A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering

SongJie Gong
Zhejiang Business Technology Institute, Ningbo 315012, China
Email: songjie_gong@sina.com

***Abstract*—Personalized recommendation systems can help people to find interesting things and they are widely used with the development of electronic commerce. Many recommendation systems employ the collaborative filtering technology, which has been proved to be one of the most successful techniques in recommender systems in recent years. With the gradual increase of customers and products in electronic commerce systems, the time consuming nearest neighbor collaborative filtering search of the target customer in the total customer space resulted in the failure of ensuring the real time requirement of recommender system. At the same time, it suffers from its poor quality when the number of the records in the user database increases. Sparsity of source data set is the major reason causing the poor quality. To solve the problems of scalability and sparsity in the collaborative filtering, this paper proposed a personalized recommendation approach joins the user clustering technology and item clustering technology. Users are clustered based on users' ratings on items, and each users cluster has a cluster center. Based on the similarity between target user and cluster centers, the nearest neighbors of target user can be found and smooth the prediction where necessary. Then, the proposed approach utilizes the item clustering collaborative filtering to produce the recommendations. The recommendation joining user clustering and item clustering collaborative filtering is more scalable and more accurate than the traditional one.**

***Index Terms*—recommender systems, collaborative filtering, user clustering, item clustering, scalability, sparsity, mean absolute error**

I. INTRODUCTION

As the development of the internet, intranet and electronic commerce systems, there are amounts of information arrived we can hardly deal with. Thus, personalized recommendation services exist to provide us the useful data employing some information filtering technologies. Information filtering has two main methods. One is the content based filtering and the other is the collaborative filtering. Collaborative filtering (CF) has proved to be one of the most effective for its simplicity in both theory and implementation [1,2].

Many researchers have proposed various kinds of CF technologies to make a quality recommendation. All of them make a recommendation based on the same data structure as user-item matrix having users and items consisting of their rating scores. There are two methods in CF as user based collaborative filtering and item based collaborative filtering [3,4]. User based CF assumes that a good way to find a certain user's interesting item is to find other users who have a similar interest. So, at first, it tries to find the user's neighbors based on user similarities and then combine the neighbor users' rating scores, which have previously been expressed, by similarity weighted averaging. And item based CF fundamentally has the same scheme with user based CF. It looks into a set of items; the target user has already rated and computes how similar they are to the target item under recommendation. After that, it also combines his previous preferences based on these item similarities. The challenge of these two CF as following [5,6]:

Sparsity: Even as users are very active, there are a few rating of the total number of items available in a user-item ratings database. As the main of the collaborative filtering algorithms are based on similarity measures computed over the co-rated set of items, large levels of sparsity can lead to less accuracy.

Scalability: Collaborative filtering algorithms seem to be efficient in filtering in items that are interesting to users. However, they require computations that are very expensive and grow non-linearly with the number of users and items in a database.

Cold-start: An item cannot be recommended unless it has been rated by a number of users. This problem applies to new items and is particularly detrimental to users with eclectic interest. Likewise, a new user has to rate a sufficient number of items before the CF algorithm be able to provide accurate recommendations.

To solve the problems of scalability and sparsity in the collaborative filtering, in this paper, we proposed a personalized recommendation approach joins the user clustering technology and item clustering technology. Users are clustered based on users' ratings on items, and each users cluster has a cluster center. Based on the similarity between target user and cluster centers, the nearest neighbors of target user can be found and smooth the prediction where necessary. Then, the proposed approach utilizes the item clustering collaborative filtering to produce the recommendations. The recommendation joining user clustering and item clustering collaborative filtering is more scalable and more accurate than the traditional one.

doi:10.4304/jsw.5.7.745-752

II. TRADITIONAL COLLABORATIVE FILTERING ALGORITHM

A. User Item Rating Content

The task of the traditional collaborative filtering recommendation algorithm concerns the prediction of the target user's rating for the target item that the user has not given the rating, based on the users' ratings on observed items. And the user-item rating database is in the central. Each user is represented by item-rating pairs, and can be summarized in a user-item table, which contains the ratings Rij that have been provided by the ith user for the jth item, the table as following [7,8].

TABLE I
USER-ITEM RATINGS TABLE

Item / User	Item1	Item2	… …	Itemn
User1	R11	R12	… …	R1n
User2	R21	R22	… …	R2n
… …	… …	… …	… …	… …
Userm	Rm1	Rm2	… …	Rmn

Where Rij denotes the score of item j rated by an active user i. If user i has not rated item j, then Rij =0. The symbol m denotes the total number of users, and n denotes the total number of items.

