# Solving the Sparsity Problem in Recommender Systems Using Association Retrieval

YiBo Chen
Computer school of Wuhan University, Wuhan, Hubei, China
chenyibo8224@yahoo.com.cn

ChanLe Wu, Ming Xie and Xiaojun Guo
Computer school of Wuhan University, Wuhan, Hubei, China
National Engineering Research Center for Multimedia Software, Wuhan, China
Email:{chanle.wu, Guoxiaojun}@gmail.com

*Abstract*—**Recommender systems are being widely applied in many fields, such as e-commerce etc, to provide products, services and information to potential customers. Collaborative filtering as the most successful approach, which recommends contents to the current customers mainly is based on the past transactions and feedback of the similar customer. However, it is difficult to distinguish the similar interests between customers because the sparsity problem is caused by the insufficient number of the transactions and feedback data, which confined the usability of the collaborative filtering. This paper proposed the direct similarity and the indirect similarity between users, and computed the similarity matrix through the relative distance between the user's rating; using association retrieval technology to explore the transitive associations based on the user's feedback data, realized a new collaborative filtering approach to alleviate the sparsity problem and improved the quality of the recommendation. In the end, we implemented experiment based on Movielens data set, the experiment results indicated that the proposed approach can effectively alleviate the sparsity problem, have good coverage rate and recommendation quality.**

**Index Terms—collaborative filtering; association retrieval; sparsity problem; recommendation quality**

## I. INTRODUCTION

Along with the rapidly development of the Internet, the number of the servers connected to Internet and the Webs on WWW show a trend of exponential growth. The rapidly development of the Internet present a mass of information to us at the same time, for example, there are tens of thousands movies in Netflix, millions of books in Amazon, more than 10 billion page collection in Del.icio.us, so much information, not to mention find some interesting contents, it is impossible that to gave all of information the once-over. The traditional search algorithm only presents the same ordered results to all of users; can not to provide different service to different users according to their different interests The information explosion reduced the use ratio of the information, this phenomenon is called information overload. Personalized recommendation, included personalized search, has been thought as one of the most effective tools to resolve the problem of information overload. Radically, the recommendation problem is to substitute user to evaluate the strange products, which include books, movies, CD, web and so on, it is a process from know to unknown [1]。.

Recommendation as a social process plays an important role in many applications for consumers, because it is overly expensive for every consumer to learn about all possible alternatives independently. Depending on the specific application setting, a consumer might be a buyer, an information seeker, or an organization searching for certain expertise [2].

Until 1990s, personalized recommendation research as an independent concept be advanced. It rapidly development origin from the web2.0's maturity, which make the user become a participant from browser. In an actually recommender system, there are tens of thousands, or even more than one millions products need to be recommended, for instance, Amazon, eBay, Youtube, etc, also there are huge users. Accurate and high-performance recommender system can mine the potential propensity to consume of the user and provide personalized services for users. In the increasingly fierce competitive environment, personalized recommendation system is not just business marketing means, more importantly, it can improve the user's loyalty and prevent the loss of users.

A recommender system is compose of three parts: action recorder module collect the user's information, model analysis module analyze the user's preference and recommendation algorithm module, thereinto, the recommendation algorithm module is the most core part of the recommendation system [3]. At present, recommendation algorithm mainly includes collaborative filtering algorithm, content-based algorithm, the bipartite

Manuscript received January 1, 2010; revised June 1, 2010; accepted July 1, 2010.
Copyright credit, project number, corresponding author, etc.


doi:10.4304/jcp.6.9.1896-1902

relationship graph recommendation algorithm based on user-product and hybrid recommendation algorithm. This paper focus on the sparsity and precision problem, compute the similarly matrix through the relatively distance between the user's rating and use the association retrieval technology to realize a new collaborative filtering approach.

