Hybrid Personalized Recommender System Using Modified Fuzzy C-Means Clustering Algorithm

Subhash K. Shinde skshinde@rediffmail.com
Department of Computer Engineering,
Bharati Vidyapeeth College of Engineering,
Navi Mumbai 400 614, India.

Uday V. Kulkarni kulkarniuv@yahoo.com
Department of Computer Science and Engineering,
SGGS Institute of Engineering and Technology,
Nanded 431605, India

Abstract

Recommender Systems apply machine learning and data mining techniques for filtering unseen information and can predict whether a user would like a given resource. This paper proposes a novel Modified Fuzzy C-means (MFCM) clustering algorithm which is used for Hybrid Personalized Recommender System (MFCMHPRS). The proposed system works in two phases. In the first phase, opinions from the users are collected in the form of user-item rating matrix. They are clustered offline using MFCM into predetermined number clusters and stored in a database for future recommendation. In the second phase, the recommendations are generated online for active users using similarity measures by choosing the clusters with good quality rating. We propose coefficient parameter for similarity computation when weighting of the users' similarity. This helps to get further effectiveness and quality of recommendations for the active users. The experimental results using Iris dataset show that the proposed MFCM performs better than Fuzzy C-means (FCM) algorithm. The performance of MFCMHPRS is evaluated using Jester database available on website of California University, Berkeley and compared with fuzzy recommender system (FRS). The results obtained empirically demonstrate that the proposed MFCMHPRS performs superiorly.

Keywords: Fuzzy C-means, Modified Fuzzy C-means, Personalized Recommender System.

1. INTRODUCTION

Modern consumers are inundated with choices. Electronic retailers and content providers offer a huge selection of products with unprecedented opportunities to meet a variety of special needs and tastes. Matching consumers with the most appropriate products is the key to enhancing user satisfaction and loyalty. Therefore, more retailers have become interested in recommender systems, which analyze patterns of user interest in products to provide personalized recommendations that suit a user's taste. As good personalized recommendations can add another dimension to the user experience, e-commerce leaders like Amazon.com and Netflix have made recommender systems a salient part of their websites [1]. Such systems are particularly useful for entertainment products such as movies, music, jokes, and TV shows. Many customers will view the same movie, and each customer is likely to view numerous different movies. Customers have proven willing to indicate their level of satisfaction with particular movies, so a huge volume of data is available about which movies appeal to which customers. Companies can analyze this data to recommend movies to particular customers.

The remainder of this paper is organized as follows. The section 2 summarizes the different strategies for recommender systems and their drawbacks. The proposed clustering based hybrid personalized recommender system is described in the section 3. The section 4 illustrates experimental setup of the proposed recommendation system. This section also gives performance evaluation with the existing algorithms. Finally, the section 5 concludes the paper.

2. RECOMMENDER SYSTEM STRATEGIES

In the recent years web personalization has undergone through tremendous changes. The content [2, 3], collaborative [4, 5] and hybrid [6] based filtering are three basic approaches used to design recommendation systems.

The content based filtering [7] relies on the content of an item that user has experienced before. The content based information filtering has proven to be effective in locating text, items that are relevant to the topic using techniques such as Boolean queries, vector space queries etc. However, content based filtering has some limitations. It is difficult to provide appropriate recommendation because all the information is selected and recommended based on the content. Moreover, the content based filtering leads to overspecialization i.e. it recommends all the related items instead of the particular item liked by the user.

The collaborative-filtering [8] aims to identify users who have relevant interests and preferences by calculating similarities and dissimilarities between their profiles. The idea behind this method is that to one’s search the information collected by consulting the behavior of other users who shares similar interests and whose opinions can be trusted may be beneficial. The different techniques have been proposed for collaborative recommendation; such as correlation based method, semantic indexing etc. The collaborative filtering overcomes some of the limitations of the content based filtering. The system can suggest items to the user, based on the rating of items, instead of the content of the items which can improve the quality of recommendations. However, collaborative filtering has some drawbacks. The first drawback is that the coverage of rating could be very sparse thereby resulting in poor quality recommendation. In the case of the addition of new items into database, the system would not be able to recommend until that item is served to a substantial number of users known as cold-start. Secondly, when new users are added, the system must learn the user preferences from the rating of users, in order to make accurate recommendations. Moreover, these recommendation algorithms seem to be very extensive and grow non-linearly when the number of users and items in a database increase. The hybrid recommendation systems [9, 10, 11] combine content and collaborative based filtering to overcome these limitations. As stated below, there are different ways of combining content and collaborative based filtering [12].

i. Implementing these approaches separately and combining them for prediction.
ii. Incorporating some content based characteristics into collaborative approach and vice versa.
iii. Constructing a general unified model that incorporates both content and collaborative based characteristics.