B. Measuring the Rating Similarity

Collaborative filtering approaches have been popular for both researchers and practitioners alike evidenced by the abundance of publications and actual implementation cases. Although there have been many algorithms, the basic common idea is to calculate similarity among users using some measure to recommend items based on the similarity. The collaborative filtering algorithms that use similarities among users are called user based collaborative filtering [9,10].

A set of similarity measures are presented and a metric of relevance between two vectors. When the values of these vectors are associated with a user's model then the similarity is called user based similarity, whereas when they are associated with an item's model then it is called item based similarity. The similarity measure can be effectively used to balance the ratings significance in a prediction algorithm and therefore to improve accuracy.

There are several similarity algorithms that have been used in the collaborative filtering recommendation algorithm [1,3]: Pearson correlation, cosine vector similarity, adjusted cosine vector similarity, mean-squared difference and Spearman correlation.

Pearson's correlation, as following formula, measures the linear correlation between two vectors of ratings.

$$sim(i,j)=\frac{\sum_{c\in I_{ij}}(R_{i,c}-A_i)(R_{j,c}-A_j)}{\sqrt{\sum_{c\in I_{ij}}(R_{i,c}-A_i)^2\sum_{c\in I_{ij}}(R_{j,c}-A_j)^2}} \quad (1)$$

Where Ri,c is the rating of the item c by user i, Ai is the average rating of user i for all the co-rated items, and Iij is the items set both rating by user i and user j.

The cosine measure, as following formula, looks at the angle between two vectors of ratings where a smaller angle is regarded as implying greater similarity.

$$sim(i,j)=\frac{\sum_{k=1}^{n}R_{ik}R_{jk}}{\sqrt{\sum_{k=1}^{n}R_{ik}^{\ 2}\sum_{k=1}^{n}R_{jk}^{\ 2}}} \quad (2)$$

Where Rik is the rating of the item k by user i and n is the number of items co-rated by both users. And if the rating is null, it can be set to zero.

The adjusted cosine, as following formula, is used in some collaborative filtering methods for similarity among users where the difference in each user's use of the rating scale is taken into account.

$$sim(i,j)=\frac{\sum_{c\in I_{ij}}(R_{ic}-A_c)(R_{jc}-A_c)}{\sqrt{\sum_{c\in I_{ij}}(R_{ic}-A_c)^2*\sum_{c\in I_{ij}}(R_{jc}-A_c)^2}} \quad (3)$$

Where Ri,c is the rating of the item c by user i, Ac is the average rating of user i for all the co-rated items, and Ii,j is the items set both rating by user i and user j.

Literature provides rich evidence on the successful performance of collaborative filtering methods. However, there are some shortcomings of the methods as well. Collaborative filtering methods are known to be vulnerable to data sparsity and to have cold-start problems. Data sparsity refers to the problem of insufficient data, or sparseness. Cold-start problems refer to the difficulty of recommending new items or recommending to new users where there are not sufficient ratings available for them.

C. Selecting Neighbors

Select of the neighbors who will serve as recommenders. Two techniques have been employed in the collaborative filtering recommender systems.

Threshold-based selection, according to which users whose similarity exceeds a certain threshold value are considered as neighbors of the target user.

The top-n technique, n-best neighbors is selected and the n is given at first.

D. Producing Prediction

Since we have got the membership of user, we can calculate the weighted average of neighbors' ratings, weighted by their similarity to the target user.

The rating of the target user u to the target item t is as following:

$$P_{ut} = A_u + \frac{\sum_{i=1}^{c} (R_{it} - A_i) * sim(u,i)}{\sum_{i=1}^{c} sim(u,i)} \quad (4)$$

Where A_u is the average rating of the target user u to the items, Rit is the rating of the neighbour user i to the target item t, Am is the average rating of the neighbour user i to the items, sim(u, i) is the similarity of the target user u and the neighbour user i, and c is the number of the neighbours.

III. Related Works

L.H. Ungar et al. [11,12] present a formal statistical model of collaborative filtering and compare different algorithms for estimating the model parameters including variations of K-means clustering and Gibbs Sampling. This formal model is easily extended to handle clustering of objects with multiple attributes. And it is better than the traditional one.

