2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

A Method for Hybrid Personalized Recommender based on Clustering of Fuzzy User Profiles

Shan XU
Graduate School of Information, Production & Systems
Waseda University
2-7 Hibikino, Wakamatsu, Kitakyushu 808-0135 JAPAN
Email: shinexushan@gmail.com

Junzo WATADA
Graduate School of Information, Production & Systems
Waseda University
2-7 Hibikino, Wakamatsu, Kitakyushu 808-0135 JAPAN
Email: junzo.watada@gmail.com

***Abstract*—Personalized Recommenders can help to find potential items and then recommend them for particular users. Conventional recommender methods always work on a rating schema that items are rated from 1 to 5. However, there are several rating schemas (ways that items are rated) in reality, which are overlooked by conventional methods. By transforming rating schemas into fuzzy user profiles to record users' preferences, our proposed method can deal with different system rating schemas, and improve the scalability of recommender systems. Additionally, we incorporate user-based method with item-based collaborative methods by clustering users, which can help us to gain insight into the relationship between users. The aim of this research is to provide a new method for personalized recommendation. Our proposed method is the first to normalize the user vectors using fuzzy set theory before the k-medians clustering method is adjusted, and then to apply item-based collaborative algorithm with item vectors. To evaluate the effectiveness of our approach, the proposed algorithm is compared with two conventional collaborative filtering methods, based on MovieLens data set. As expected, our proposed method outperforms the conventional collaborative filtering methods as it can improve system scalability while maintaining accuracy.**

I. Introduction

The rapid development of web technology provides us with a burgeoning amount of data and information. As a result, users are hard to distinguish necessary information from an incredible amount of information. In a traditional way, most developers or sellers would like to recommend an attractive item for each user based on known data or experiences. With the advancement of computer and information technologies, people build recommender systems, which can automatically predict the preference or rating score for items, and then recommend personalized popular items, contents, or service for each user.

In recent years, recommender systems have become extremely common and are widely applied in a variety of fields, such as movies, books, music, mobile applications, and products. Among them, famous examples include recommending books and products by Amazon.com [1], recommending movies at MovieLens [2], and recommending news by AdaptiveInfo.com [3].

Recommender systems have come into our view since the collaborative filtering algorithm was put forward in 1995 [4]. As one of the most popular recommendation algorithms, User-based collaborative filtering methods are based on k nearest neighbor (KNN) to find the closest users. They were very successful in the past, but become inefficient when facing large or sparse database. Comparing with user-based methods, item-based collaborative filtering methods attempt to discover the similarity relations between items instead of the most k similar users. Owing to the relatively static relationship between items, item-based methods compared with user-based methods may be able to figure out similar items with less online computation time. As a result, our proposed approach aims to build on item-based collaborative filtering methods.

Item-based collaborative filtering methods are based on some similar items to predict unknown items for a target user. These similar items are calculated on users who rated both of these items. And this calculation may involve the whole set of users, which consumes much computation time. So we apply clustering method to users to find groups of roughly similar users and potential relationships between users, which may have been ignored by item-based methods. Although such similar groups have been used in collaborative filtering methods in the past, clustering methods for recommendation systems concentrate on grouping items or items and users, except for users. Our proposed approach combines user-clustering methods with item-based collaborative filtering methods, which investigates both users aspect and item aspect.

The truth is, the far majority of recommendation algorithm research is focused on the application of movies, which uses rating score (positive integer from 1 to 5) to represent the preference degree of users. However, there are different ways to show users' preference in reality. For example, Twitter and Facebook use "like" button, and Apple store encourages customers to rate from one-star to five-star. As traditional recommender systems overlooked this problem, we use fuzzy set theory to transform different rating schemas (two-value, linguistic description, or in a certain range) into membership grade rating profiles. In fuzzy profile, the degree of users' preferences is shown by real number within [0, 1]. Using fuzzy profile can help to deal with different rating schemas, and improve the scalability of recommender systems.

