

Employing User Attribute and Item Attribute to Enhance the Collaborative Filtering Recommendation

SongJie Gong

Zhejiang Business Technology Institute, Ningbo 315012, China

Email: songjie_gong@sina.com

Abstract—Recommender systems are web based systems that aim at predicting a customer's interest on available products and services by relying on previously rated products and dealing with the problem of information and product overload. Collaborative filtering is the most popular recommendation technique nowadays and it mainly employs the user item rating data set. Traditional collaborative filtering approaches compute a similarity value between the target user and each other user by computing the relativity of their ratings, which is the set of ratings given on the same items. Based on the ratings of the most similar users, commonly referred to as neighbors, the algorithms compute recommendations for the target user. They only consider the ratings information. User attribute information associated with a user's personality and item attribute information associated with an item's inside are rarely considered in the collaborative filtering recommendation process. In this paper, a new collaborative filtering personalized recommendation algorithm is proposed which employs the user attribute information and the item attribute information. This approach combines the user rating similarity and the user attribute similarity in the user based collaborative filtering process to fill the vacant ratings where necessary, and then it combines the item rating similarity and the item attribute similarity in the item based collaborative filtering process to produce recommendations. The hybrid collaborative filtering employs the user attribute and item attribute can alleviate the sparsity issue in the recommender systems.

Index Terms—personalized services, collaborative filtering, rating similarity, user attribute similarity, item attribute similarity, sparsity, mean absolute error

I. INTRODUCTION

Everyday there are amount of information produced on the Internet, coupled with the diversity of user information needs, the problem of information overload is becoming increasing serious and we all have experienced the feeling of being overwhelmed. Many researchers and practitioners pay more attention on building a proper tool which can help users obtain resources and services which wanted. Personalized recommendation systems are used to help users obtain recommendations for unseen items based on their preferences, which are able to distinguish one user from another to provide information [1, 2]. The famous electronic commerce website Amazon and CD-Now have

employed recommendation technique to recommend products to customers and it has improved quality and efficiency of their services.

The most techniques used in today's recommendation systems fall into two distinct categories: content-based methods and collaborative filtering methods [3, 4]. And collaborative filtering has been known to be the most successful recommendation techniques. Collaborative methods recommend items based on aggregated user ratings of those items and these techniques do not depend on the availability of textual descriptions. They share the common goal of assisting in the user's search for items of interest, and thus attempt to address one of the key research problems of the information age: locating needles in a haystack that is growing exponentially. Collaborative filtering systems can deal with large numbers of people and with many different items. However there is a problem that the set of ratings is sparse, such that any two users will most likely have only a few co-rated items. The high dimensional sparsity of the user-item rating matrix and the problem of scalability result in low quality recommendations.

Personalized recommendation methods operate upon user ratings on observed items or item features making predictions concerning users' interest on unobserved items [5,6]. In most cases particularly in real-world applications, the number of ratings obtained from users is usually very small compared to the number of ratings that must be predicted. And this problem is called the Sparsity which significantly affects recommendation methods reducing the accuracy of prediction. The sparsity of ratings problem is particularly important in domains with a large number of items as well as a large number of users. Different solutions are required and different prediction techniques must be employed to solve the problem.

Traditional collaborative filtering approaches compute a similarity value between the target user and each other user by computing the relativity of their ratings, which is the set of ratings given on the same items. Based on the ratings of the most similar users, commonly referred to as neighbors, the algorithms compute recommendations for the target user. They only consider the ratings information. User attribute information associated with a user's personality and item attribute information associated with an item's inside are rarely considered in

the collaborative filtering recommendation process. In this paper, a new collaborative filtering personalized recommendation algorithm is proposed which employs the user attribute information and the item attribute information. This approach combines the user rating similarity and the user attribute similarity in the user based collaborative filtering process to fill the vacant ratings where necessary, and then it combines the item rating similarity and the item attribute similarity in the item based collaborative filtering process to produce recommendations. The hybrid collaborative filtering employs the user attribute and item attribute can alleviate the sparsity issue in the recommender systems.