M.O. Conner [13] reports on work in progress related to applying data clustering algorithms to ratings data in collaborative filtering. They use existing data partitioning and clustering algorithms to partition the set of items based on user rating data. Predictions are then computed independently within each partition. Ideally, partitioning will improve the quality of collaborative filtering predictions and increase the scalability of collaborative filtering systems. They report preliminary results that suggest that partitioning algorithms can greatly increase scalability, but they have mixed results on improving accuracy. However, partitioning based on ratings data does result in more accurate predictions than random partitioning, and the results are similar to those when the data is partitioned based on a known content classification.

A. Kohrs et al. [14] identify two important situations with sparse ratings in the collaborative filtering recommendation systems. Bootstrapping a collaborative filtering system with few users and providing recommendations for new users who rated only few items. Further, they present a novel algorithm for collaborative filtering based on hierarchical clustering which tries to balance robustness and accuracy of predictions and experimentally show that it is especially efficient in dealing with the previous situations.

Lee, WS et al. [15] study two online clustering methods for collaborative filtering. In the first method, they assume that each user is equally likely to belong to one of m clusters of users and that the user's rating for each item is generated randomly according to a distribution that depends on the item and the cluster that the user belongs to. In the second method, they assume that each user is equally likely to belong to one of m clusters of users while each item is equally likely to belong to one of n clusters of items. And the result is that the proposed methods are good in some way.

The rating for a user item pair is generated randomly according to a distribution that depends on the cluster that the user belongs to and the cluster that the item belongs to. They derive performance bounds for Bayesian sequential probability assignment for these two methods to elucidate the trade offs involved in using these methods. Bayesian sequential probability assignment does not appear to be computationally tractable for these model classes. They propose heuristic approximations to Bayesian sequential probability assignment for the model classes and performed experiments on a movie rating data set. The proposed algorithms are fast, perform well and the results of the experiments agree with the insights derived from the theoretical considerations.

Automated collaborative filtering is a popular technique for reducing information overload. K Honda et al. [16] propose a new approach for the collaborative filtering using local principal components. The new method is based on a simultaneous approach to principal component analysis and fuzzy clustering with an incomplete data set including missing values. In the simultaneous approach, they extract local principal components by using lower rank approximation of the data matrix. The missing values are predicted using the approximation of the data matrix. In numerical experiment, they apply the proposed technique to the recommendation system of background designs of stationery for word processor.

S.H.S. Chee et al. [17] develop an efficient collaborative filtering method, called RecTree that addresses the scalability problem with a divide-and-conquer approach. The method first performs an efficient k-means-like clustering to group data and creates neighborhood of similar users, and then performs subsequent clustering based on smaller, partitioned databases. Since the progressive partitioning reduces the search space dramatically, the search for an advisory clique will be faster than scanning the entire database of users. In addition, the partitions contain users that are more similar to each other than those in other partitions. This characteristic allows RecTree to avoid the dilution of opinions from good advisors by a multitude of poor advisors and thus yielding a higher overall accuracy. Based on they experiments and performance study, RecTree outperforms the well-known user based collaborative filtering, in both execution time and accuracy. In particular, RecTree's execution time scales by O(nlog2(n)) with the dataset size while the traditional user based collaborative filtering recommendation scales quadratically.

B. Sarwar et al. [18] address the performance issues by scaling up the neighborhood formation process through the use of clustering techniques.

The high cardinality and sparsity of a collaborative recommender's dataset is a challenge to its efficiency. D. Bridge et al. [19] generalize an existing clustering technique and apply it to a collaborative recommender's dataset to reduce cardinality and sparsity. They systematically test several variations, exploring the value of partitioning and grouping the data.

J. Kelleher et al. [20] present a collaborative recommender that uses a user-based model to predict user ratings for specified items. The model comprises summary rating information derived from a hierarchical clustering of the users. They compare their algorithm with several others. They show that its accuracy is good and its coverage is maximal. They also show that the proposed algorithm is very efficient: predictions can be made in time that grows independently of the number of ratings and items and only logarithmically in the number of users.

Xue, G. et al. [21] present a novel approach that combines the advantages of memory based collaborative filtering and model based collaborative filtering of approaches by introducing a smoothing-based method. In their approach, clusters generated from the training data provide the basis for data smoothing and neighborhood selection. As a result, they provide higher accuracy as well as increased efficiency in recommendations. Their empirical studies on two datasets as EachMovie and MovieLens show that their new proposed approach consistently outperforms other user based traditional collaborative filtering algorithms.