The remainder of the paper is organized as follows. Section 2 surveys existing work on collaborative filtering and discusses the sparsity problem in detail. Section 3 introduces associative retrieval and summarizes our associative retrieval-based approach to dealing with the sparsity problem and improve the quality of the recommendation. Section 4 presents an experimental study and the experimental data analysis. Section 5 concludes the article by summarizing our research contributions and pointing out future directions.

## II. Collaborative filtering and The sparsity problem

### A. Collaborative filtering

Collaborative filtering aggregates the experiences of similar users in the system to generate personalized recommendations. One key aspect of collaborative filtering is the identification of users similar to the one who needs a recommendation depends on the preference patterns of users makes it more general than other tasks such as ad hoc information retrieval and content-based filtering [4].

Collaborative filtering has been the most successful recommendation system approach to date and has been widely applied in various applications, thereinto, Grundy have been considered the first collaborative filtering system [5]. Grundy system can build user's preference model to recommend relevant books to every users. Tapestry mail processing system, manpower deal with the similarity between users. The more users, the lower precision [6]. GroupLens build the user's information group, within group of users can publish their own information, and with other users make collaborative recommendation [7]. Ringo make use of the same social information filtering method to recommend music to users [8]. There are some other typically collaborative recommendation system, such as Amazon.Com [9], Jester [10], Phoaks [11], and so on.

Many algorithms have been proposed to deal with the collaborative filtering problem. Most collaborative filtering algorithms can be categorized into two classes [12]: Memory-based algorithms and model-based algorithms.

The memory-based algorithms first find the users from the training database that are most similar to the current test user in terms of the rating pattern, and then combine the ratings given by those similar users to obtain the prediction for the test user. The two most commonly methods is Pearson correlation and cosine of the angle. Many enhanced method have been applied into the Pearson correlation and cosine of the angle. For example, absentee voting, case extended, weighted advantage predication, etc. Otherwise, Chen and Cheng make use of the order within product list to compute the similar degree between users; the high-order products have higher weight when computing the user's comparability [13]. But Yang and Gu proposed that using user's behavior information to construct the user's interest point, make use of the interest point to compute the comparability [3][14].

Model-based algorithm collects rating data to study, infer the user's action model, and predicate rating for a product. The difference between model-based collaborative filtering and memory-based collaborative filtering is that model-based approach not based on some of heuristic rule to predicate, but based on data application statistics and machine learning to get model to predicate. Breese et al. proposed two selection probability models: Clustering model and Bayes network [15]. In first model, suppose the user's rating independently, the similarly user cluster into a class, give the user class a mark number. In Bayes network, the number of class and model parameter can obtain from existing data through learning. Other model-based collaborative filtering system have probability correlation model [16], maximum entropy model, linear regression model, and so on.

Despite its success in many application settings, the collaborative filtering approach nevertheless has been reported to have several major limitations including the sparsity, scalability, and synonymy problems. The sparsity problem occurs when transactional or feedback data is sparse and insufficient for identifying neighbors and it is a major issue limiting the quality of recommendations and the applicability of collaborative filtering in general. Our study focused on developing an effective approach to making high-quality recommendations even when sufficient data is unavailable.

### B. The sparsity problem

In collaborative filtering systems, users or consumers are typically represented by the items they have purchased or rated. For instance, in an online cinema have 3 million movies; each consumer is represented by a Boolean feature vector of 3 million elements. The value for each element is determined by whether this consumer has viewed the corresponding movie in the past time. Typically the value of 1 to 5 indicates that such a view had occurred and 0 indicates that no such event has occurred. When multiple consumers are concerned, a matrix composed of all vectors representing these consumers can be used to capture past view events. We call this matrix the consumer–product interaction matrix. In this article, we use C to denote the set of consumers and I to represent the set of items. We represent the consumer–product interaction matrix by a $|C|\times|I|$ matrix $R = (r_{ij})$, such that

$$r_{ij} = \begin{cases} k,\ k = 1 \ldots 5, if\ user\ i\ rated\ item\ j, \\ 0,\ Otherwise \end{cases} \quad (1)$$

In many large-scale applications, both the number of items and the number of consumers are large. In such cases, even when many events have been recorded, the consumer–product interaction matrix can still be extremely sparse, that is, there are very few elements in R whose value is not 0. This problem, commonly referred to as the sparsity problem, has a major negative impact on the effectiveness of a collaborative filtering approach. Because of sparsity, it is highly probable that the similarity (or correlation) between two given users is zero, rendering collaborative filtering useless [17]. Even for pairs of users that are positively correlated, such correlation measures may not be reliable.