The remaining of this paper is organized as follows. In Section 2, we review the history and related technologies for recommender systems (content-based filtering, demographic filtering, collaborative filtering and hybrid filtering), as well as clustering methods, following by the basic concepts of fuzzy set. Section 3 not only describes the methodology for our proposed method, but also explains the procedural steps for realizing the system. In Section 4, we introduce the dataset-MovieLens, and describe measures for evaluating the performance of methods. At the same time, the results of comparing our proposed method with the two other conventional recommendation methods are shown, too. Finally, Section 5 draws conclusions to our work along with the future works for improving our proposed method.

II. RELATED WORK

A. Recommender Algorithms

Now much work has been done both in the industry and the academia on developing new approaches to recommendation, as the demand for the customers' personalized recommendation increases. According to [5,6,7], recommendation technologies are usually divided into the following four categories.

1) Demographic filtering: Demographic information (e.g. age, gender, occupation, nationality, etc.) may be used to discover similar users with certain common demographic features. This method advocated: Individuals with certain common personal attributes will have common preferences [8].

2) Content-based filtering: Content-based filtering methods were raised by Lang in 1995 [9]. In these methods, recommender systems focus on the relationship between items, which are described by keywords. According to the keywords of items, candidate items are calculated the similarity with items chosen by users in the past. Finally, items with high similarities and good scores are recommended to users.

However, content-based recommender systems limit types of recommended items. For example, if a user saw a romantic film in the past, a movie recommender system would never recommend action films for this user. In other words, users are also limited to these items, which keep similar keywords or characteristics associated with items chosen by users in the past. The content-based recommender system information cannot go beyond the past.

3) Collaborative filtering: As content-based filtering methods do not consider the relationship between items and users, collaborative filtering methods are built on user-item matrix, which records preferences for all items for users. Collaborative filtering methods [10] have been the most widely used recommender algorithms so far. These methods are justified on the principle that similar users have similar tastes and preferences. Compared with content-based filtering methods, collaborative filtering methods try to discover users with same interests. Similar users are valued by the similarity on ratings of items, which have been rated. So predicting rating scores for items has become a primary problem for collaborative filtering methods.

4) Hybrid filtering: Hybrid filtering methods develop collaborative filtering methods with content-based filtering or demographic filtering. Another way is to incorporate probabilistic methods, such as genetic algorithms [11], fuzzy genetic [12], neural networks [13], Bayesian networks [14], and clustering [15] into collaborative filtering approaches. In our proposed personalized recommender method, we combine clustering of fuzzy user profiles with collaborative filtering methods.

A widely accepted taxonomy divides collaborative filtering recommendation methods into memory-based (user-based) and model-based (item-based) categories [16].

5) Memory-based (User-based) Collaborative Filtering Algorithms: Memory-based collaborative filtering algorithms apply the whole user-item collection in dataset to make predictions. The systems find out some most similar users, known as nearest neighbor search. Above the similar users, these systems employ an aggregate of the preferences of these similar users to predict rating scores of unknown items. Memory-based (User-based) collaborative filtering algorithms are also named neighbor-based collaborative filtering algorithms.

6) Model-based (Item-based) Collaborative Filtering Algorithms: In contrast to memory-based algorithms, model-based collaborative filtering methods apply an existing dataset to training a model, which is then used to make predictions for ratings of unknown items. A probabilistic approach has been used to model-based methods. And other model-based collaborative recommendation approaches include statistical model [17], Bayesian network model [18], probabilistic relational model [19], a linear regression [20], probabilistic latent semantic indexing [21], and a maximum entropy model [22].

B. Clustering on Collaborative Filtering

Clustering is used to divide a large data set into clusters according to their similarity so that it can shorten runtime and improve performance [23].In the same way, we can use clustering methods to group users or items for reducing the size of data set and finding similar users or items.

As mentioned above, recommender systems have applied clustering to collaborative filtering methods as hybrid methods for filtering, which heighten the prediction quality and reduce the cold-start problem [8]. In hybrid filtering methods, clustering typically works on items [15] or both items and users (bi-clustering) [24].

C. Fuzzy set

Fuzzy set theory is also known as possibility theory. A fuzzy set consists of elements that have degrees of membership. Rather than a crisp ridged two-value (1 represents that item belongs to particular set, otherwise 0), the degree of belonging to a certain category is between 0 and 1.