George, T. et al. [22] consider a novel collaborative filtering approach based on a recently proposed weighted co-clustering algorithm that involves simultaneous clustering of users and items. They design incremental and parallel versions of the co-clustering algorithm and use it to build an efficient real-time collaborative filtering framework. Their empirical evaluation of the proposed approach on large movie and book rating datasets demonstrates that it is possible to obtain accuracy comparable to that of the correlation and matrix factorization based approaches at a much lower computational cost.

Rashid, A.M. et al. [23] propose ClustKnn, a simple and intuitive algorithm that is well suited for large data sets. The proposed method first compresses data tremendously by building a straightforward but efficient clustering model. Recommendations are then generated quickly by using a simple Nearest Neighbor-based approach. They demonstrate the feasibility of ClustKnn both analytically and empirically. They also show, by comparing with a number of other popular collaborative filtering algorithms that, apart from being highly scalable and intuitive, ClustKnn provides very good recommender accuracy as well.

Cantador, I. et al. [24] propose a multilayered semantic social network model that offers different views of common interests underlying a community of people. The applicability of the proposed model to a collaborative filtering system is empirically studied. Starting from a number of ontology-based user profiles and taking into account their common preferences, they automatically cluster the domain concept space. With the obtained semantic clusters, similarities among individuals are identified at multiple semantic preference layers, and emergent, layered social networks are defined, suitable to be used in collaborative environments and content recommenders.

Panagiotis Symeonidis et al. [25, 26] use bi-clustering to disclose this duality between users and items, by grouping them in both dimensions simultaneously. They propose a novel nearest bi-clusters collaborative filtering algorithm, which uses a new similarity measure that achieves partial matching of users' preferences. They apply nearest bi-clusters in combination with two different types of bi-clustering algorithms Bimax and xMotif for constant and coherent biclustering, respectively. Extensive performance evaluation results in three real-life data sets are provided, which show that the proposed method improves substantially the performance of the CF process.

IV. Rating Smoothing Based on User Clustering

A. User Clustering

User clustering techniques work by identifying groups of users who appear to have similar ratings. Once the clusters are created, predictions for a target user can be made by averaging the opinions of the other users in that cluster. Some clustering techniques represent each user with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. Once the user clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller [18].

The idea is to divide the users of a collaborative filtering system using user clustering algorithm and use the divide as neighborhoods, as Figure 1 show. The clustering algorithm may generate fixed sized partitions, or based on some similarity threshold it may generate a requested number of partitions of varying size.

	Item-1	Item-2	...	Item-n
User-1	R11	R12		R1n
User-2	R21	R22		R2n
...				
User-m	Rm1	Rm2		Rmn

user clustering

	Item-1	Item-2	...	Item-n
center1	a11	a12		a1n
center2	a21	a22		a2n
...				
centerc	ac1	ac2		acn

Figure1. Collaborative filtering based on user clustering.

Where Rij is the rating of the user i to the item i, aij the average rating of the user center i to the item i, m is the number of all users, n is the number of all items, and c is the number of user centers.

B. Smoothing

In this paper, we user the k means clustering algorithm to cluster the users into some groups as clustering centers.

Specific algorithm as follows:

Input: clustering number k, user-item rating matrix
Output: smoothing rating matrix
Begin
Select user set U={U1, U2, …, Um};
Select item set I={I1, I2, …, In};
Choose the top k rating users as the clustering CU={CU1, CU2, …, CUk};
The k clustering center is null as c={c1, c2, …, ck};
do
for each user Ui∈U
for each cluster center CUj∈CU
calculate the sim(Ui, CUj);
end for
sim(Ui, CUm)=max{sim(Ui, CU1), sim(Ui, CU2), …, sim(Ui, CUk);
cm=cm∪Ui
end for
for each cluster ci∈c
for each user Uj∈U
CUi=average(ci, Uj);
end for
end for
while (C is not change)
End

C. New Ratings

One of the challenges of the collaborative filtering is the data sparsity problem. To prediction the vacant values in user-item rating dataset, we make explicit use of item clusters as prediction mechanisms.

Based on the item clustering results, we apply the prediction strategies to the vacant rating data as follows:

$$R_{ij} = \begin{cases} R_{ij} & if\ user\ i\ rate\ the\ item\ j \\ c_j & else \end{cases} \quad (5)$$

Where cj denotes the prediction value for user i rating towards an item j and cj has calculated in above specific algorithm.