II. RELATED WORKS

Goldberg first proposed the collaborative filtering to publish an account of using CF techniques in information filtering and built a system called Tapestry for filtering emails [7]. But the system was not automated, and required users to construct complex queries in a special query language designed for the task.

GroupLens [8] first introduced an automated collaborative filtering system using a neighborhood-based algorithm. By calculating the similarity using Pearson coefficient between the active user and others, the system selected a set of appropriate neighbors. Then it computed each neighbor's weight to generate prediction for active user. This was the most successful and widely used technology in CF research domain.

Sarwar proposed a new CF algorithm based on similarity of items instead of neighbors [9]. Its accuracy was better than neighbor-based CF. In this approach, the historical information is analyzed to identify relations between the items such that the purchase of an item often leads to the purchase of another item. This approach can quickly recommend a set of items and has been shown to produce recommendation results that in some cases are comparable to traditional.

Schafer present a detailed taxonomy and examples of recommender systems used in E-commerce and how they can provide one-to-one personalization and at the same can capture customer loyalty. Although these systems have been successful in the past, their widespread use has exposed some of their limitations such as the problems of sparsity in the data set, problems associated with high dimensionality and so on.

Bayesian networks construct user models based on the training set. Each node corresponds to each item and the states for each node is the possible rank value for that item. A decision tree represents the conditional probability table at each node. Training of the model could be off-line and it was proved practical for environments in which consumer's preference changes slowly with respect to time needed to build the model. But it was unsuitable for environments in which consumer preference models must be updated rapidly or frequently [10].

The clustering technique clusters the users based on similarities between the users' preference [11,12,13]. Then it recommends items or generates prediction for

active users by the average ratings of other users of the same group. The most popular clustering algorithms are k-neighbors and k-means technique and their improved techniques. The clustering technique can be trained off-line. The personalization of clustering technique recommendation is not so good as other CF. In some cases the clusters are less accurate than the memory-based algorithms. Once the clustering is completed, efficiency of the recommendation may be good due to the smaller amount of the members of the group [14,15].

Classifiers technique trains the training data set using the classifiers. When training finished, classifiers can be used to classify new items. This kind of technique has been quite successful in varied domains from identification of fraud, credit risks in financial transactions to medical diagnosis to intrusion detection. Also it is used in the CF recommender systems [16,17].

III. TRADITIONAL SIMILARITY MEASURES IN THE COLLABORATIVE FILTERING

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 R_{ij} that have been provided by the i th user for the j th item, the table as following [18,19].

TABLE I
USER-ITEM RATINGS TABLE

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

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

B. Similarity Measurement

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 [20, 21].

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 [22,23,24]: 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 $R_{i,c}$ is the rating of the item c by user i , A_i is the average rating of user i for all the co-rated items, and I_{ij} 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 R_{ik} 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.

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

Where $R_{i,c}$ is the rating of the item c by user i , A_i is the average rating of user i for all the co-rated items, and I_{ij} 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.

IV. EMPLOYING USER ATTRIBUTE FOR THE USER BASED COLLABORATIVE FILTERING

Traditional user based collaborative filtering just considers the effect of the rating of neighbors, but do not consider the user attribute information. In this section, we discuss the user attribute and segment the factors of the user attribute information in order to use in our proposed collaborative filtering algorithm.

A. Analyzing the problem

In real world, user has owner demography not relation to the ratings and the information is important to the personalized recommendation system. For example, all users are required to register and to provide demographic information including sex, age, profession, department, specialty, etc. The demographic information of each user can be used to classify users that like similar categories or subjects of items. The specialty information is very useful for generating recommendations. The specialty field refers to the research field of the user. So if two users have the same specialty, they will have the same interest in some items. But, it is not reflecting in the user-item ratings. Therefore, if we know the specialty to which a user belongs, we can partially know which items the user will be interested in. This relationship can be used to initialize the user preferences of a new user [25].

B. User Attribute

We use MovieLens collaborative filtering data set. MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. The historical dataset consists of 100,000 ratings from 943 users on 1682 movies with every user having at least 20 ratings. Ratings follow the 1 to 5 numerical scales.