Definition 1: Given a set of objects, X, a fuzzy set, S, is a subset of X that allows each object in X to have a membership degree between 0 and 1. Formally, a fuzzy set, S, can be modeled as a function, FS: X $\rightarrow$ [0,1].[23]

III. PROPOSED METHOD

Facing problems out of large sparse data set, how to keep accuracy and efficiency for recommender systems becomes a big challenge. Here, we propose a hybrid method for personalized recommender, which is based on clustering and fuzzy set theory. The proposed process consists of four phases, as shown in Fig. 1.

We use user-item matrix as the input, which is presented as user profiles. A user profile records particular user' s preferences for some items in a database. In phase 1, we

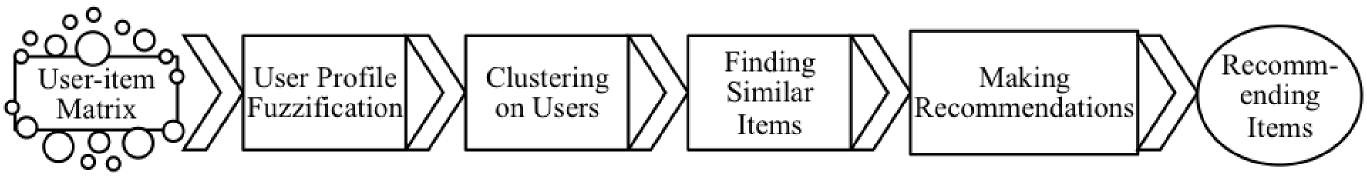

Fig. 1. The four phases of our proposed approach

transform user-item matrix to fuzzy user profiles by fuzzing rating records. After getting fuzzy user profiles, phase 2 is clustering user vectors. Our proposed method is built on item-based collaborative filtering method, which is trying to discover similar items to predict unknown items for a target user. Phases 3 and 4 build the item-based collaborative filtering methods and make recommendations. The four phases are explained in details in the following:

A. Phase 1: User profile Fuzzification

The number of user and items in database are separately set as m and n. So the input is a $m \times n$ user-item matrix. Let R be the set of rating scores. And a rating schema means the way that rating score is represented, in which ratings can be non-negative integers or real numbers within a certain range. For example, YouTube uses ”like” or ”dislike” to rate, which corresponds to 1 or 0 in a two-value rating schema. Amazon encourages customers to rate using one-star to five-star, which means rating score 1 to 5 in another rating schema. Here, the value of R indicates the degree of users’ preferences for items. The higher score that item gets, the more like that user presents.

According to Definition 1 in subsection 2.3, we can use (1) to compute the preference degree that the target user likes a particular item. Within the scope of rating score, R_{down} and R_{up} are chose as lower and upper bounds for the rating set R, R_{down}, $R_{up} \in N$(Natural number). So the membership function of rating scores for fuzzy set, $\widetilde{R}$, can be denoted as

$$\widetilde{R}(x) = \begin{cases} 0 & \text{if} \quad x \leq R_{down} \\ \frac{p}{q} & \text{if} \quad R_{down} < x < R_{up} \\ 1 & \text{if} \quad R_{up} \leq x \end{cases} \tag{1}$$

where x is a rating record in R, p is the number of records that rating score is smaller than x in training dataset, and q is the number of all rating records in training dataset. Corresponding degree of users’ affection for items is shown in Fig. 2 .

In the same way, we can get the membership function of rating scores, which indicate the degree for users’ negative feelings for items. In general, Fig. 3 shows the membership degree of users’ affection (short for ”like” in the figure) or negative feeling (short for ”dislike”) for items.