The cold-start problem further illustrates the importance of addressing the sparsity problem. The cold-start problem refers to the situation in which a new user or item has just entered the system [18]. Collaborative filtering cannot generate useful recommendations for the new user because of the lack of sufficient previous ratings or purchases. Similarly, when a new item enters the system, it is unlikely that collaborative filtering systems will recommend it to many users because very few users have yet rated or purchased this item. Conceptually, the cold-start problem can be viewed as a special instance of the sparsity problem, where most elements in certain rows or columns of the consumer–product interaction matrix A are 0 [2].

Many researchers have attempted to alleviate the sparsity problem. In [19], the author proposed an item-based approach to addressing both the scalability and sparsity problems. Another proposed approach, dimensionality reduction, aims to reduce the dimensionality of the consumer–product interaction matrix directly. A simple strategy to reduce the dimensionality is to form clusters of items or users and then use these clusters as basic units in the prediction. More advanced techniques can be applied to achieve dimensionality reduction. Examples are statistical techniques such as Principle Component Analysis (PCA) [10] and information retrieval techniques such as Latent Semantic Indexing (LSI). Essentially, dimensionality reduction approaches deal with the sparsity problem by generating a denser user-item interaction matrix that considers only the most relevant users and items. Predictions are then made using this reduced matrix. Empirical studies indicate that dimensionality reduction can improve recommendation quality significantly in some applications, but performs poorly in others, the potentially useful information might be lost during this reduction process [20].

Researchers have also attempted to combine collaborative filtering with content-based recommendation approaches to alleviate the sparsity problem [21][22]. In addition to user-item interactions, such techniques also consider similarities between items derived from their content, which allow them to make more accurate predictions. However, the hybrid approach requires additional information regarding the products and a metric to compute meaningful similarities among them. In practice, such product information may be difficult or expensive to acquire and a related similarity metric may not be readily available.

Another category of methods consider the data as a bipartite graph where nodes represent the users and items, and an edge (i, j) exists between a user i and an item j if i has rated j. Moreover, edge (i, j) is given a weight corresponding to the rating given by i to j. These methods then derive global similarities between users or items using graph theoretic measures. For instance, one such method computes similarities between two users as the average commute time between their respective nodes in a random-walk of the graph. Other graph theoretic measures were also investigated, such as the minimal hop distance between nodes of the graph, and the spread activation of the nodes in the graph. The main drawback of these approaches is that there is often no good interpretation of the similarity measures in the context of the prediction problem [23].

Our research focuses on developing a computational approach to exploring transitive between users to address the sparsity problem and improving the accurate in the context of collaborative filtering.

## III. COLLABORATIVE FILTERING BASED ON ASSOCIATION RETRIEVAL

### *A. Association retrieval*

Associative retrieval has its origin in statistical studies of associations among terms and documents in a text collection. The basic idea behind associative retrieval is to build a graph or network model of documents and index terms and queries, and then to explore the transitive associations among terms and documents using this graph model to improve the quality of information retrieval.

This relationship is also reflected in people's daily life, for instance, Lisi is Wanwu’s friend, Zhanshan is Lisi’s friend, Wanwu can recommend movie A to Zhanshan, so there is an association relationship between Zhanshan and Wanwu. We found that recommender system can make use of this relationship between users to address the sparsity by studying.