The hybrid approach proposed in this paper extracts user’s current browsing patterns using web usage mining, and forms a cluster of items with similar psychology to obtain implicit users rating for the recommended item.

3. PROPOSED MFCMHPRS

We have developed and tested the MFCMHPRS for Jester dataset available on website of California University, Berkeley. The system architecture has been partitioned into two main phases; offline and online. The Fig. 1 depicts the architecture of MFCMHPRS with its essential components.

The phase I is offline. It does the preprocessing and clustering. In this phase background data in the form of user-item rating matrix is collected and clustered using the proposed approach which is described in section 3.1.2. Once the clusters are obtained the cluster data along with their centroids are stored for future recommendations. The phase II is online in which the recommendation takes place for the active user. Here, similarity between active users and clusters are calculated for choosing best clusters for making recommendations. The rating quality of each item unrated by active user is computed in the chosen clusters. To generate the recommendations, clusters are further selected based on rating quality of an item. The recommendations are then made by computing the weighted average of the rating of items in the selected clusters. The working of MFCMHPRS is described below in detail with the Jester dataset.

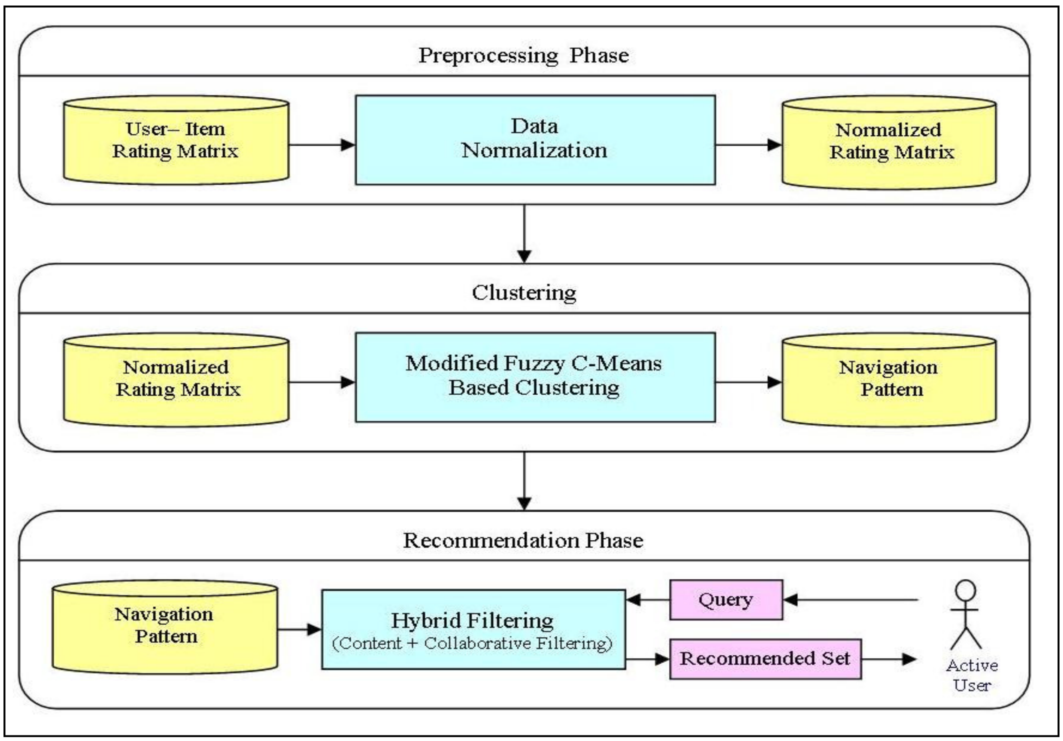

Figure 1: The architecture of MFCMHPRS

Preprocessing phase

3.1.1 Normalization of data

User-item rating taken from Jester dataset rated in the scale of -10 to +10 is normalized in the scale of 0 to 1, where 0 indicates that item is not rated by corresponding user. To facilitate the discussion, running example shown in the Table 1 is used, where U_1-U_{10} are the users and J_1-J_{10} are the items (jokes) rated or unrated by users. The last row of Table 2 gives ratings of the active user (U^1).