As recommender systems are to discover potential items that interest users, we focus on the membership function that indicates the preference degree that a target user likes a particular item, which is shown in Fig. 3. The set of fuzzy user profiles in recommender systems is defined as $U = \{\overrightarrow{U_1}, \overrightarrow{U_2}, \cdots, \overrightarrow{U_m}\}$, while set of all items in the dataset

Fig. 2. The membership degree of users’ affection for items

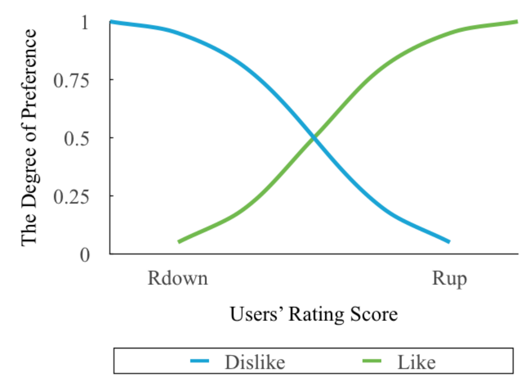

Fig. 3. The membership degree of users’ affection or negative feeling for items

is defined as $I = \{\overrightarrow{I_1}, \overrightarrow{I_2}, \cdots, \overrightarrow{I_n}\}$, in which items could be movies, songs, restaurants, or mobile applications. In fuzzy user profiles, corresponding user vector of user t can be presented as $\overrightarrow{U_t} = (\widetilde{R}_{t,1}, \widetilde{R}_{t,2}, \cdots, \widetilde{R}_{t,n})$, where $t = 1, 2, \cdots, m$. Table 1 shows the result of user profile fuzzification.

TABLE I. USER- ITEM MATRIX

	$\overrightarrow{I_1}$	$\overrightarrow{I_2}$	$\cdots$	$\overrightarrow{I_i}$	$\cdots$	$\overrightarrow{I_n}$
$\overrightarrow{U_1}$	$\widetilde{R}_{1,1}$	$\widetilde{R}_{1,2}$	$\cdots$	$\widetilde{R}_{1,i}$	$\cdots$	$\widetilde{R}_{1,n}$
$\overrightarrow{U_2}$	$\widetilde{R}_{2,1}$	$\widetilde{R}_{2,2}$	$\cdots$	$\widetilde{R}_{2,i}$	$\cdots$	$\widetilde{R}_{2,n}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\overrightarrow{U_t}$	$\widetilde{R}_{t,1}$	$\widetilde{R}_{t,2}$	$\cdots$	$\widetilde{R}_{t,i}$	$\cdots$	$\widetilde{R}_{t,n}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\overrightarrow{U_m}$	$\widetilde{R}_{m,1}$	$\widetilde{R}_{m,2}$	$\cdots$	$\widetilde{R}_{m,i}$	$\cdots$	$\widetilde{R}_{m,n}$

B. Phase 2: Clustering of Fuzzy User Profiles

In the research history of recommender systems, most clustering methods concentrate on finding groups of similar items or working on items and users, except for only users. Normally, there are thousands of users in a dataset. However, most of them have little bearing on predicting a target item' s score. Therefore, reducing the number of users is helpful to ignore noisy data, and improve the accuracy or efficiency of systems. In the meantime, clustering of users can help to discover potential relationship between users, and to focus on a particular set of clusters which includes the target user.

In our proposed method, k-medians clustering (a variation of k-means clustering) method is chose to find a particular set that includes a target user. In phase 1, we get the fuzzy user profile for target user, $\overrightarrow{U_t}$.

Step 1: Decide the number of clusters, k, and arbitrarily choose k users from user set U. There are k clusters in set $C = \{C_1, C_2, \cdots, C_k\}$, where corresponding medians are $c = \{\overrightarrow{c_1}, \overrightarrow{c_2}, \cdots, \overrightarrow{c_k}\}$.

Step 2: Assign each user of U to a cluster, which the user is most similar to its median based on Manhattan distance. As a result, we can get k clusters.

Step 3: Find a new point in each cluster, from which the sum of the distances to the remaining points in this cluster is the smallest. The new point would be new median for each cluster. At the same time, using (2) to keep record of the sum of each point to the cluster' s median that it belongs to, Distance, for C.

$$\text{Distance} = \sum_{\overrightarrow{U_i} \in \mathrm{U}} \min_{\overrightarrow{c_j} \in \mathrm{C}} \mathrm{d}(\overrightarrow{U_i}, \overrightarrow{c_j}) \tag{2}$$

where $i = 1, 2, \cdots, m$; $j = 1, 2, \cdots, k$.