### *B. Finding the relationship between users by association retrieval*

Firstly, we supposed that $C = \{c_1, c_2, c_3\}$ represent a user set which includes 3 users, $I = \{i_1, i_2, i_3, i_4\}$ represents a movie set which includes 4 movies, $R = |C| \times |I|$ represent a user’s rating matrix which includes $3 \times 4 = 12$ elements.

$$R = \begin{bmatrix} 0 & 3 & 0 & 4 \\ 0 & 2 & 3 & 5 \\ 4 & 0 & 3 & 0 \end{bmatrix}$$

The rows represent the user, the columns represents the movie, for example, the first row represents the user $c_1$ viewed the movies $i_2$ and $i_4$, the rating is 3 and 4 respectively.

$$B = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

From the second line in the matrix B, we can know that the user $c_2$ viewed the movie $i_2$, $i_3$ and $i_4$. It is easy to find that the user $c_1$ and $c_2$ viewed the movie $i_2$ and $i_4$ from matrix R and B. According to similarity theory, we can ascertain that the user $c_1$ is similarity with the user $c_2$, so the movie $i_3$ can be recommended to the user $c_1$ through the user $c_2$, but the movie $i_1$ cannot be recommended to $c_1$ forever. However, the above example only has 4 movies. At present, the online movie provider more than millions movies, the "dark information" will appear if only through the direct similarity users to recommend, some of movies will cannot be recommended to some of users, the requirements of the user cannot be satisfied.

According to the association retrieval theory, users as a set of nodes, the products as a set of nodes, we use the bipartite graph to express the matrix B, as shown in Fig1.

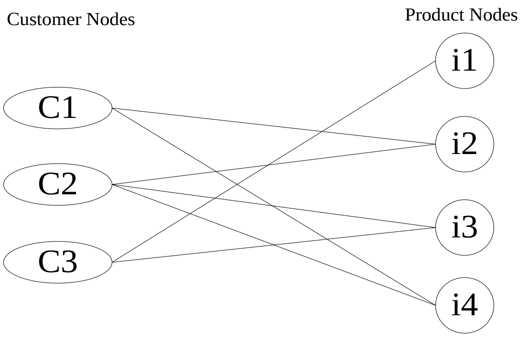


Figure 1. Transitive associations in collaborative filtering.

Accordering to Fig 1, the length of the association path is assumed to be 3, there are $c_1$-$i_2$-$c_2$-$i_3$ and $c_1$-$i_4$-$c_2$-$i_3$ two paths, the movie $i_3$ is recommended to the user $c_1$, but there is not a path whose length is 3 between $i_1$ and $c_1$, so $i_1$ will not be recommended to the user $c_1$. If the length of the path is extended to 5, we can find that the movie $i_1$ can be recommended to the user $c_1$ through the path $c_1$-$i_2$-$c_2$-$i_3$-$c_3$-$i_1$ and $c_1$-$i_4$-$c_2$-$i_3$-$c_3$-$i_1$.

Accorder to the above analysis, this paper makes some of define are as follows:

**Definition 1:** direct recommendation path represent a user recommend item to a target user directly.

**Definition 2:** indirect recommendation path represent a user recommend item to a target user through one or more than one user.

**Definition 3:** user direct similarity degree represents the similarity degree between users in direct recommendation path.

**Definition 4:** user indirect similarity degree represents the similarity degree between recommendation user and target user in indirect recommendation path.

From the above analysis, we know that the association retrieval method can explore the transitive between users to get a set of paths and the direct or indirect similarity degree. Through formula (2) to compute the value of $r_{ij}$ in the sparsity matrix to address the sparsity problem.

$$r_{ij} = \sum_{\hat{c}_n \in \hat{C}} \prod_{c_i, c_j \in \hat{c}_n} a_{ij} \qquad (2)$$

Note that i represents user, j is item, $\hat{C} = (\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_n)$is the set of recommendation path, $\hat{c}_n$ represents an ordered set of a recommendation path the user passed, $a_{ij}$ is similarity degree between $c_i$ and $c_j$.