Users	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}
U_1	0.15	0.94	0.06	0.13	0.16	0.11	0.05	0.72	0.09	0.29
U_2	0.71	0.51	0.82	0.73	0.41	0.06	0.48	0.26	0.94	0.96
U_3	0.00	0.00	0.00	0.00	0.95	0.96	0.95	0.96	0.00	0.00
U_4	0.00	0.92	0.00	0.00	0.60	0.91	0.38	0.81	0.00	0.61
U_5	0.92	0.74	0.32	0.26	0.58	0.60	0.85	0.74	0.50	0.79
U_6	0.23	0.35	0.54	0.11	0.18	0.31	0.11	0.48	0.20	0.43
U_7	0.00	0.00	0.00	0.00	0.93	0.05	0.89	0.94	0.00	0.00
U_8	0.84	0.67	0.96	0.22	0.13	0.44	0.96	0.59	0.27	0.31
U_9	0.34	0.35	0.07	0.19	0.10	0.51	0.27	0.09	0.14	0.44
U_{10}	0.66	0.76	0.76	0.66	0.82	0.76	0.94	0.64	0.66	0.91
U^1	0.38	0.71	0.00	0.00	0.20	0.00	0.64	0.27	0.00	0.59

TABLE 1: Running example of rating matrix from Jester data set after normalization in the range of 0 to 1

3.1.2 Modified Fuzzy C-means Clustering

Fuzzy C-Means algorithm also known as Fuzzy ISODATA, was introduced by Bezdeck [13] as an extension to Dunn's algorithm [14]. The FCM- based is the most widely used fuzzy clustering algorithms in practice. However in FCM there are several constraints that affect the performance. The first limitation is the selection of random centorids at initial level. So the algorithm takes more time to find clusters. The second constraint is its inability to calculate the membership value if the distances of data point is zero. Whereas, the proposed MFCM algorithm initially calculates centorids appropriately and proposes new member function to calculate the membership value even if the distances of data point is zero.

Let $X = \{x_1, x_2, ..., x_n\}$ where $x_i \in \Re^n$ present a given set of feature data. The objective of MFCM algorithm is to minimize the cost function formulated as

$$J(U,V) = \sum_{j=1}^{C}\sum_{i=1}^{C}(\mu_{ij})^m \left\|x_i - v_j\right\|^2 \quad (1)$$

$V = \{v_1, v_2, ..., v_c\}$ are the cluster centers. The cluster centers are initially calculated as follows. To determine the centroid of the cluster, all the patterns are applied to each of the pattern and the patterns having Euclidian distance less than or equal to α (user defined value) are counted for all the patterns. Later the pattern with the maximum count is selected as the centroid of the cluster.

$$\text{If } \left(\left| R_i \overset{P}{\underset{j=1}{-}} R_j \right| \le \alpha \right) \text{ then } D_i = D_i + 1 \text{ for } i = 1, 2, \cdots, p. \quad (2)$$

If D_{max} is the maximum value in the row vector *D and* D_{ind} is the index of maximum value

$$[D_{max} D_{ind}] = max[D] \quad C_1 = R_{ind}$$

For instance the most appropriate centorids at the initial level are using centering process to form three clusters of running example shown in the Table 1 are $V = \{U_1, U_6, U_7\}$.

$U = (\mu_{ij})_{N \times C}$ is fuzzy partition matrix, in which each member μ_{ij} indicates the degree membership between the data vector x_i and cluster j. The values of matrix U should satisfy the following conditions

$$\mu \in [0,1], \forall i = 1 \cdots, N \text{ , } \forall j = 1 \cdots, C \tag{3}$$

$$\sum_{j=1}^{C} \mu i_j = 1, \forall i = 1 \cdots, N \tag{4}$$

Appropriate initialize of the membership matrix U using

$$f(x,v,r) = 1 - f(.) \quad , \tag{5}$$

where $f(.) = \begin{cases} 1 & \text{if } r \geq 1 \\ 0 & \text{if } r = 0 \\ r\gamma & \text{if } 0 < r < 1 \quad (\gamma \text{ is sensitive parameter}) \end{cases}$,

$r = \left\| x_i - v_j \right\|, r \geq 1$, and if $r > 1$ then $r\gamma$ is set to 1.

