Method of Collaborative Filtering Based on Uncertain User Interests Cluster

Xiang Cui, Guisheng Yin
College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China;
Email: cuixiang1220@hotmail.com, yinguisheng@hrbeu.edu.cn

Long Zhang
China Academy of Electronics and Information Technology, CETC, Beijing 100800, China

Yongjin Kang
LS Industrial Systems, Anyang, 431-080, Korea

***Abstract*—Recommender systems have been proven to be valuable means for Web online users to cope with the information overload and have become one of the most powerful and popular tools in electronic commerce. The suggestions provided are aimed at supporting their users in various decision-making processes, such as what items to buy, what music to listen, or what news to read. In the paper, we introduce the uncertain interests of users because computer logs take down the data that have uncertain features. Consequently, user interest data recorded in computer logs is uncertain data. First of all the definition of uncertain interests are put forward. Then, some method using a clustering algorithm can solve the uncertain feature. In the algorithm, we compute the between-class entropy of any two clusters and get the stable classes. Secondly, because calculation input is uncertainty, the results of clustering algorithm are latent non-determinacy. The trustworthy degree of uncertain interests is defined that it can measure the rationality of clustering algorithm results. Thirdly, improvement of collaborative filtering is presented as an advanced CF algorithm based on trustworthy degree of uncertain interests. At last, we provide some evaluation of the algorithms and propose the more improve ideas in the future.**

***Index Terms*—collaborative filtering, recommender systems, clustering, uncertain interests**

I. Introduction

With the development of the Internet, people have entered an age of information explosion from the age of information poor. As web information is speeding up incredibly, it is not easy for people to find the proper information of their interests at the present. More and more people wish for a method to obtain the proper information fastest and most accurately. Recommender Systems (RS) are tools and techniques providing suggestions for items to be of use to a user. The suggestions provided are aimed at supporting their users in various decision-making processes, such as what items to buy, what music to listen, or what news to read. Recommender systems have been proven to be valuable means for online users to cope with the information overload and have become one of the most powerful and popular tools in electronic commerce.

Collaborative filtering is a key technology of recommender systems. At present, problems for traditional collaborative filtering commended systems have some drawbacks. First of all, scalability is the main challenge. The more the number of users or items increases, the higher the computational complexity is. Generalized nearest neighbor collaborative filtering algorithms are capable of searching thousands of latent nearest neighbors for one user [1]. As time goes on, recommended systems have millions of users and items. So any modern recommended system running on those existing algorithms will suffer serious problems due to the enormous growth of database data. As a result, it is a pressing task that some new methods must be found which can work on these huge scale problems and can produce high quality predictions.

In actual application of traditional collaborative filtering recommender systems, collaborative filtering algorithm can improve the quality of recommendations based on user ratings. But rating data is maybe sparsely sometimes. The number of items far exceeds what any individual can hope to absorb, thus matrices containing the ratings for all items for all users are very rare, and actual matrices are sparely filled. Some collaborative filtering systems are being applied to larger sets of items resulting very long item rows for each user [2]. So users cannot give ratings to all these items and hence the density of item ratings decreases. Because of this problem of sparsely, maintaining the quality of predictions will be difficult. Since the density of ratings matrix is very less, most of the entries are blanks and new approaches should suggest some techniques for replacing the blanks with some suitable values. Pure collaborative filtering cannot provide predictions for an item when it first appears since

The project is supported by the National Nature Science Foundation of China (No. 60973075), the Provincial Natural Science Foundation under Grant No.F200937, the Foundation of Harbin Science and Technology Bureau under Grant No. RC2009XK010003, and Basic Scientific Research Foundation of Harbin Engineering University under Grant No. HEUCF1015 and No. HEUCF100605.

Corresponding author: Xiang Cui, cuixiang1220@hotmail.com

Manuscript received July 31, 2011; revised June 19, 2011; accepted June 19, 2011.

doi:10.4304/jcp.8.1.186-193

there are no users' ratings on which to base the predictions [3]. Thus an item cannot be recommended unless a user has rated it before. This problem also comes whenever new items are added to the database.