V. Using The Item Clustering Method to Produce Recommendations

Through the calculating the vacant user's rating by user clustering algorithm, we gained the dense users' ratings. Then, to generate prediction of a user's rating, we use the item clustering based collaborative filtering algorithms.

A. The dense user-item matrix

After we used the user clustering algorithm, we gained the dense ratings of the users to the items. So, the original sparse user-item rating matrix is now becoming the dense user-item matrix.

B. Item Clustering

Item clustering techniques work by identifying groups of items who appear to have similar ratings. Once the clusters are created, predictions for a target item can be made by averaging the opinions of the other items in that cluster. Some clustering techniques represent each item with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. Once the item clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller.

The idea is to divide the items of a collaborative filtering system using item clustering algorithm and use the divide as neighborhoods, as Figure 2 show. The clustering algorithm may generate fixed sized partitions, or based on some similarity threshold it may generate a requested number of partitions of varying size.

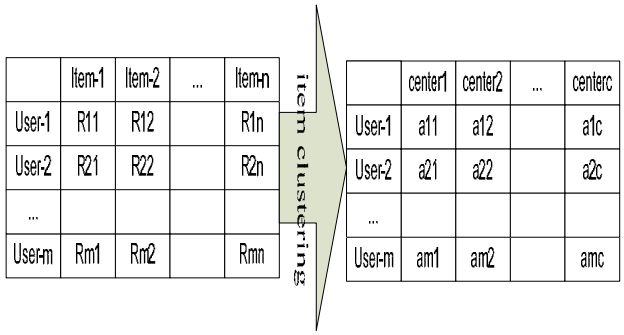

Figure2. Collaborative filtering based on item clustering.

Where Rij is the rating of the user i to the item i, aij the average rating of the user i to the item center j, m is the number of all users, n is the number of all items, and c is the number of item centers.

C. Algorithm

There are many algorithms that can be used to create item clustering. In this paper, we choose the k means algorithm as the basic clustering algorithm. The number k is an input to the algorithm that specifies the desired number of clusters. Firstly, the algorithm takes the first k items as the centers of k unique clusters. Each of the remaining items is then compared to the closest center. In the following passes, the cluster centers are re-computed based on cluster centers formed in the previous pass and the cluster membership is re-evaluated.

Specific algorithm as follows:

Input: clustering number k, user-item rating matrix
Output: item-center matrix
Begin
Select user set U={U1, U2, …, Um};
Select item set I={I1, I2, …, In};
Choose the top k rating items as the clustering CI={CI1, CI2, …, CIk};
The k clustering center is null as c={c1, c2, …, ck};
do
for each item Ii∈I

```
        for each cluster center CIj∈CI
          calculate the sim(Ii, CIj);
        end for
        sim(Ii, CIx)=max{sim(Ii, CI1), sim(Ii, CI2), …, sim(Ii, CIk);
        cx=cx∪Ii
      end for
      for each cluster ci∈c
        for each user Ij∈I
          CIi=average(ci, Ij);
        end for
      end for
  while (CU and c is not change)
End
```

We use the pearson's correlation, as following formula, to measure the linear correlation between two vectors of ratings as the target item t and the remaining item r.

$$sim(t,r) = \frac{\sum_{i=1}^{m}(R_{it} - A_t)(R_{ir} - A_r)}{\sqrt{\sum_{i=1}^{m}(R_{it} - A_t)^2 \sum_{i=1}^{m}(R_{ir} - A_r)^2}} \quad (6)$$

Where R_{it} is the rating of the target item t by user i, R_{ir} is the rating of the remaining item r by user i, A_t is the average rating of the target item t for all the co-rated users, A_r is the average rating of the remaining item r for all the co-rated users, and m is the number of all rating users to the item t and item r.

D. Selecting Clustering Centers

An important step of item based collaborative filtering algorithm is to search neighbors of the target item. Traditional memory based collaborative filtering is to search the whole ratings database and it suffers from poor scalability when more and more users and items are added into the database [21].

When we cluster the items, we get the items centers. This center is represented as an average rating over all items in the cluster. So we can choose the target item neighbors in some of the item center clustering. We use the Pearson's correlation to the similarity between the target item and the items centers.

After calculating the similarity between the target item and the items centers, we take the items in the most similar centers as the candidates.

E. Selecting Neighbors

After we select the target item nearest clustering centers, we also need to calculate the similarity between the target item and items in the selected clustering centers.