To satisfy the condition 2 and 3, divide the total sum of attributes to the each attribute for every pattern.

The exponent $m \in [1, \infty]$ is the weighting exponent which determines the fuzziness of the clusters. Minimization of the cost function $J[U,V]$ is nonlinear optimization problem, which can be minimized with following iterative algorithm:

Step 1: Find appropriate centriods using equation (2).
Choose appropriate exponent m and termination criteria.

Step 2: Initialize the membership matrix U using equation (4)

Step 3: Calculate the cluster center V according to the equation:

$$v_j = \frac{\sum_{i=1}^{N} (\mu_{ij})^m x_{ij}}{\sum_{i=1}^{N} (\mu_{ij})^m}, \ \forall j = 1 \cdots, C \tag{6}$$

Step 4: Calculate new distances norm:

$$r = \left\| x_i - v_j \right\|, \ \forall i = 1 \cdots, N \text{ }, \forall j = 1 \cdots, C$$

Step 5: Update the fuzzy partition matrix U :

If $r > 0$ (indicating that $(x_i \neq v_j)$)

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{r_{ij}}{r_{ik}}\right)^{\frac{2}{m-1}}} \quad (7)$$

Else

$$\mu i_j = 1$$

Step 6: If the termination criteria has been met, stop.
Else go to step 2.

A suitable termination criterion could be to evaluate the cost function (Eq. 1) and to see whether it is below a certain tolerance value or if its improvement compared to the previous iteration is below a certain threshold. Also the maximum number of iteration cycles can be used as a termination criterion.

3.1.3 Computing centorid of each cluster

The proposed MFCM is used for clustering of the Jester data set. The clustering is resulted in the three clusters with $\alpha = 0.9$ and $\varepsilon = 0.01$ (ε termination criteria). The details of the clusters are created and users in each cluster are shown in the Table 3. After clustering as stated in the MFCM algorithm, knowing the members of each group, we have recomputed new centroids of each cluster. As an example the cluster 3 has two members. Thus the centorid is the average of all corresponding coordinates of the two members

C_3= {(0.00+0.00)/2, (0.00+0.00)/2, (0.00+0.00)/2, (0.00+0.00)/2, (0.60+0.93/2), (0.91+0.05/2), (0.00+0.00)/2, (0.00+0.00)/2, (0.00+0.00)/2, (0.00+0.00/2)}. Similarly, we have calculated the centroids of the cluster 1 and 2.

Cluster No.	Users	Centroid
1	U_2, U_6, U_8, U_{10}	C_1
2	U_1, U_4, U_5, U_9	C_2
3	U_3, U_7	C_3

TABLE 2: Users in each cluster with the centroid

Recommendation phase

This phase consists of two steps: (i) find the nearest neighbors and (ii) produce recommendations set.

Find the nearest neighbors:

In order to find the nearest neighbors of the active user, it must measure the similarity of the users. Calculate similarity between clusters centroids and active users. Select cluster that have the highest similarity. We use cosine similarity algorithm to measure the similarity between active user u^1 and cluster c_u. User rating can be treated as a vector on an n-dimensional item space. Assuming the rating of the n-dimensional item space rated by U^1 and C_u is respectively vector $\overline{u^1}$ and $\overline{c_u}$.

$$sim(u^1, c_u) = cos(\overline{u^1}, \overline{c_u}) = \frac{\overline{u^1} \cdot \overline{c_u}}{\left\|\overline{u^1}\right\| \cdot \left\|\overline{c_u}\right\|}$$

In most cases, the number of items usually one or two jointly rated by two users is few. Even the rating of these items rated by the two users has high similarity. According to common sense we can not judge the two users are similar; but the semblance of the two users is very high if we use traditional similarity measurement method. In order to solve this problem, we introduce a coefficient: the coefficient is large if there are many items that the two users jointly rate; on the contrary, the coefficient is small. We suppose that the coefficient is k given as

$$k = \frac{\cap(u^1, c_u)}{\cup(u^1, c_u)}.$$

$\cap(u^1, c_u)$ is represents the number of items in the intersection set that rated both by user u^1 and c_u , $\cup(u^1, c_u)$ represents the number of items in the union set that rated both by user u^1 and c_u. The range of k is in between 0 and 1.
Hence the similarity between active users and cluster centriods computed as:

$$sim(u^1, c_u) = cos(\overline{u^1}, \overline{c_u}) = \frac{\overline{u^1} \cdot \overline{c_u}}{\left\|\overline{u^1}\right\| \cdot \left\|\overline{c_u}\right\|} * k \qquad (8)$$

For running example, the similarity value of active user of three clusters is shown in Table 3. Choose the clusters having high similarity value.