There have been also plenty of problems about traditional collaborative filtering recommender systems. Typically, uncertain data of user appear in abundance for various reasons, such as user interest data, causing the output of result of recommend inaccurately which is based on computing those input uncertain data. In this paper, we introduce the uncertain interests of users. In General, one computer is used for one user. We can collect the user information such as interests through computer log records. But the log can not sign who used the computer and keep back the user information like interests. Logs take down the data that have uncertain features. Consequently, user interests data recorded in computer logs is uncertain data, as uncertain interests. We can define monitoring points, which make a distinction between the computer logs which take notes multi user uncertain interests.

Based on the definition of uncertain interests, some method using a clustering algorithm can solve the uncertain feature. In the algorithm, we group the user pro les into clusters in order to provide the semantic content information. Then Compute the between-class entropy of any two clusters and get the stable classes. Secondly, because calculation input is uncertainty, the results of clustering algorithm are latent non-determinacy. The trustworthy degree of uncertain interests is defined that it can measure the rationality of clustering algorithm results. Thirdly, improvement of collaborative filtering is presented as an advanced CF algorithm based on trustworthy degree of uncertain interests. At last, we provide some evaluation of the algorithms and propose the more improve ideas in the future.

II. RELATED WORKS

A. Collaborative Filtering Recommender System (CFRS)

Collaborative ltering (CF) is the most successful recommender system technology to date, and is used in many of the most successful recommender systems on the Web. The most popular applications of recommender systems adopt collaborative filtering technology to recommend the rightness items, based on a great many advices of users beforetime, for a special user. These systems employ statistical techniques to find a set of users known as neighbors, which have a history of agreeing with the target user. For example, in an e-commerce scenario, customer either rate different products similarly or they tend to purchase similar set of products [4]. Once a neighborhood of users is formed, these systems use several algorithms to emerge recommendations results to special users. We need give some definitions to describe the collaborative filtering recommender system (CFRS).

Definition 1. *User*

In a typical recommender application such as E-Commerce scenario, there is a list of m users or customers $U = \{u_1, u_2, ..., u_m\}$, which is a set identified users.

Definition 2. *Item*

There is a list of n items such as products chosen by users in e-commerce scenario. It is shown on $I = \{i_1, i_2, ...i_n\}$, that is a set indentified items.

Definition 3. *Rating*

Each user u_i expresses his/her opinions about a list of items which is subset of I. This set of opinions is called the "ratings" of user u_i and is denoted by R_{mn} .The example of r_{mn} is shown in table Ⅰ, that m is user scalar and n is item scalar.

TABLE I.
EXAMPLEYPE OF USER ITEM RATING

	i_1	i_2	…	i_j	…	i_n
u_1	r_{11}	r_{12}	…	r_{1i}	…	r_{1n}
u_2	…	…	…	…	…	…
…	…	…	…	…	…	…
u_i	…	…	…	r_{ij}	…	…
…	…	…	…	…	…	…
u_m	…	…	…	…	..	r_{mn}

There exists a distinguished user $u_a \in U$ called the active user for whom the task of a collaborative ltering algorithm is to nd an item suggestion.

Most collaborative ltering based recommender systems build a neighborhood of likeminded users. The Neighborhood formation scheme usually uses Pearson correlation or cosine similarity as a measure of proximity. The neighborhood formation process is in fact the model-building or learning process for a recommender system algorithm. The main goal of neighborhood formation is to nd, for each user u_a , an ordered list of k users $U = \{u_{1,}u_{2,...,}u_K\}$ such that $u_a \notin U$ and $sim(u_a, u_1)$ is maximum, $sim(u_a, u_2)$ is the next maximum and so on. Where $sim(u_a, u_1)$ indicates similarity between two users, which is most often computed by nding the Pearson-r correlation between the user u_a and u_i .

Once these systems determine the proximity neighborhood, they produce recommendations that can be of two types:

(a). Prediction is a numerical value, $R_{a,j}$, expressing the predicted opinion-score of item i_j for the active user u_a. This predicted value is within the same scale as the opinion values provided by u_a .

(b). Recommendation is a list of N items, $TI_r = \{T_{i1}, T_{i2}, ..., T_{iN}\}$, that the active user will like the most. The recommended list usually consists of the items not already purchased by the active user. This output interface of CF algorithms is also known as *Top-N* recommendation.

The schematic architecture of the collaborative ltering process is shown in Fig. 1. CF algorithms represent the entire $m \times n$ user-item data as a ratings matrix, R .Each entry r_{ij} in R represent the preference score (ratings) of the *i*th user on the *j*th item. Each individual rating is within a numerical scale and it can as well be 0, indicating that the user has not yet rated that item.

