0.54	**0.66**	**0.6**	***0.32***	0.49
Animation	**0.73**	0.57	**0.69**	**0.61**	***0.24***	0.53	0.57
Art_Foreign	**0.76**	**0.65**	**0.68**	***0.28***	**0.66**	***0.25***	**0.66**
Classic	**0.86**	**0.7**	**0.77**	**0.76**	**0.77**	0.58	**0.71**
Comedy	**0.69**	0.45	0.56	0.52	0.51	***0.3***	0.45
Drama	**0.78**	0.59	**0.68**	0.58	**0.65**	***0.34***	0.59
Family	**0.74**	0.48	**0.67**	0.57	***0.39***	0.46	0.48
Horror	**0.75**	0.48	***0.29***	0.56	**0.61**	***0.14***	0.48
Romance	**0.73**	0.54	**0.72**	0.53	0.59	0.41	0.54
Thriller	**0.74**	0.52	0.54	**0.66**	**0.63**	***0.31***	0.52

Bold numbers mean likes and italicized and bold numbers are dislikes

Table 7. Paired t-test

	p-value			
	FCF	CCF	DFCF	TCF
FCF	.	0.033**	0.029**	0.031**
CCF		.	0.335	0.463
DFCF			.	0.428
TCF				.

** Significant at the .05 level

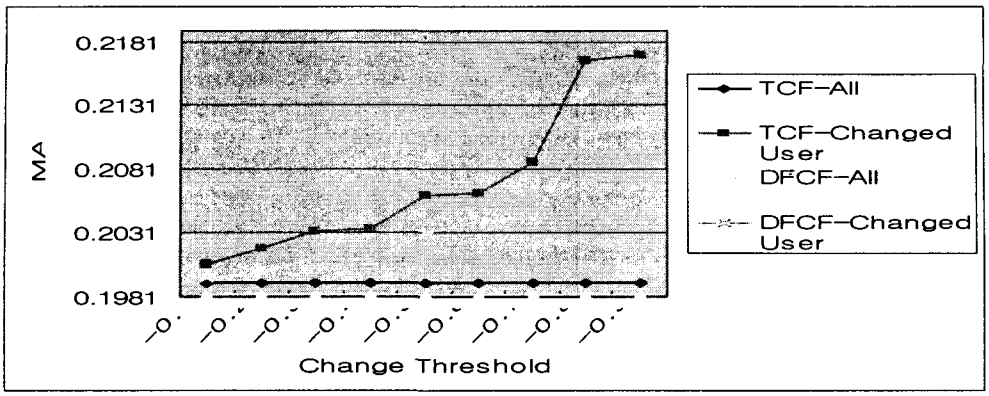

Fig. 1. Sensitivity of change threshold value

6 Conclusion

This paper proposed a new approach to CF based on the dynamic fuzzy cluster which can detect user's time-variant patterns and dynamically reflect this information for more accurate recommendations. We modified the FCM algorithm and applied it to CF for dynamic recommendations. We added a time dimension to the original input data of CF for finding the fuzzy cluster at different timeframes. We computed the dynamic degree of membership and determined the neighborhood for a given user based on dynamic fuzzy cluster.

We conducted an experiment to evaluate the proposed model on the EachMovie data set and compared them with the traditional CF algorithm. The results show the proposed model's improvement in making recommendations.

In this paper, we applied the concept of time to CF algorithm and proposed a new model which is adaptive to users' changing patterns. This research is expected to help marketer to catch customer's changing needs. In this paper, there are some limitations. First, we used only EachMovie data set for experiments. It is needed to evaluate our model using other data set. Second, we didn't consider change in the customer cluster structure. We assumed that cluster structure doesn't change in a short term.

In our future work, we intend to evaluate our model using other data set. We would also like to develop a model considering the change in cluster structure.

References

1. Bezdek, J.C., (1981). Pattern Recognition with Fuzzy Objective Function Algorithm. Plenum, New Your.
2. Breese, J.S., Heckerman, D., Kadie, C. (1998). Empirical Analysis of Predictive Algorighms for Collaborative Filtering. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98), pp. 43-52.
3. Herlocker, J.L., Konstan, J.A. and Riedl, J., (2000). Explaining collaborative filtering recommendations. Proceedings on the ACM 2000 Conference on Computer Supported Cooperative Work, (pp. 241–250). Philadelphia.
4. Sarwar,B.M., Konstan,J.A., Borchers,A., Herlocker,J.L., Miller,B.N., Ried1,J. (1998). Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. Proceedings of CSCW'98. Seattle, WA.
5. Schafer, J.B., Konstan, J.A. and Riedl, J. (2001). Electronic Commerce Recommender Applications. Data Mining and Knowledge Discovery 5(1/2), pp. 115-153.
6. Xie, X.L., Beni, G.A., (1991). Validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Machine Intell. 3(8), 841-846.
7. Yu, K.A., et al. (2000). Improving the performance of collaborative recommendation by using muli-level similarity computation. IASTED Interenational Conference on Artificial Intelligence and Soft Computing, July 2000.
8. http://www.research.compaq.com/SRC/eachmovie