We select the Top K most similar items based on the cosine measure, as following formula, which looks at the angle between two vectors of ratings as the target item t and the remaining item r.

$$sim(t,r) = \frac{\sum_{i=1}^{m} R_{it} R_{ir}}{\sqrt{\sum_{i=1}^{m} R_{it}^2 \sum_{i=1}^{m} R_{ir}^2}} \quad (7)$$

Where R_{it} is the rating of the target item t by user i, R_{ir} is the rating of the remaining item r by user i, and m is the number of all rating users to the item t and item r.

F. Producing Recommendations

Since we have got the membership of item, we can calculate the weighted average of neighbors' ratings, weighted by their similarity to the target item.

The rating of the target user u to the target item t is as following:

$$P_{ut} = \frac{\sum_{i=1}^{c} R_{ui} \times sim(t,i)}{\sum_{i=1}^{c} sim(t,i)} \quad (8)$$

Where Rui is the rating of the target user u to the neighbour item i, sim(t, i) is the similarity of the target item t and the neighbour it user i for all the co-rated items, and m is the number of all rating users to the item t and item r.

VI. EXPERIMENT RESULTS

In this section, we describe the dataset, metrics and methodology for the comparison between traditional and proposed collaborative filtering algorithm, and present the results of our experiments.

A. Data Set

We use MovieLens collaborative filtering data set to evaluate the performance of proposed algorithm. MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota and MovieLens is a web-based research recommender system that debuted in Fall 1997. Each week hundreds of users visit MovieLens to rate and receive recommendations for movies [3,27]. The site now has over 45000 users who have expressed opinions on 6600 different movies. We randomly selected enough users to obtain 100, 000 ratings from 1000 users on 1680 movies with every user having at least 20 ratings and simple demographic information for the users is included. The ratings are on a numeric five-point scale with 1 and 2 representing negative ratings, 4 and 5 representing positive ratings, and 3 indicating ambivalence.

B. Performance Measurement

Several metrics have been proposed for assessing the accuracy of collaborative filtering methods. They are divided into two main categories: statistical accuracy metrics and decision-support accuracy metrics. In this paper, we use the statistical accuracy metrics [28,29].

Statistical accuracy metrics evaluate the accuracy of a prediction algorithm by comparing the numerical

deviation of the predicted ratings from the respective actual user ratings. Some of them frequently used are mean absolute error (MAE), root mean squared error (RMSE) and correlation between ratings and predictions. All of the above metrics were computed on result data and generally provided the same conclusions. As statistical accuracy measure, mean absolute error is employed.

Formally, if n is the number of actual ratings in an item set, then MAE is defined as the average absolute difference between the n pairs. Assume that p1, p2, p3, ..., pn is the prediction of users' ratings, and the corresponding real ratings data set of users is q1, q2, q3, ..., qn. See the MAE definition as following:

$$MAE = \frac{\sum_{i=1}^{n} | p_i - q_i |}{n} \quad (9)$$

The lower the MAE, the more accurate the predictions would be, allowing for better recommendations to be formulated. MAE has been computed for different prediction algorithms and for different levels of sparsity.

C. Sensitivity of different training-test ratio x

To determine the sensitivity of density of the dataset we carried out an experiment where we varied the value of x from 0.2 to 0.8 in an increment of 0.1. For each of these training-test ratio values we ran our experiments using our proposed algorithm and the traditional CF algorithm. The results are shown in Figure 3. We observe that the quality of prediction increase as we increase x and our proposed CF is better than the traditional.

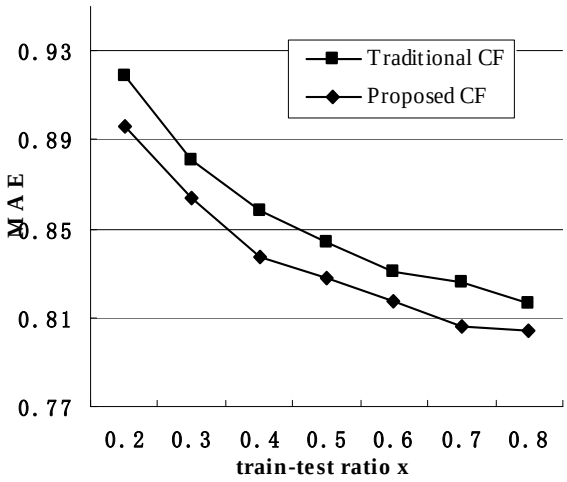

Figure3. MAE of the different prediction algorithm with respect to train-test ratio x.