Except for ratings awarded by users on items, the MovieLens data set includes information regarding specifically the users. The included data consists of a sequential list, with 943 vectors of the following form [3]:

user id | age | gender | occupation | zip code

The user ids are the ones also used in the main data file. The gender can be either ‘M’, for male, or ‘F’, for female. The occupation takes a value from a list of 21 distinct possibilities.

1. Age: The user demand is different according to the user age, and so the user interest is the same. Children would like to watch animation and children’s films, young people would like to watch romance film, the middle age people would like to watch life film, and the old people would like to watch documentary film.

2. Gender: In many aspects, users choose different items as the different genders. The females would like to watch fantasy film, and the males would like to watch war film.

3. Occupation: Many users can divide into one category according to their occupation. The level of the artists is higher than the educations, so they have different interest in the films.

4. Zip code: Users in the same region may have the same interest in same ways.

C. *Combing the rating similarity and user attribute similarity*

We propose a hybrid method that groups users by integrating the user rating similarity and user attribute information similarity. The relative weighting is adopted to adjust the importance of rating similarity and attribute similarity. We initially establish a user-item rating matrix and a user-attribute matrix. Then, users rating similarity and the user attribute similarity are computed. The integrated measurement of similarity is then derived as following formula.

$$sim(i, j) = \omega sim_1(i, j) + (1 - \omega) sim_2(i, j) \quad (4)$$

Where, ω and $1 - \omega$ represent the relative importance of the user rating similarity and user attribute similarity, respectively. If $\omega = 0$, then the method becomes user information-based method. If $\omega = 1$, then the method becomes traditional CF method.

D. *Selecting neighbors*

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

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

(b) The top-n technique is used, which a predefined number of n-best neighbors is selected.

E. *Fill the Vacant Ratings*

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. When count the object user U ratings for not graded items, produce the prediction according to the nearest neighbor for user ratings.

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

$$P_{ut} = A_u + \frac{\sum_{i=1}^c (R_{it} - A_i) * sim(u, i)}{\sum_{i=1}^c sim(u, i)} \quad (5)$$

Where A_u is the average rating of the target user u to the items, R_{it} is the rating of the neighbour user i to the target item t, A_i is the average rating of the neighbour user i to the items, $sim(u, i)$ is the combining similarity of the target user u and the neighbour user i, and c is the number of the neighbours.

V. USING ITEM ATTRIBUTE TO PRODUCE RECOMMENDATIONS

Through the calculating the vacant user's rating by user attribute algorithm, we gained the dense users' ratings. Then, to generate prediction of a user's rating, we

use the item attribute based collaborative filtering algorithms.

A. *The dense user-item matrix*

After we used the user attribute 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 attribute content*

The content of many items such as books, videos, or CDs is difficult to analyze automatically by a computer, but the items may be categorized or clustered based on the attributes of the items. For example, in the context of movies, every movie can be classified according to the "genre" attribute of each item. Other item descriptions such as title, category, subject, authors, and published time also reflect the interests of a user when a user reads or downloads items [26]. Table 2 shows examples of the descriptive information of items.

TABLE II
ITEM-ITEM ATTRIBUTE TABLE

Attribute Item	A1	A2	At
Item1	r11	r12	r1t
Item2	r21	r22	r2t
...
Itemn	m1	m2	mt

Where, r_{ij} denotes the express value of the item to its attribute. The symbol n denotes the total number of items, and t denotes the total number of item attributes.

C. *Data set including item ratings and attributes*

We use MovieLens collaborative filtering data set. MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. The historical dataset consists of 100,000 ratings from 943 users on 1682 movies with every user having at least 20 ratings. Ratings follow the 1 to 5 numerical scales.

The complete data set includes in random order 100,000 vectors of the following form [3]:

user id | item id | rating | time stamp

Obviously, users are enumerated from 1 to 943, items from 1 to 1682, while ratings take values between 1 and 5. The time stamps are unix seconds since 1/1/1970 UTC.

Except for ratings awarded by users on items, the MovieLens data set includes information regarding specifically the items. The items, which in the case of the MovieLens data set correspond to movies, there is another sequential list, with 1682 vectors of the following form:

movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western

The movie ids are the ones used in the main data set. The movie title is a string with the title of the movie. The

release dates are of the form dd-mmm-yyyy, e.g. 14-Jan-1967. The IMDb URL is a web link leading to the Internet Movie Database page of the corresponding movie. The last 19 fields are the film genres. Items can belong to more than one genres at the same time.