Step 4: Return to step 2 until there is no changes between the current distance value and previous value.

Step 5: We get the final set of clusters, C. In the set C, target cluster C_t containing the target user t is chose, which consists of a users. Here, $C_t \subset U$; $0 < a \leq m - k + 1$; $a \in N$. $C_t = \{\overrightarrow{U_{t1}}, \overrightarrow{U_{t2}}, \cdots, \overrightarrow{U_{tt}}, \cdots, \overrightarrow{U_{ta}}\}$, where $t1$, $t2$, tt, $ta \in \{1, 2, \cdots, m\}$.

C. Phase 3: Finding Similar Items

In phase 2, our proposed method clusters users before we realize a collaborative filtering method.

In the usual application of model-based algorithms, models need to be built offline, and then work online. Because the relationships between items are relatively static, item-based algorithms may be able to provide the same quality as the user-based algorithms with less online computation [20]. Our proposed approach is built on item-based collaborative filtering.

Step1: In phase 2, we get the cluster set C_t consisted of a users, which contains the target user t: For the sake of brevity, the target set, C_t, is redefined: $C_t = \{\overrightarrow{u_1}, \overrightarrow{u_2}, \cdots, \overrightarrow{u_t}, \cdots, \overrightarrow{u_a}\}$, where $C_t \subset U$; $0 < a < m - k + 1$; $a \in N$. Now the original input matrix $n \times m$ has been reduced to $a \times n$, which is composed of a users and n items. For the new fuzzy user profiles, User t is renamed $\overrightarrow{u_t} = (r_{t,1}, r_{t,2}, \cdots, r_{t,n})$, $t \in \{1, 2, \cdots, a\}$. We do the same to item i, $\overrightarrow{I_i} = (r_{1,i}, r_{2,i}, \cdots, r_{a,i})$, $i \in \{1, 2, \cdots, n\}$.

Step 2: We define the rating score given by user t for item i:

$$r_{t,i} = \begin{cases} r_{t,i} & \text{if } u_t \text{ rated item } i \\ p_{t,i} & \text{otherwise} \end{cases} \tag{3}$$

Step 3: Let the set of users who rated both items i and j, denoted it as T. $i, j \in \{\mathrm{item_1}, \mathrm{item_2}, \cdots, \mathrm{item_n}\}$.

Step 4: Here, we use the adjusted cosine similarity [18] to measure the item similarity between object target item i and item j as follows. $sim(i, j)$ is to present the value of similarity function measuring the similarity between items i and j.

$$sim(i,j) = \frac{\sum_{t \in T} (r_{t,i} - r_t) \times (r_{t,j} - r_t)}{\sqrt{\sum_{t \in T} (r_{t,i} - r_t)^2} \times \sqrt{\sum_{t \in T} (r_{t,j} - r_t)^2}} \tag{4}$$

where $r_{t,i}$ means the value of target user t on item i, so does $r_{t,j}$. And r_t is the average rating for the user t.

Step 5: As a result, we can get similarity set between target item i and other items: $sim1 = \{sim(i, 1), \cdots, sim(i, i-1), sim(i, i+1), \cdots, sim(i, n)\}$.

Step 6: The elements in this similarity set $sim1$ are ordered in descending, and define a threshold value θ. If the similarity element in $sim1$ is larger than θ, it will be chosen as an optional candidate to the set $sim2$. Finally, $sim2$ is the set of most similar items for object item i.

D. Phase 4: Making Recommendations

In item-based collaborative filtering method, the rating value of unknown items is computed as an aggregate of the ratings of other similar items. In phase 3, we get the aggregate of the most similar items, $sim2$. The next step is to apply a prediction formula to calculate the scores for unknown items.