D. Comparing with the traditional CF

We compare the proposed method combining user clustering and item clustering collaborative filtering with the traditional collaborative filtering. The size of the neighborhood has a significant effect on the prediction quality. In our experiments, we vary the number of neighbors and compute the MAE. The obvious conclusion from Figure 4, which includes the Mean Absolute Errors for the proposed algorithm and the traditional collaborative filtering as observed in relation to the different numbers of neighbors, is that our proposed algorithm is better.

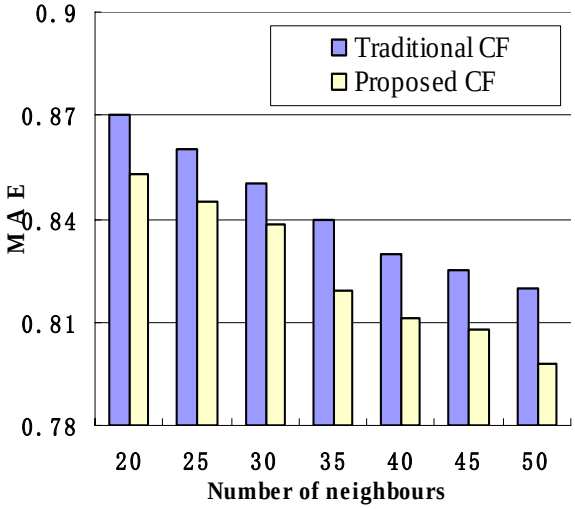

Figure4. Comparing the proposed CF algorithm with the traditional CF algorithm.

VII. CONCLUSIONS

Recommender systems can help people to find interesting things and they are widely used in our life with the development of electronic commerce. Many recommendation systems employ the collaborative filtering technology, which has been proved to be one of the most successful techniques in recommender systems in recent years. With the gradual increase of customers and products in electronic commerce systems, the time consuming nearest neighbor collaborative filtering search of the target customer in the total customer space resulted in the failure of ensuring the real time requirement of recommender system. At the same time, it suffers from its poor quality when the number of the records in the user database increases. Sparsity of source data set is the major reason causing the poor quality. To solve the problems of scalability and sparsity in the collaborative filtering, this paper proposed a personalized recommendation approach joins the user clustering technology and item clustering technology. Users are clustered based on users' ratings on items, and each users cluster has a cluster center. Based on the similarity between target user and cluster centers, the nearest neighbors of target user can be found and smooth the prediction where necessary. Then, the proposed approach utilizes the item clustering collaborative filtering to produce the recommendations. The recommendation joining user clustering and item clustering collaborative filtering is more scalable and more accurate than the traditional one.

ACKNOWLEDGMENT

A Project Supported by Scientific Research Fund of Zhejiang Provincial Education Department (Grant No. Y200806038).

REFERENCES

[1] Breese J, Hecherman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98). 1998. 43~52.

[2] Chong-Ben Huang, Song-Jie Gong, Employing rough set theory to alleviate the sparsity issue in recommender system, In: Proceeding of the Seventh International Conference on Machine Learning and Cybernetics (ICMLC2008), IEEE Press, 2008, pp.1610-1614.

[3] Sarwar B, Karypis G, Konstan J, Riedl J. Item-Based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International World Wide Web Conference. 2001. 285-295.

[4] Manos Papagelis, Dimitris Plexousakis, Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents, Engineering Application of Artificial Intelligence 18 (2005) 781-789.

[5] Hyung Jun Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Sciences 178 (2008) 37-51.

[6] SongJie Gong, The Collaborative Filtering Recommendation Based on Similar-Priority and Fuzzy Clustering, In: Proceeding of 2008 Workshop on Power Electronics and Intelligent Transportation System (PEITS2008), IEEE Computer Society Press, 2008, pp. 248-251.

[7] SongJie Gong, GuangHua Cheng, Mining User Interest Change for Improving Collaborative Filtering, In:Second International Symposium on Intelligent Information Technology Application(IITA2008), IEEE Computer Society Press, 2008, Volume3, pp.24-27.

[8] Duen-Ren Liu, Ya-Yueh Shih, Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences, The Journal of Systems and Software 77 (2005) 181–191.

[9] Yu Li, Liu Lu, Li Xuefeng, A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce, Expert Systems with Applications 28 (2005) 67–77.