D. Measuring the item rating similarity

There are several similarity algorithms that have been used in the item based collaborative filtering: Pearson correlation, cosine vector similarity, adjusted cosine vector similarity, mean-squared difference and Spearman correlation.

In this paper, we will use the Pearson correlation measurement.

Pearson's correlation, as following formula, measures the linear correlation between two vectors of ratings as the target item t and the remaining item r .

$$sim_1(t, r) = \frac{\sum_{i=1}^m (R_{it} - A_t)(R_{ir} - A_r)}{\sqrt{\sum_{i=1}^m (R_{it} - A_t)^2 \sum_{i=1}^m (R_{ir} - A_r)^2}} \quad (6)$$

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 .

E. Measuring the item attribute similarity

We also use the Pearson correlation measurement to compute the item attribute similarity, as following formula.

$$sim_2(t, r) = \frac{\sum_{a=1}^m (R_{ta} - A_t)(R_{ra} - A_r)}{\sqrt{\sum_{a=1}^m (R_{ta} - A_t)^2 \sum_{a=1}^m (R_{ra} - A_r)^2}} \quad (7)$$

Where R_{ta} is the express value of the target item t to its attribute a , R_{ra} is the express value of the remaining item r to the attribute a , A_t is the average value of the target item t for all the co-rated attributes, A_r is the average rating of the remaining item r for all the co-rated attributes, and m is the number of all rating attribute to the item t and item r .

F. Combining the two similarities

We propose a hybrid method that clusters items by combining the item rating similarity and item attribute similarity. The relative weighting is adopted to adjust the importance of rating similarity and attribute similarity. The integrated measurement of similarity is then derived as following formula.

$$sim(i, j) = wsim_1(i, j) + (1-w)sim_2(i, j) \quad (8)$$

Where, w and $1-w$ represent the relative importance of the item rating similarity and item attribute similarity, respectively. If $w=0$, then the method becomes item attribute-based method. If $w=1$, then the method becomes traditional item-based CF method.

G. Selecting neighbors

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

(a) Threshold-based selection, according to which items whose similarity exceeds a certain threshold value are considered as neighbors of the target item.

(b) The top- n technique in which a predefined number of n -best neighbors is selected.

H. 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:

$$P_{ut} = \frac{\sum_{i=1}^c R_{ui} \times sim(t, i)}{\sum_{i=1}^c sim(t, i)} \quad (9)$$

Where R_{ui} 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 For the Collaborative Filtering

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 [28,29,30].

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 $p_1, p_2, p_3, \dots, p_n$ is the prediction of users' ratings, and the corresponding real ratings data set of users is $q_1, q_2, q_3, \dots, q_n$. See the MAE definition as following:

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

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. The combination coefficient of ω in the user attribute and user rating algorithm

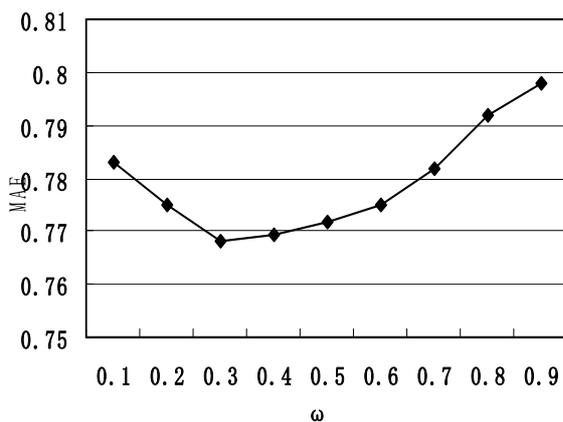


Figure1. MAE of the user attribute based algorithm with respect to the combination coefficient ω .

To determine the sensitivity of the combination coefficient of ω , we carried out an experiment where we varied the value of ω from 0.1 to 0.9 in an increment of 0.1. The generated Mean Absolute Errors according to the value ω are displayed in Figure 1. We observe from the results that the optimal value of the combination

coefficient ω is about 0.3 to 0.4. So in the next experiment we will use in this range.