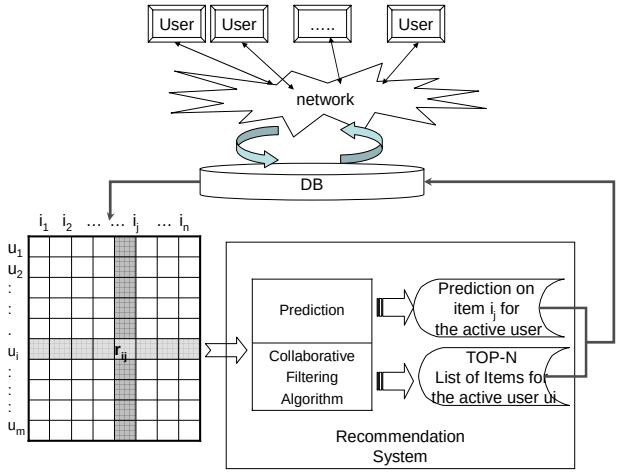

Figure 1. Architecture of the CFRS.

B. Clustering Based On Uncertain Interests

The main problem for scaling a CF classi er is the amount of operations involved in computing distances for nding the best k-nearest neighbors, for instance. A possible solution is to reduce dimensionality [5]. But, even if we reduce dimensionality of features, we might still have many objects to compute the distance to. This is where clustering algorithms can come into play. The same is true for content-based RS, where distances among objects are needed to retrieve similar ones. Clustering is sure to improve efficiency because the number of operations is reduced. However, and unlike dimensionality reduction methods, it is unlikely that it can help improve accuracy. Therefore, clustering must be applied with care when designing a RS, measuring the compromise between improved efficiency and a possible decrease in accuracy.

There are two main categories of clustering algorithms: hierarchical and partitioned. Partitioned clustering algorithms divide data items into non-overlapping clusters such that each data item is in exactly one cluster. Hierarchical clustering algorithms successively cluster items within found clusters, producing a set of nested cluster organized as a hierarchical tree. [6]

Many clustering algorithms try to minimize a function that measures the quality of the clustering. Such a quality function is often referred to as the objective function, so clustering can be viewed as an optimization problem: the ideal clustering algorithm would consider all possible partitions of the data and output the partitioning that minimizes the quality function. But the corresponding optimization problem is NP hard, so many algorithms resort to heuristics (e.g., in the k-means algorithm using only local optimization procedures potentially ending in local minima). The main point is that clustering is a difficult problem for which nding optimal solutions is often not possible [7]. For that same reason, selection of the particular clustering algorithm and its parameters (e.g., similarity measure) depend on many factors, including the characteristics of the data. In the following paragraphs we describe the k-means clustering algorithm and some of its alternatives.

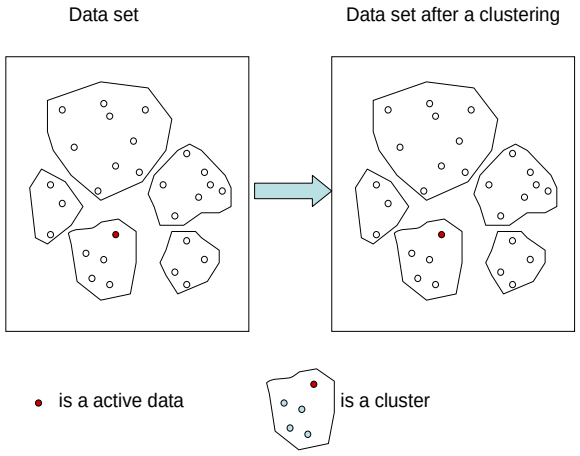

Figure 2. Note how the caption is centered in the column.

In the face of complicated and changeable uncertain prediction, the results are often one sided. Sometimes it is missed the effect of other relational factors in thinking only of the immediate factor, or is concentrated obsessively on the influence of one factor. There are multiple ways to solve the uncertain problem, not only using Bays' theorem to calculate the probability of user interest but also using the information entropy theory to describe the uncertain interests features [8]. In the train sets construction phase, the sample values of uncertain interests for the same monitoring point can vary obviously because the path from last monitoring point to the current monitoring point can vary. Therefore, the expected choice behavior constructed is inaccurate if all samples of the same monitoring point are taken as one training set. We solve the problem by clustering these samples based on multi uncertain interests.