### C. Computing the direct similarly matrix

In the computing of the direct similarity matrix, we do not use the Pearson-correlation and cosine of the angle. Through the research we find that whatever the user rating is high or low after the user viewed the movie, to some extent, which express some of similarity between users both on the personal preferences and the preference of ratings. For example, in the matrix R, the user $c_1$ and $c_2$ rated $i_2$ and $i_4$, the rating value of the $c_1$ is 3 and 4, the rating value of the $c_2$ is 2 and 5, we can use formula (3) to compute the rating similarity degree between $c_1$ and $c_2$ for the same movie $sim\_2_{(12)} = 0.8$ and $sim\_4_{(12)} = 0.8$,

$$sim_{K(ij)} = \begin{cases} \frac{max(R) - abs(r_{ik} - r_{jk})}{max(R)}, if\ r_{ik}\ and\ r_{jk}\ not\ equal\ 0, \\ 0 \qquad\qquad Otherwise \end{cases} \qquad (3)$$

max is the maximum value function; abs is the absolute value function; R represents the value set of the rating, such as R={0,1,2,3,4,5}; $r_{ik}$, the value of the user i rate product k. Formula (4) was used to compute the user similarity $a_{ij}$ between i and j after get the rating similarity degree.

$$a_{ij} = \begin{cases} \frac{\sum_{k=1}^{m} sim\_K_{(ij)}}{m}, \ if\ i\ not\ equal\ j, \\ 1 \qquad , \ if\ i\ equal\ j \end{cases} \qquad (4)$$

Note that m, the number of the products. We use the rating matrix R as an example, the user similarity $a_{ij} = (0.8 + 0.8) \div 4 = 0.4$, according to this method, we can get the user similarity matrix $A_{sim}$ as follows:

$$A_{sim} = \begin{bmatrix} 1 & 0.4 & 0 \\ 0.4 & 1 & 0.25 \\ 0 & 0.25 & 1 \end{bmatrix}$$

Next, we combine the association retrieval and direct similarity matrix to compute in order to get the recommendation matrix after getting the user similarity matrix.

### D. Computing the recommender matrix

We use the data the section 3.2 provided to recommend for the user $c_1$. When M=3, we can find that $c_1$ has two recommendation path $c_1$-$i_2$-$c_2$-$i_3$ and $c_1$-$i_4$-$c_2$-$i_3$ from the data; the similarity between $c_1$ and $c_2$ is 0.4 from the similarity matrix in section 3.3, the weight of the path is 0.4; so we get the correlation degree of the $i_3$ is $0.4 \times 2 = 0.8$; Because $c_1$ and $c_2$ have the highest similarity, the rating value of the $c_2$ for $i_3$ is 3, so the recommendation value is $0.8 \times 3 = 2.4$. When M=5, there are two recommendation path $c_1$-$i_2$-$c_2$-$i_3$-$c_3$-$i_1$ and $c_1$-$i_4$-$c_2$-$i_3$-$c_3$-$i_1$, the weight is $a_{12} \times a_{23} = 0.4 \times 0.25 = 0.1$, the value of the correlation degree is $0.1 \times 2 = 0.2$, the rating value of the $c_3$ for $i_1$ is 4, so the recommendation value is $0.2 \times 4 = 0.8$.

The recommendation matrix Matrix_R was defined in (5)

$$Matrix\_R^M = \begin{cases} R, & M = 1, \\ B \cdot B^T \cdot A_{sim} \cdot Matrix\_R^{M-2}, & M = 3,5,7 \ldots \end{cases} \qquad (5)$$

Note that R, the rating matrix, $A_{sim}$ is the similarity matrix, B is the marked matrix. Using the data in section 3.2, we get the recommendation matrix $Matrix\_R^3$ and $Matrix\_R^5$ through formula (4) when M=3 and M=5.