D. Selecting the optional value of w in the item attribute and item rating algorithm

This experimental step involved trying different values of w , aiming to identify the one that would lead to the best accuracy. The different value of w affect the different weight of the item rating similarity and item attribute similarity. To determine the sensitivity of the combination coefficient of w , we carried out an experiment where we varied the value of w from 0.1 to 0.9 in an increment of 0.1. The generated Mean Absolute Errors according to the value w are displayed in Figure 2. We observe from the results that the optimal value of the optional value w is about 0.5 to 0.6. So in the next experiment we will use in this range.

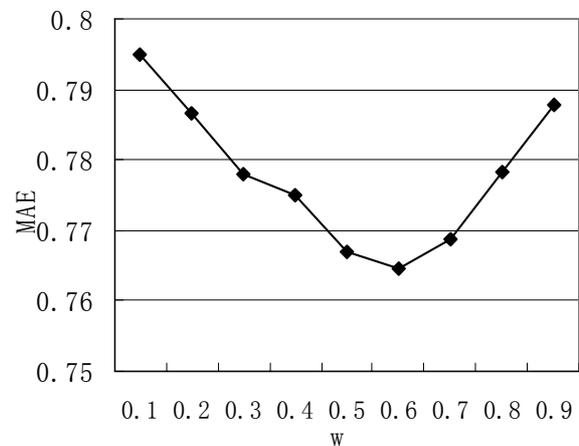


Figure2. MAE of the item attribute based algorithm with respect to the optional value w .

E. Effect of different sparsity

To evaluate the performance of the dataset sparsity used in the proposed algorithm and traditional CF, experiments realize in different ways. Figure 3 illustrates the sensitivity of the algorithms in relation to the different levels of sparsity applied, which compares the performance of two different CF algorithms.

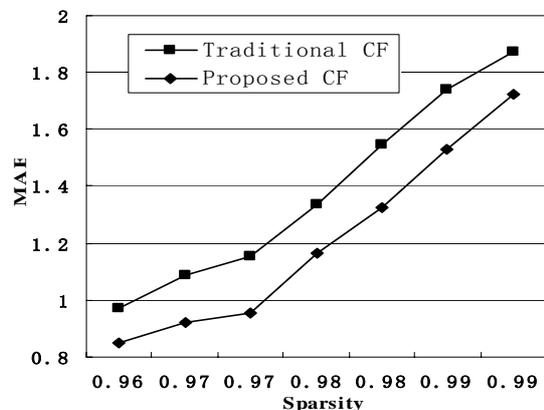


Figure3. MAE of the different prediction algorithm with respect to different sparsity

F. Comparing the proposed collaborative filtering with the traditional CF

We compare the proposed method combining user attribute and item attribute 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.

Because of the high sparsity in rating matrix, the number of co-rated users is very small, even zero in many cases. So it reduces accuracy of similarity coefficient computation. Thus the rating prediction is unreliable. The proposed algorithm computes the vacant value using the user attribute user-based collaborative filtering at first. Thus it has the dense rating matrix, and the lost information was that users do not adopt to compute similarity and often discarded as unrelated information. We can produce the accuracy predictions.

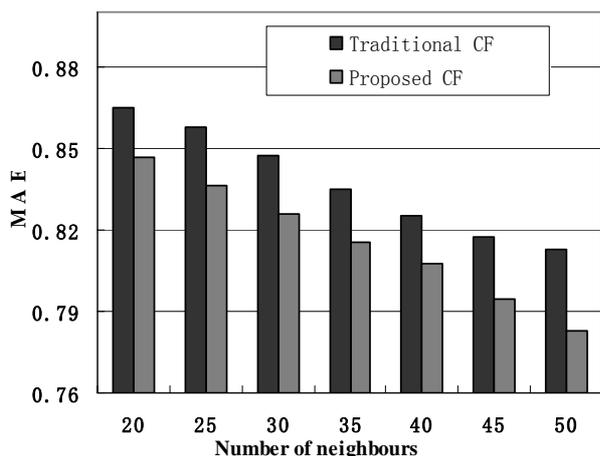


Figure4. Comparing the proposed CF algorithm with the traditional CF algorithm.