For n samples $X_1, X_2, ..., X_n$, each sample has m uncertain interests $A_1, A_2, ..., A_m$, denoted by sample matrix $X = (x_{ij})_{n \times m} (i = 1, 2...n; j = 1, 2, ...m)$ .These samples are clustered by a clustering algorithm based on entropy.

The clustering process of *n* samples is as follows:

Step1. Count each sample as a class C_k , where $1 \le k \le m$ and C_k constitute N_k sample sets.

Step2. Choose any sample, add it to another class, and compute the entropy of C_k after adding the sample to it according to equation (1) (s is the smoothing constant).

$$H(C_k) = -\log[\frac{1}{(2\pi)^m N_k^2 h^{2m}} \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} A] \qquad (1)$$

A in equation (1) is as follows:

$$A = \exp(-\frac{(x_i - x_j)(x_i - x_j)^T}{2h^2}) \qquad (2)$$

Step3. Assign the sample to the class according to the smallest added value of entropy.

Step4. Update the class's number after steps3.

Step5. Repeat step2 to step 4, until no class is changed.

Step6. Compute the between-class entropy of any two classes according to equation (3) and merge two classes with smallest between-class entropy. The number of classes and samples for each class must be also modified.

$$H(C_i, C_j) = -\log[\frac{1}{(2\pi)^m N_i^2 N_j^2 h^{2m}} \sum_{i=1}^{n} \sum_{j=1}^{n} (M \cdot A)] \quad (3)$$

M is in equation (3) is $M(x_i, x_j)$ defined as :

$$M(x_i, x_j) = \begin{cases} 1, x_i \in C_i, x_j \in C_j \; or \quad x_i \in C_j, x_j \in C_i \\ 0, else \end{cases} \quad (4)$$

Step7. Repeat step6 until there are only two classes. If the changes of the smallest between-class entropy at a certain time are more significant than before, the classes at this time, represented as $C_1, C_2, ..., C_m$ (*m* is the number of classes), is the result of clustering.

III. COLLABORATIVE FILTERING IMPROVEMENT MATHOD

A. Traditional Collaborative Filtering

In this section, we review several major approaches for collaborative filtering. The memory-based approaches [9] are among the most popular prediction techniques in collaborative filtering. The basic idea is to compute the active user's predicted vote of an item as a weighted average of votes by other similar users or K nearest neighbors (KNN). Two commonly used memory-based algorithms are the Pearson Correlation Coefficient (PCC) algorithm [10] and the Vector Space Similarity (VSS) algorithm [11]. These two approaches differ in the computation of similarity.

Two popular model-based algorithms are the clustering for collaborative filtering [12] and the aspect models. Clustering techniques work by identifying groups of users who appear to have similar preferences. Once the clusters are created, predictions for an individual 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 the degree of participation. The aspect model is a probabilistic latent-space model, which considers individual preferences as a convex combination of preference factors. The latent class variable is associated with each observation pair of a user and an item. The aspect model assumes that users and items are independent from each other given the latent class variable.

Pennock et al. proposed a hybrid memory- and model-based approach [13]. Given a user's preferences for some items, they compute the probability that a user belongs to the same "personality diagnosis" by assigning the missing rating as a uniform distribution over all possible ratings. Previous empirical studies have shown that the method is able to outperform several other approaches for collaborative filtering, including the PCC method, the VSS method and the Bayesian network approach. However, the method neither takes the whole aggregated information of the training database into account nor considers the diversity among users when rating the non-rated items. From our point of view, the clustering-based smoothing could provide more representative information for the rating.

Several other related methods have also been proposed to deal with the sparsity problem. The dimension-reduction method aims to reduce the dimensionality of the user-item matrix directly. A simple strategy is to form clusters of users or items and then use these clusters as basic units in making recommendation. Principle Component Analysis (PCA) and information retrieval techniques such as Latent Semantic Indexing (LSI) are also proposed.

The dimensionality-reduction approach addresses the sparsity problem by removing unrepresentative or insignificant users or items so as to condense the user-item matrix. However, potentially useful information might be lost during this reduction process. By considering the association between users and items, transitive associations of the associative-retrieval technique [14] are proposed to iteratively reinforce the similarity of the users and the similarity of items.