Sarwar *et al.* [20] show two widely used techniques for prediction computation, which are weighted sum and regression. However, weighted sum does not consider that different users may use the rating scale differently. The parameters in regression method are hard to decide and may have a great influence on the result. As a result, we choose adjusted weighted sum [18] to predict. First, multiplier b serves as a normalizing factor and is usually selected as (5),

$$b = \frac{1}{\sum_{l \in sim2} |sim(i,l)|} \tag{5}$$

where item l belongs to the similarity set $sim2$ for target item i.

Afterwards the prediction for user t on item i is calculated by (6)

$$p_{t,i} = r_i + b \sum_{l \in sim2} sim(i,l) \times (r_{t,l} - r_l) \tag{6}$$

where the respective average rating scores of items i and l are defined as r_i and r_l.

Then the top N recommending items are chose after getting all unknown items for the target user. According to the rating score calculated as above, we can get the rating vector for user t: $\overrightarrow{u_t} = (r_{t,1}, r_{t,2}, \cdots, r_{t,n})$.

The elements in vector u_t are sorted in descending. Within those elements, the first N items are chose to recommend.

IV. EXPERIMENTAL EVALUATION AND RESULTS

To evaluate the effectiveness of our approach, we conducted an experimental study using a MovieLens dataset. The proposed algorithm was compared with two conventional collaborative filtering algorithms respectfully, which were user-based collaborative filtering methods (short for "user-based CF") and collaborative filtering based on traditional clustering methods (short for "CF based on k-means").

A. Data Set

The data were obtained from a public dataset MovieLens (http://www. movielens.umn.edu). MovieLens is a virtual web-based movie recommender system, where users can visit and rate movies. From September 19th, 1997 to April 22nd, 1998, MovieLens web site collected 100,000 ratings, which were from 943 users on 1682 movies. All of the users have rated at least 20 movies. In order to calculate, we represented the dataset as user-item matrix, where items and users represent as columns and rows, and the value of each record corresponds to a rating value.

B. Evaluation Metrics

In this experiment, we used Mean Absolute Error (MAE) to make quality comparisons for our proposed method and conventional well-known methods. MAE measures the quality of predictions for recommender systems. The lower MAE is, the better recommender systems perform.

We defined the user set of dataset as U, and the items set of dataset as I, $r_{i,j}$ is the rating of user i on item j. For particular user t, the prediction value of items are defined as $\{p_{t,1}, p_{t,2}, \cdots, p_{t,n}\}$, and real value of items are $\{r_{t,1}, r_{t,2}, \cdots, r_{t,n}\}$.

The $MAE(t)$ [16, 25] is given by formula (7):

$$MAE(t) = \frac{\sum_{i=1}^{n} |p_{t,i} - r_{t,i}|}{n} \tag{7}$$

where n is the cardinality of the test ratings set of user t.

The average MAE for whole testing data set is defined as MAE in (8):

$$MAE = \frac{\sum_{i=1}^{N_u} MAE(i)}{N_u} \tag{8}$$

where Nu is the cardinality of the test set of users, with $t \in \{1, 2, \cdots, N_u\}$.

C. Experimental Results

Some certain parameters have a great effect on the performance of our proposed method, which will be shown in the following experiment. Our experiments is divided into two parts:

1) The number of clusters: In conventional clustering methods, deciding the number of clusters, k, is one of the major sticking points. In order to find out an optimal value for k, we proposed the number of clusters as independent variable in this experiment. The value of k was taken as 5, 10, 15, 20, 25, 30, 35, and 40. Here, we conducted a study that compared CF based on $k-$means to our proposed method, with MAE measuring the performance. The result is shown in Fig. 4.

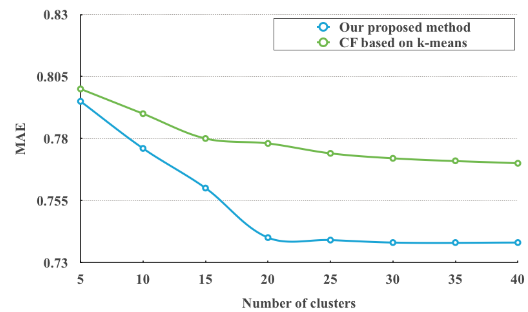

Fig. 4. Comparison for different number of clusters

First, we concentrated on the line for our own method' s performance in Fig. 4. Before the number of clusters, k, rose to 20, the value of MAE dropped rapidly. That is to say, the performance of our method improves sharply with the cluster number increasing before k reaches a certain number. When k reached 20, our method continued to maintain the same MAE. Hence, we selected the certain number of cluster, 20, for the next part of our experiment.