VII. CONCLUSIONS

Recommender systems are web based systems that aim at predicting a customer's interest on available products and services by relying on previously rated products and dealing with the problem of information and product overload. Collaborative filtering is the most popular recommendation technique nowadays and it mainly employs the user item rating data set.

Traditional collaborative filtering approaches compute a similarity value between the target user and each other user by computing the relativity of their ratings, which is the set of ratings given on the same items. Based on the ratings of the most similar users, commonly referred to as neighbors, the algorithms compute recommendations for the target user. They only consider the ratings information. User attribute information associated with a

user's personality and item attribute information associated with a item's inside are rarely considered in the collaborative filtering recommendation process. In this paper, we proposed a new collaborative filtering personalized recommendation algorithm which employs the user attribute information and the item attribute information. This approach combines the user rating similarity and the user attribute similarity in the user based collaborative filtering process to fill the vacant ratings where necessary, and then it combines the item rating similarity and the item attribute similarity in the item based collaborative filtering process to produce recommendations. The hybrid collaborative filtering employs the user attribute and item attribute can alleviate the sparsity issue in the recommender systems.

ACKNOWLEDGMENT

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

REFERENCES

- [1] George Lekakos, George M. Giaglis, Improving the prediction accuracy of recommendation algorithms: Approaches anchored on human factors, *Interacting with Computers* 18 (2006) 410–431.
- [2] 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.
- [3] M.G. Vozalis, K.G. Margaritis, Using SVD and demographic data for the enhancement of generalized Collaborative Filtering, *Information Sciences* 177 (2007) 3017–3037.
- [4] 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.
- [5] George Lekakos, George M. Giaglis, A hybrid approach for improving predictive accuracy of collaborative filtering algorithms, *User Model User-Adap Inter* (2007) 17:5–40.
- [6] 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.
- [7] Goldberg D, Nichols D, Oki BM, Terry D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 1992,35(12):61~70.
- [8] Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. Grouplens: An open architecture for collaborative filtering of netnews. In: *Proceedings of the ACM CSCW'94 Conference on Computer-Supported Cooperative Work*. 1994. 175~186.
- [9] 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.
- [10] Chickering D, Hecherman D. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*. 1997,29(2/3):181~212.
- [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] 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.
- [13] A. Kohrs and B. Merialdo. Clustering for Collaborative Filtering Applications. In Proceedings of CIMCA'99. IOS Press, 1999.
- [14] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering, Proceedings of the Fifth International Conference on Computer and Information Technology, 2002
- [15] 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.
- [16] George, T., & Merugu, S. A scalable collaborative filtering framework based on co-clustering. In Proceedings of the IEEE ICDM Conference. 2005
- [17] Rashid, A.M.; Lam, S.K.; Karypis, G.; Riedl, J.; ClustKNN: A Highly Scalable Hybrid Model- & Memory-Based CF Algorithm. WEBKDD 2006.
- [18] Marco Degenmis, Pasquale Lops, Giovanni Semeraro, A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation, User Model User-Adap Inter (2007) 17:217–255.
- [19] 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.
- [20] Hyung Jun Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Sciences 178 (2008) 37-51.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] YaE Dai, SongJie Gong, Personalized Recommendation Algorithm using User Demography Information, WKDD2009, IEEE Computer Society Press.
- [26] SongJie Gong, XiaoYan Shi, A Collaborative Recommender Combining Item Rating Similarity and Item Attribute Similarity, ISBIM2008, IEEE Computer Society Press.
- [27] Huang qin-hua, Ouyang wei-min, Fuzzy collaborative filtering with multiple agents, Journal of Shanghai University (English Edition), 2007,11(3):290-295.
- [28] 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.
- [29] GuangHua Cheng, SongJie Gong, An Efficient Collaborative Filtering Algorithm with Item Hierarchy, In: Second International Symposium on Intelligent Information Technology Application(IITA2008), IEEE Computer Society Press, 2008, Volume3, pp.28-31.
- [30] 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.

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.