Content-boosted CF approaches require additional information regarding items as well as a metric to compute meaningful similarities among them. In [16], A. Popescul et al. also proposed a unified probabilistic model for integrating content information to solve the sparse-data problem. Most previous studies have demonstrated significant improvement in recommendation quality. However, in practice, such item information may be difficult or expensive to acquire. Given an item, similar items rated by the active user in the past are identified and then used for recommendation. Item similarities are computed as the correlations between the corresponding column (item) vectors.

The traditional similarity measuring methods are mainly three classes, based on user rating data can be represented by matrix R . Here, *m* is the number of users,

n is the number of items, and R_{ij} denotes the rating of user u_i on item *j*.

(1). Cosine-based Similarity: In this case, two items are thought of as two vectors in the *m* dimensional user-space. Vector *i* denotes the ratings on item *i* and vector *j* denotes the ratings on item *j*. And similarity between items *i* and *j*, denoted by $sim(i, j)$ is given by

$$\sin(i, j) = \cos(i, j) = \frac{i \cdot j}{\|i\| \|j\|} \quad (5)$$

(2). Correlation-based Similarity: In this case, similarity between two items *i* and *j* is measured by computing the Pearson-r correlation $corr_{ij}$ .To make the correlation computation accurate we must isolate the rated cases. Let the set of users who both rated *i* and *j* be denoted by *U*, then the correlation similarity is $\sin(i, j) = corr_{ij}$, given by

$$corr_{ij} = \frac{\sum_{u \in U} (R_{u,i} - \overline{R_i})(R_{u,j} - \overline{R_j})}{\sqrt{\sum_{u \in U} (R_{u,i} - \overline{R_i})^2} \sqrt{\sum_{u \in U} (R_{u,j} - \overline{R_j})^2}} \quad (6)$$

(3). Adjusted Cosine Similarity: Computing similarity using basic cosine measure in item-based case has one important drawback; the differences in rating scale between different users are not taken into account. The adjusted cosine similarity offsets this drawback by subtracting the corresponding user average from each rated pair. Formally, the similarity between items *i* and *j* using this scheme is given by

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \overline{R_u})(R_{u,j} - \overline{R_u})}{\sqrt{\sum_{u \in U} (R_{u,i} - \overline{R_u})^2} \sqrt{\sum_{u \in U} (R_{u,j} - \overline{R_u})^2}} \quad (7)$$

Based on the above methods to compute similarity, we select one of them to get the most similar collection of items.

B. Collaborative Filtering Based on Uncertain User Interests Cluster

The collaborative filtering recommendation algorithm based on uncertain user interests' model in this paper improves the performance of similarity measuring method by introducing the trustworthy degree d_{ij}^q of uncertain interest A_j when the user rating data are extremely sparse and uncertain. The trustworthy degree d_{ij}^q of uncertain interest A_j can be defined based on some values. Let $y_{1j}^q, y_{2j}^q, ..., y_{n'j}^q$ be n' sample values of the uncertain interest A_j , for the training samples $Y_1^q, Y_2^q, ..., Y_{n'}^q$ of class $C_q (1 \le q \le M)$.

And then, we can determine the trustworthy degree of uncertain interests according to the degree of deviation from the average value, because uncertain interests are approximately normally distributed and the normal value of each uncertain interest changes around the average value after removing various effects of the environment. d_{ij}^q is divided two parts, where $1 \le i \le n', j = 1, 2, ..., m$.

One part is $y_{ij}^q \in [\min_{1 \le i \le n'} \{y_{ij}\}, (\sum_{i=1}^{n'} y_{ij}^q)/n']$, then:

$$d_{ij}^q = \frac{y_{ij}^q}{(\sum_{i=1}^{n'} y_{ij}^q)/n'} \quad (8)$$

Another is $y_{ij}^q \in [(\sum_{i=1}^{n'} y_{ij}^q)/n', \max_{1 \le i \le n'} \{y_{ij}\}]$, then:

$$d_{ij}^q = 1 + \frac{(\sum_{i=1}^{n'} y_{ij}^q)/n'}{\max_{1 \le i \le n'} \{y_{ij}\}} - \frac{y_{ij}^q}{\max_{1 \le i \le n'} \{y_{ij}\}} \quad (9)$$

From this description, it is clear that $d_{ij}^q \in [0,1]$. The trustworthy degree d_i^q of training sample Y_i^q is $d_i^q = \sum_{j=1}^{m} w_j^q d_{ij}^q$, where w_j^q is the weight of uncertain interest A_j for class C_q .