	Cluster1	**Cluster 2**	**Cluster 3**
$sim(u^1, c_u)$	0.53	1.13	1.25
k	0.25	0.39	0.80
$sim(u^1, c_u) * k$	0.1325	0.4407	1.00

TABLE 3: Nearest neighbors of the active user

Produce recommendation data set

The predication rating of item *i* by u^1 is $P_{u^1}(i)$ which is gained by the rating of nearest neighbors set c_u rated by active user u^1, the computation method is as the follows:

$$P_{u^1}(i) = \overline{R_{u^1}} + \frac{\sum_{u^1 \in c_u} sim(u^1, c_u) * (\overline{R}_u(i) - R_{u^1}(i))}{\sum_{u^1, c_u} (|\, sim(u^1, c_u)|)} \qquad (9)$$

where $\overline{R_{u^1}}$ - average rating of items rated by active user u^1;

$sim(u^1, c_u)$- Similarity between active user and users clusters;

$\overline{R}_u(i)$ - average rating of item i rated by all user;

$R_{u^1}(i)$- rating of active user of item i.

According to the rating of items, we select N items (N is user define parameter) that have the highest rating to compose recommendation set and recommend them to active users. The predication rating of active user for running example is shown in Table 4.

	Jokes	Predicating Rating
Active User (u^1)	J_3	0.73
	J_4	0.43
	J_6	0.55
	J_9	0.42

TABLE 4: Predication rating for active user

Once the quality rating of each item is calculated, the recommendation to the active user is provided, e.g., joke 3 predication rating up to 0.73 will be recommended and so on.

4. EXPERIMENTS

We have conducted a set of experiments to examine the effectiveness of our proposed recommender system in terms of accuracy of neighbor-selection, cold start and recommendation quality. In particular, we addressed the following issues [15, 16, 17, 18, 19].

i. How does the confidence parameter affect the performance of the prediction? In this paper, we have conducted few experiments to show the accuracy of the prediction for different settings of the parameter values.
ii. How does the neighbor-selection method affect the efficiency of prediction? Experiments are conducted to examine the accuracy of MFCM algorithm for neighbor-selection.
iii. How do the clusters formed influence the prediction accuracy? Experiments are conducted to examine the impact of clustering methods on the final performance of item or user content based collaborative filtering.
iv. The performance MFCMHPRS is evaluated and compared with FRS using Precision, Recall.

The proposed MFCMHPRS is implemented in MATLAB version 7.2. The experiments are conducted on a 2.0 GHz, Intel Pentium 4 PC with 512 MB memory, running Microsoft Windows XP Professional.

4.1 Simulation results and performance evaluation

4.1.2 Performance evaluation of clustering

In order to check the performance of the proposed clustering algorithm, we have first applied the algorithm to real data set, ‘Iris’ data, whose true classes are known. The Iris data set is available in UCI repository (ftp://www.ics.uci.edu/ pub/machinelearningdatabases/), which includes 150 objects (50 in each of three classes – ‘Setosa’, ‘Versicolor’, and ‘Virginica’) having four variables (‘sepal length’, ‘sepal width’, ‘petal length’, and ‘petal width’).

The performance was measured by the accuracy, which is the proportion of objects that are correctly grouped together against the true classes. To investigate the performance more objectively, a simulation study was carried out by generating artificial data sets repetitively and calculating the average performance of the method.

We have applied the proposed MFCM, and FCM to create three clusters using this data without the class information. The table 5 shows the result obtained using existing and proposed clustering method.

Algorithms	Setosa	Versicolor	Virginica	Computational Time
FCM	50	34	66	13.5790 seconds
MFCM	50	39	61	11.3790 seconds

TABLE 5: Cluster result of Iris data by the proposed and traditional methods

The table 7 shows that the proposed MFCM clustering algorithm works superior than the traditional algorithms because the algorithm calculates centroids and Initialize the membership matrix properly instead of selecting randomly.