[10] George Lekakos, George M. Giaglis, Improving the prediction accuracy of recommendation algorithms: Approaches anchored on human factors, Interacting with Computers 18 (2006) 410–431.

[11] L. H. Ungar and D. P. Foster. Clustering Methods for Collaborative Filtering. In Proc. Workshop on Recommendation Systems at the 15th National Conf. on Artificial Intelligence. Menlo Park, CA: AAAI Press.1998

[12] L. H. Ungar and D. P. Foster. A Formal Statistical Approach to Collaborative Filtering. Proceedings of Conference on Automated Leading and Discovery (CONALD), 1998.

[13] M. O. Conner and J. Herlocker. Clustering Items for Collaborative Filtering. In Proceedings of the ACM SIGIR Workshop on Recommender Systems, Berkeley, CA, August 1999.

[14] A. Kohrs and B. Merialdo. Clustering for Collaborative Filtering Applications. In Proceedings of CIMCA'99. IOS Press, 1999.

[15] Lee, WS. Online clustering for collaborative filtering. School of Computing Technical Report TRA8/00. 2000.

[16] K Honda, N Sugiura, H Ichihashi, S Araki. Collaborative Filtering Using Principal Component Analysis and Fuzzy Clustering,Lecture Notes in Computer Science, 2001

[17] S.H.S. Chee，J Han，K. Wang．Rectree: An efficient collaborative filtering method. Lecture Notes in Computer Science, 2114, 2001

[18] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, Recommender systems for large-scale e-commerce: Scalableneighborhood formation using clustering, Proceedings of the Fifth International Conference on Computer andInformation Technology, 2002

[19] D. Bridge and J. Kelleher, Experiments in sparsity reduction: Using clustering in collaborative recommenders, in Procs. of the Thirteenth Irish Conference on Artificial Intelligence and Cognitive Science, pp. 144–149. Springer, 2002.

[20] J. Kelleher and D. Bridge. Rectree centroid: An accurate, scalable collaborative recommender. In Procs. of the Fourteenth Irish Conference on Artificial Intelligence and Cognitive Science, pages 89–94, 2003.

[21] Xue, G., Lin, C., & Yang, Q., et al. Scalable collaborative filtering using cluster-based smoothing. In Proceedings of the ACM SIGIR Conference 2005 pp.114–121.

[22] George, T., & Merugu, S. A scalable collaborative filtering framework based on co-clustering. In Proceedings of the IEEE ICDM Conference. 2005

[23] Rashid, A.M.; Lam, S.K.; Karypis, G.; Riedl, J.; ClustKNN: A Highly Scalable Hybrid Model- & Memory-Based CF Algorithm. WEBKDD 2006.

[24] Cantador, I., Castells, P. Multilayered Semantic Social Networks Modelling by Ontologybased User Profiles Clustering: Application to Collaborative Filtering. EKAW 2006, pp. 334-349.

[25] Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos Papadopoulos, Yannis Manolopoulos,Nearest-Biclusters Collaborative Filtering,WEBKDD 2006

[26] Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos N. Papadopoulos, Yannis Manolopoulos. Nearest-biclusters collaborative filtering based on constant and coherent values. Inf Retrieval 2007 DOI 10.1007/s10791-007-9038-4.

[27] Gao Fengrong, Xing Chunxiao, Du Xiaoyong, Wang Shan, Personalized Service System Based on Hybrid Filtering for Digital Library, Tsinghua Science and Technology, Volume 12, Number 1, February 2007,1-8.

[28] Huang qin-hua, Ouyang wei-min, Fuzzy collaborative filtering with multiple agents, Journal of Shanghai University (English Edition), 2007,11(3):290-295.

[29] Songjie Gong, Chongben Huang, Employing Fuzzy Clustering to Alleviate the Sparsity Issue in Collaborative Filtering Recommendation Algorithms, In: Proceeding of 2008 International Pre-Olympic Congress on Computer Science, World Academic Press, 2008, pp.449-454.

SongJie Gong was born in Cixi, Zhejiang Province, P.R.China, in July 1, 1979. He received B. Sc degree from Tongji University and M. Sc degree in computer application from Shanghai Jiaotong University, P.R. China in 2003 and 2006 respectively. He is currently a teacher in Zhejiang Business technology Institute, Ningbo, P.R.China.

His research interest includes data mining, information processing and intelligent computing. He has published more than 30 papers in journals and conferences.