Using the similarity measuring method based on trustworthy degree of uncertain interest, there is no need to isolate the users who have rated same items but values not the same one, and no need to isolate items that have been rated by same monitoring point users. We regard all ratings of an item as an input to establish a trustworthy model for each item, and the similarity between items can be determined by the weight *w*, as trustworthy degree of uncertain interest. That is to say, we do not focus on a rating of an item rated by a specific user, but the statistical the degree characteristics of the overall uncertain interests by clustering technique described above.

After grouping the user pro les, we obtain a new rating matrix. We can then use the classic collaborative algorithms to calculate the similarity between users and make predictions. There are numerous different ways to calculate the similarity between users. Considering the fact which users with similar interests in items may have very different rating patterns: some users tend to assign a higher rating to all items than other users. The adjusted cosine correlation algorithm is defined $sim(k,l)_{user}$.

Define weight value, w= d_{ij}^q . Prediction for an item is then calculated by performing a weighted average of deviations from the uncertain interests class's neighbor.

Here we use the top N rule to select the nearest N neighbors based on the similarities of users. The general formula for a prediction on item i by user k is:

$$p_{k,i} = \overline{R}_k + \frac{\sum_{u=1}^{n}(R_{u,i} - \overline{R}_u) \times sim(k,u)}{\sum_{u=1}^{n} | sim(k,u) |} \quad (10)$$

Where $p_{k,i}$ represents the prediction for user *k* of item *i*; *n* denotes the top *N* nearest neighbors of user *k*; and $\overline{R}_k$ represents the average ratings of user u on the items.

Consider the uncertain interests define above; we can do the partial prediction according to the trustworthy degree d_j^q of uncertain interest A_j for class C_q.

Input: The trustworthy degree d^q of test sample Z for class C_q is: $d^q = \sum_{j=1}^{m} w_j^q d_j^q$.

Output: $p_{k,i}$

(1). Firstly, trustworthy degree d of test sample Z is: $d = \max_{1 \le q \le M} \{d^q\}$.

(2). For given threshold of trustworthy degree τ ,

if $d \in [\tau, 1]$,

then

$$p_{k,i} = \overline{R}_k (1-d) + \frac{\sum_{u=1}^{n}(R_{u,i} - \overline{R}_u) \times sim(k,u)}{\sum_{u=1}^{n} | sim(k,u) |} \times d \quad (11)$$

if $d < \tau$, it is untrustworthy.

IV. Experimental Evaluation

In this section we present a brief discussion of our experimental data set, evaluation metric and experimental platform followed by the experimental results and discussion.

A. Data Sets

Experimental data sets we used are MovieLens, which is a web-based research recommender system that debuted in fall 1997. MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. Each week hundreds of users visit MovieLens to rate and receive recommendations for movies. Actually, over 50, 000 users have expressed opinions on lots of deferent movies. The data set contains 943 independent users information. These users have given marks to 1682 movies, and voted 100, 000 ratings for the movies from the database. Particularly we only consider users that had rated 20 or more movies. We divided the database into 70% training set and 30% test set. The data set was converted into a user-movie matrix R that had 943 rows (users) and 1682 columns (movies that were rated by at least one of the users).

Moreover, a new applicable clustering rating matrix is obtained through using information entropy to preprocess the user rating matrix, as the scarcity of the matrix obtained by original data sets. Table Ⅱ is some of the rating data parameters.

TABLE II.
MovieLens Rating Data Set Sparameters

Ratings average	1	2	3	4	5
3．58	56161	107534	261156	348883	226266

All ratings are contained in the file "ratings.dat" and are have four attributes. Userids range between 1 and 6040, movieids range between 0 and 3592, ratings are made on a 5-star scale, timestamp is represented in seconds since the epoch as returned by time, and each user has at least 20 ratings.

B. MAE Evaluation Metrics

Recommender systems researchers use a number of different measures for evaluating the success of the recommendation or prediction algorithms. For our experiments, we use a widely popular statistical accuracy metric named *Mean Absolute Error (MAE)*, which is a measure of the deviation of recommendations from their true user specified value. For each ratings prediction pair $< p_i, q_i >$, this metric treats the absolute error between them i.e., $|p_i - q_i|$ equally. The MAE is computed by rst summing these absolute errors of the *N* corresponding ratings-prediction pairs and then computing the average. Formally, $MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N}$, The lower the MAE, the more accurately results.