Now we turn our attention to the two lines in Fig. 4. At the start point where k is 5, CF based on k-means methods and our proposed method almost obtained the same MAE. With k increasing, the two methods both had the downward trends. However, our proposed method had a sharper downward trend before k reached the certain number. Afterward there were obvious differences between our proposed method and CF based on k-means methods, when the number of clusters was larger than 20.

2) Sensitivity of Neighborhood Size: From fig.4, we chose an optimum value for cluster number as 20. Finally the results are shown in Fig. 5. In step 6 of phase 3, threshold value θ decides the size of $sim2$, which acts as the set of neighborhood for target item. Small threshold value θ leads to a large set of similar items, which are known as neighborhood items. In traditional KNN method, the number of neighborhood has significant impact on the prediction quality. In order to compare with the conventional user-based methods, we changed the neighborhood size of our proposed method by choosing different threshold value θ.

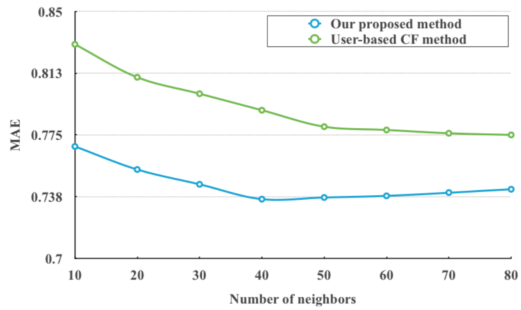

Fig. 5. Sensitivity of the neighborhood size at a selected cluster number

This second experiment conducted several numbers of neighbors between 10 and 80. As Fig. 5 showed, our proposed method had a lower MAE than user-based CF methods over the entire range, which meant our proposed method was superior to the conventional one. However, there were some differences could be found between their trends. In Fig. 5, user-based CF methods had lower MAE when the number of neighbors became larger. Our proposed method had a downward trend before the number of neighbors reached 40, and then had a tiny upward trend in the latter half.

3) Discussion: In experimental phase, we made performance measurements from two parameters k and θ, which were separately observed as the number of clusters and sensitivity of the neighborhood size. In the first experiment, the MAE of our method dropped fast at the beginning and then kept smooth after k reach a certain value. Therefore, increasing the number of clusters in a certain range is helpful to improve the performance of our method. A sharper downward trend than CF based on $k-$means methods in Fig. 4 is a proof for the success of our method. By analyzing variation trend of k in Fig. 4, we could select an optimum value of cluster number meanwhile, which was used in the second experiment. Here, the number of neighbors meant the size of user set, which was used in phase 3. The trend for our method in Fig. 5 serves to bring a fact to our attention: when the set of users is too small, the rating records cannot be predicted accurately. On the other hand, it may consume much computation time with a large set of users, when the number of neighbors is larger than a certain value. In general, our proposed method is sure to win out over conventional collaborative filtering methods.

V. Conclusion

In this paper, we propose a hybrid personalized recommender method based on clustering of fuzzy user profile. The experiments show that our method is superior to the conventional collaborative filtering methods. Here, we apply the clustering of user profile to item-based collaborative filtering method, which can combine items' and users' characteristics. A considerable improvement has been made in accuracy. Furthermore, fuzzy set theory helps the system to understand different rating schemas.

Much work on recommender systems has been done in not only the academic but also the research area. A variety of approaches were proposed. However, most research studies focus on application of movies. In our following research, we would try to apply our proposed method to a different field, such as application in mobile phone, or combine recommendation with mobile phone location.

Acknowledgment

This work was supported partly by Grants-in-Aid for Science Research MEXT (No23500289) and also in part under a Waseda University Grant for Special Research Projects (Project number 2014A-040).

References

[1] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76-80, 2003.