4.1.2 Performance evaluation of recommender system

The Jester dataset is available online on the site www.ieor.berkeley.edu/~goldberg/jester-data. The Jester is a WWW Based Joke Recommender System, developed by University of California, Berkeley. This data has 73421 user entered numeric rating for 100 jokes, ranging on real value scale from -10 to 10. The experiments are performed on the small Jester dataset consisting of user-item rating matrix of size 100 (users) $\times$10 (jokes) as shown in the Table 1.

The measurement method of evaluating the recommendation quality of recommendation system mainly includes statistical precision measurement method and decision supporting precision measurement method [20, 21]. Statistical precision measurement method adopts MAE (Mean Absolute Error) to measure the recommendation quality [22]. MAE is a commonly used recommendation quality measurement method. So we use MAE as the measurement criteria.

MAE calculates the irrelevance between the recommendation value predicted by the system and the actual evaluation value rated by the user. We represent each pair of interest predicted rank as $<p_i, q_i>$, p_i is the system predicted value, q_i is the user evaluation value. Basing on the entire $<p_i, q_i>$ pairs, MAE calculates the absolute error value $|p_i-q_i|$ and the sum of all the absolute error value, and then calculates their average value. If the MAE value is small, it indicates good recommendation quality.

The predicted user rating set can be represented as $\{p_1, p_2, \cdots, p_N\}$, its corresponding actual user rating set can be represented as $\{q_1, q_2, \cdots, q_N\}$, the MAE can be defined as the following[23]:

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{10}$$

In addition to MFCMHPRS, FRS [24] is also implemented to compare the performance with our proposed system. Let us examine the influence of various nearest neighbor set on predictive validity. We gradually increase the number of neighbors; the experiment result is shown in Table 6:

Size of Neighbor Set	*MAE*	
	FRS	MFCMHPRS
04	1.3272	1.2213
08	1.2531	1.2163
12	1.2615	1.2182
16	1.2480	1.2203
20	1.2573	1.2232

Table 6: Influence of various size of nearest neighbor set on predictive validity

As Fig. 2 shown, the MFCMHPRS has smaller MAE value than FRS in most cases, which means that the sparseness has the less impact on our proposed algorithm.

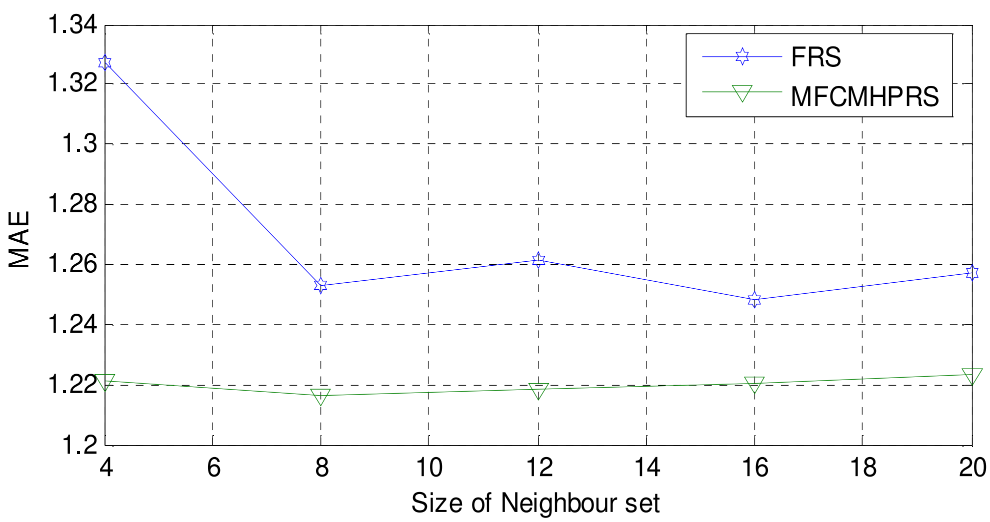

Figure: 2 MAE on each algorithm. (A small value means a better performance)

5. CONCULSIONS

This paper describes a novel fuzzy personalized recommender system that utilizes clustering of user-item rating matrix through proposed MFCM and provides the recommendations for the active user with good quality rating using similarity measures. The results from various simulations using Iris data set shows that the proposed MFCM clustering algorithm performs better than FCM clustering, which helps to improve the quality of rating. Through the experiment analysis, it is found that the proposed MFCMHPRS performs better than FRS and the sparseness has less impact on the proposed system.