We implement uncertain interests cluster methods and carry out our experiments with different numbers of clusters. Figure 3 shows the experimental results. It can be observed that the number of clusters affects the quality of prediction.

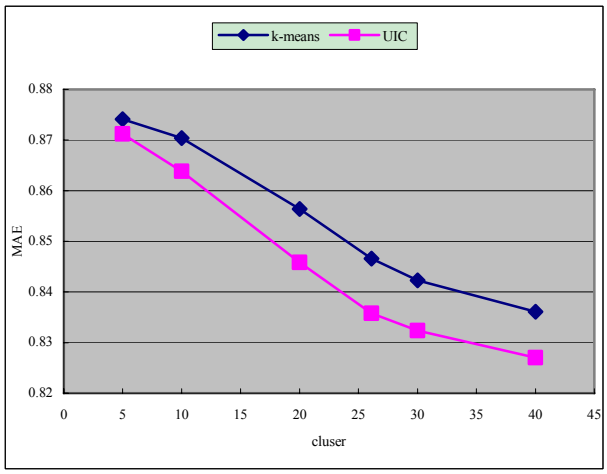

Figure 3. Compare Clustering Experimental Results.

As discussed previously, the UIC clustering algorithm appears to represent better than the k-means algorithm. Since the computation complexity of the UIC algorithm is heavier than the k-means algorithm, we use more time to computer the values. We have done experiments of five group's films sets, for instance values 25, 50, 100, 500, 1000 films. The experimental results at traditional CFRS are compared with the UIC-CFRS in this paper. The figure 4 shows the algorithm results. The results of experiments indicated that the UIC-CFRS can obtain more accurate recommenders.

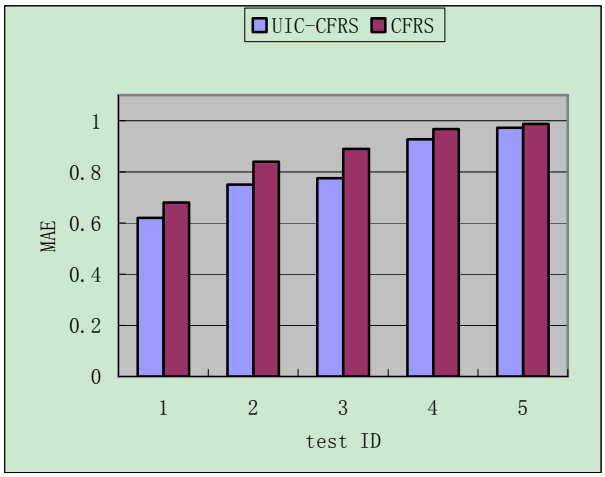

Figure 4. Compare CFRS and UIC-CFRS Results.

V. Conclusion and Future Work

The method of collaborative filtering used in recommender systems is the powerful new technology for extracting additional value for a proper user through computing related data about similar users or items. The problems about uncertain data especially user interests must be solved. In this paper, we presented and experimentally evaluated a new approach in improving the scalability of collaborative filtering recommender systems by using clustering techniques. Our experiments suggest that clustering based neighborhood provides comparable prediction quality as the basic CF approach and at the same time improves the online performance signi cantly. In this paper, we introduce the uncertain interests of users. Based on the definition of uncertain interests, some method using a clustering algorithm can solve the uncertain feature. The trustworthy degree is the weight factor which is used to measure the rationality of clustering results. And then, improvement method of collaborative filtering recommender system is presented based on the weight factor. Simulation experiments are provided for evaluating the algorithms and experimental results have proved the validity of the method through comparison of the traditional method. In the future, recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. More applications about recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web. New technologies are needed that can dramatically improve not only the scalability of recommender systems but also the feature of uncertain data of users or items.

Acknowledgment

This research is supported by the National Nature Science Foundation of China (No. 60973075)

This work is sponsored by the National Natural Science Foundation of China under Grant No. 60973075, the Provincial Natural Science Foundation under Grant No.F200937, the Foundation of Harbin Science and Technology Bureau under Grant No. RC2009XK010003, and Basic Scientific Research Foundation of Harbin Engineering University under Grant NO. HEUCF1015 and No. HEUCF100605.