[2] B. Miller, I. Albert, S.Lam , J. Konstan, and J. Riedl "MovieLens unplugged: experiences with an occasionally connected recommender system," *ACM Conference on Intelligent User Interfaces*, pp. 263-266, 2003.

[3] D. Billsus, C. A. Brunk, C. Evans, B. Gladish, and M. Pazzani, "Adaptive interfaces for ubiquitous web access," *Commun. ACM*, vol. 45, no. 5, pp. 34-38, 2002.

[4] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, vol. 1, pp. 194-201, 1995.

[5] G. Adomavicius and A. Tuzhilin "Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Trans. on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 739-749, 2005.

[6] L. Candillier, F. Meyer, M. Boulle, "Comparing state-of-the-art collaborative filtering systems," *Machine Learning and Data Mining in Pattern Recognition*, vol. 4571, pp. 548-562, 2007.

[7] J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, "Collaborative filtering recommender systems," *The Adaptive Web*, pp. 291-324, 2007.

[8] J.Bobadilla, F.Ortega, A.Hernando, and A.Gutierrez, "Recommender systems survey," *Knowledge-Based Systems 46*, pp. 109-132, 2013.

[9] K. Lang, "NewsWeeder: Learning to filter netnews," *Proceedings of the 12th International Conference on Machine Learning*, Tahoe City, CA, pp. 331-339, 1995.

[10] U. Shardanand and P. Maes "Social information filtering: Algorithms for automating "word of mouth," *Proceedings of 1995 Conference on Human Factors in Computing Systems (CHI-95)*, pp. 210 -217, 1995.

[11] L. Q. Gao, C. Li, "Hybrid personalized recommended model based on genetic algorithm," *The 4th International Conference on Wireless Communication, Networks and Mobile Computing*, pp. 9215-9218, 2008.

[12] M.Y.H. Al-Shamri, K.K. Bharadwar, "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1386-1399, 2008.

[13] M. Lee, P. Choi, and Y. Woo, "A hybrid recommender system combining collaborative filtering with neural network," *Proceeding of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*, pp. 531-534, 2002.

[14] L.M. Campos, J.M. Fernandez-Luna, J.F. Huete, and M.A. Rueda-Morales, "Combining content-based and collaborative recommendations: a hybrid approach based on Bayesian Networks," *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 785-799, 2010.

[15] S.K. Shinde, U. Kulkami, "Hybrid personalized recommender system using centering-bunching based clustering algorithm," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1381-1387, 2012

[16] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 43-52, 1998.

[17] L.H. Ungar, D. P. Foster, "Clustering methods for collaborative filtering," *Proceedings of the Workshop on Recommendation Systems, Technical Report WS-98-08*, AAAI Press, Menlo Park California, 1998.

[18] Y. -H. Chien and E.I. George, "A Bayesian model for collaborative filtering," *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*, 1999.

[19] L. Getoor and M. Sahami. "Using probabilistic relational models for collaborative filtering," *In Workshop on Web Usage Analysis and User Profiling (WEBKDD'99)*, August 1999.

[20] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," *Proceedings of the 10th International WWW Conference*, pp. 285-295, 2001.

[21] Si, L. and R. Jin, "Flexible mixture model for collaborative filtering," *Proceedings of the 20th International Conference on Machine Learning, Washington*, D.C., pp. 704-711, 2003.

[22] D. Y. Pavlov, D. M. Pennock, "A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains," *Proceedings of the 16th Annual ?Conference on Neural Information Processing Systems (NIPS'02)*, pp. 1441-1448, 2002.

[23] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*, San Francisco, CA, Morgan Kaufmann, ch. 9, sec.11.1.1, pp. 499, 2001.

[24] R.L. Zhu, S.J. Gong, "Analyzing of collaborative filtering using clustering technology," *ISECS International Colloquium on Computing, Communication, Control, and Management*, pp. 57-59, 2009.

[25] E. Vozalis, K.G. Margaritis, "Analysis of recommender systems' algorithms," *Proceedings of the 6th Hellenic-European Conference on Computer Mathematics and its Applications (HERCMA)*, Athens, Greece, 2003.