REFERENCES

1. G. Adomavicius and A. Tuzhilin. "*Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions".* IEEE Trans. Knowledge and Data Eng., 17(6): 734–74, 2005.

2. R. B. Allen. *"User models: theory, method and Practice".* International Journal of Man–Machine Studies, 43(11): 27-52, 1990.

3. D. Kalles, A. Papagelis and C. Zaliagis. "*Algorithmic aspects of web intelligent systems".* Web Intelligence, Springer, Berlin, 323–345, 2003.

4. J. L. Herlocker, J.A. Konstan, L. G. Terveen and J. T. Riedl. "*Evaluating collaborative filtering recommender systems".* ACM Trans. on Information Systems (TOIS), 22(1): 5-53, 2005.

5. T. Hofmann. "*Collaborative filtering via Gaussian probabilistic latent semantic analysis".* In Proceedings of the 26th Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 145-153, 2003.

6. M. Balabanovic and Y. Sholam. *"Combining content-based and collaborative recommendation".* Comm. ACM, 40(3): 23-43, 1997.

7. Chun Zeng and et al. "*Personalized Services for Digital Library"*. In Proceeding of 5th International Conference on Asian Digital Libraries, pp. 252-253, 2002.

8. G. Ulrike and F. R. Daniel. *"Persuasion in recommender systems."* Int'l Journal of Electronic Commerce, 11(2): 81-100, 2006.

9. S. K. Shinde and U. V. Kulkarni. "*A New Approach for on Line Recommender System in Web Usage Mining".* In Proceeding Inte'l Conference on Advanced Computer Theory and Engineering, pp. 312-317, 2008.

10. S. K. Shinde and U.V. Kulkarni. *"The hybrid web personalized recommendation based on web usage mining".* International Journal of Data Mining, Modeling and Management, 2(4): 315-333, 2010.

11. K. Yu, A. Schwaighofer and H. P. Kriegel*. "Probabilistic Memory-Based Collaborative Filtering".* IEEE Trans. Knowledge and Data Engineering,16(1) :56-69, 2004.

12. K.W. Cheung and Ch. Tsui, "*Extended latent class models for collaborative recommendation".* IEEE Trans. on Systems, Man and Cybernetics-Part A: Systems and Humans, 34(1): 143-148, 2004.

13. J. Bezdek. *"Pattern Recognition with Fuzzy Objective Function Algorithms".* Plenum Press, USA, 1981.

14. J.C. Dunn. *"A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact, Well Separated Clusters".* Journal of Cybernetics, 3(3): 32-57, 1974.

15. Y. H. Choa and J. K. Kimb*. "Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce".* Expert Systems with Applications, 26(2): 233-246, 2004.

16. B. M. Kim and Q. Li*. "A new approach for combining content-based and collaborative filters".* Journal of Intelligent Information Systems, 27(1), 79-91, 2006.

17. S. Vucetic. *"Collaborative filtering using a regression-based approaches".* Journal of Knowledge and Information Systems, 7(1): 22-35, 2005.

18. G. L. Somlo and A. Howel, "*Adaptive Lightweight Text Filtering".* In proceedings Fourth Int'l Symp. Intelligent Data Analysis, pp. 53-59, 2001.

19. Z. Huang and W. Chung. *"A graph model for e-commerce recommender systems"*. Journal of the American Society for Information Science and Technology, 55(3): 259–274, 2001.

20. D. Billsus and M. *"Learning collaborative information filter."* In Proceeding 5th International conference on Machine Learning, pp. 46-54, 1998.

21. C. H. Basu and et al. *"Recommendation as classification: Using social and content–based information in recommendation".* In Proceeding 15th International conference on Artificial Intelligence, pp. 714-720, 1998.

22. Huihong Zhou, Yijun Liu, Weiqing Zhang, and Junyuan Xie. "*A Survey of Recommender System Applied in E-commerce".* Computer Application Research, 1(1): 8-12, 2004.

23. Z. Zhao and S. Bing. *"An Adaptive Algorithm for Personal Recommendation".* Journal of Changchun University, 1(1): 22-29, 2005.

24. L. Teran and Andreas Meier. "*A Fuzzy Recommender system for eElections".* EGOVIS 2010, LNCS, Springer-Berlin, 2(2):62-76, 2010.