References

[1] Nathan N. Liu, Min Zhao, and Qiang Yang, "Probabilistic latent preference analysis for collaborative filtering," *Proc. the 18th ACM Conference on Information and Knowledge Management (CIKM 09), ACM Press*, Nov. 2009, pp. 759-766.

[2] Zhao ZD and Shang MS, "User-based collaborative-filtering recommendation algorithms on hadoop," *Proc. the 3rd International Conference on Knowledge Discovery and Data Mining (WKDD 10), IEEE Computer Society, Jan.* 2010, pp. 478-481.

[3] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms", *ACM Transactions on Information Systems*, Vol. 22, Jan. 2004, pp. 143–177.

[4] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. "Item-based collaborative ltering recommendation algorithms", *Proc. the 10th International World Wide Web Conference (WWW 01), ACM Press,* May 2001, pp. 285–295.

[5] H. Ma, I. King, and M. Ly. "prediction for collaborative ltering", *Proc. the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 07), ACM Press*, July 2007, pp. 39–46.

[6] J. Wang, A. De. "based and item-based collaborative ltering approaches by similarity fusion," *Proc. the 29th Annual International ACM SIGIR Conference on Research and Developmen Information Retrieval (SIGIR 06), ACM Press*, Aug 2006, pp. 501–508.

[7] J. F. Tian, and Y. Zhu. "Trusted Software Construction Model Based on Trust Shell". *Advanced Materials Research* 186 , 2011.

[8] H. C. Wang, and H. Peng. "A Clustering Algorithm Based on Entropy". *Computer Science* 34, 11, 2007. (in Chinese)

[9] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," *Proc. the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW 94), ACM Press*, Oct. 1994, pp. 175-186.

[10] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating 'word of mouth'," *Proc. the ACM CHI 95 Human Factors in Computing Systems Conference, ACM Press*, May 1995, pp. 210-217.

[11] H.J. Ahn, "A new similarity ltering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, Jan. 2008, pp. 37–51 .

[12] C. Zeng, C.-X. Xing, L.-Z. Zhou, and X-H. Zheng, "Similarity measure and instance selection for

collaborative ltering" , *International Journal of Electronic Commerce*

[13] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge and Data Engineering* , vol. 19, Jan.

[14] P. Symeonidis, A. Nanopoulos, A.N. Papadopoulos, and Y. Manolopoulos, "Collaborative ltering: fallacies and insights in measuring similarity" , *Proc. the 10th PKDD Workshop on Web Mining (WEBMine)*, Sep. 2, 56–67. ,2006

[15] Yue Shi, Martha Larson, and Alan Hanjalic, "Exploiting user similarity based on rated-item pools for improuser-based collaborative filtering," *Proc. the 3rd ACM conference on Recommender systems (RecSys 09), ACM Press*, Oct. 2009, pp. 125-132.

[16] C.-N. Ziegler, S.M. McNee, J.A. Konstan, and G , "Improving recommendation lists through topic diversi cation," *Proc. the 14th International World Wide Web Conference (WWW 05), ACM Press*, May 2005, pp. 22–32.

[17] Asela Gunawardana and Christopher Meek, "A unified approach to building hybrid recommender systems," *Proc. the 3rd ACM conference on Recommender systems (RecSys)*

[18] Ghazanfar MA and Prugel-Bennett A, "A scalable, accurate hybrid recommender system," *Proc. 3th International Conference on Knowledge Discovery and Data Mining (WKDD 10), IEEE Computer Society*, Jan. 2010, pp. 94-98.

Xiang Cui was born in Harbin, Heilongjiang, China, in 1979. She is a Ph. D candidate of School of Computer Science and Technology, Harbin Engineering University, China. Her research interests include database technique, data mining, network application and software engineering. She is a lecturer in Software School in Harbin University of Science & Technology.

Guisheng Yin was born in Tai Zhou, Jiang Su, China, in 1961. He received the PhD degree in automatic control from Harbin Engineering University, where he is a Full Professor and Doctoral Advisor, the Acting Dean of School of Computer Science and Technology and the Dean of School of Software Engineering. He ever worked in Tokyo University before he joined the current university. His research interests include database and virtual reality.