$$Matrix\_R^3 = \begin{bmatrix} 0.0000 & 7.6000 & \mathbf{2.4000} & 12.0000 \\ 1.0000 & 8.4000 & 9.7500 & 18.2000 \\ 8.0000 & \mathbf{0.5000} & 6.7500 & \mathbf{1.2500} \end{bmatrix}$$

$$\text{Matrix\_R}^5 = \begin{bmatrix} \mathbf{0.8000} & 21.9200 & 12.6000 & 38.5600 \\ 5.0000 & 31.4050 & 32.8575 & 64.5125 \\ 16.2500 & 3.1000 & 15.9375 & 7.0500 \end{bmatrix}$$

From the above recommendation matrix, we can know that $\text{Matrix}_R{}^3(c_1, i_3) = 2.4000$, $\text{Matrix}_R{}^5(c_1, i_1) = 0.8000$, which consistent with the above computing outcome.

## IV. ALGORITHM

The algorithm is as follows：

**Algorithm1.** Collaborative algorithm based on association retrieval

**Input**：user rating matrix R，the length of path M

**Output**：Recommendation matrix

**Step1.** Matrix B = Matrix R, If $r_{ij}$ not equal 0 then $b_{ij} = 1$ for each $r_{ij}$.

**Step2.** Set the iteration variable N=1.

**Step3.** Original recommendation matrix $\text{Matrix\_R}^N = R$.

**Step4.** Compute the direct similarity matrix $A_{sim}$ according to formula (3) and (4).

**Step5.** Compute the transpose $B^T$.

**Step6.** Compute the matrix $\text{Matrix\_R}^M$ according to formula (5).

**Step7.** If N+2 less than M then N=N+2, goto Step 3 until N larger than M.

## V. EXPERIMENT AND ANALYSIS

### A. Experiment data

The datasets were collected by the GroupLens Research Project at the University of Minnesota.
The data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies; the sparsity degree is 99.937%.

### B. Experiment procedure

For each target consumer, we retrieved the entire set of previously viewed items and sorted them into chronological order by view date. The first 90% of these items was treated as “past” views to serve as input to be fed into different methods to generate recommendations. For comparison purposes, the second 10% of these items were treated as “future” views of the customer and hidden from the recommender system.

In the experiment, we compared the outcome of the Pearson-correlation, Vector similarly, Item-based and our approach. We use precision, recall, coverage and F-measure to measure the effectiveness of a given recommendation approach. These measures are widely accepted in information retrieval and recommender system research [24].

The baseline methods are described below.

- **Pearson Correlation Coefficient (PCC)**

Pearson Correlation Coefficient method predicts the rating of a test user x on item i as:

$$r_{x,i} = \bar{r}_x + k \sum_{\bar{x} \in X} sim(x, \bar{x}) \cdot (r_{\bar{x},i} - \bar{r}_x) \tag{5}$$

Where the coefficient $sim(x, \bar{x})$ is computed as

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{y,s} - \bar{r}_y)^2}} \tag{6}$$

- **Vector Similarity (VS)**

This method is very similar to the previous method except that the correlation coefficient $sim(x, y)$ is computed as:

$$sim(x, y) = cos(x, y) = \frac{\sum_{s \in S_{xy}} r_{x,s} \cdot r_{y,s}}{\sqrt{\sum_{s \in S_{xy}} r_{x,s}^2 \sum_{s \in S_{xy}} r_{y,s}^2}} \tag{7}$$

The definition of the precision, recall, coverage and F-measure are as follows.

$$Precision = \frac{Number\ of\ recommended\ movies\ that\ match\ with\ future\ views}{Total\ number\ of\ recommended\ movies} \tag{8}$$

$$Recall = \frac{Number\ of\ recommended\ movies\ that\ match\ with\ future\ views}{Total\ number\ of\ movies\ in\ future\ views} \tag{9}$$

$$Coverage = \frac{Total\ number\ of\ actually\ recommended\ movies}{Total\ number\ of\ movies\ the\ user\ not\ view} \tag{10}$$

$$F\_measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{11}$$

### C. Experiment results

In our experiment, we called our approach ARC. We respectively compute the precision, recall and F-measure based on Movielens data set for the ARC, PC and COS algorithms. In the ARC, the value of the M is 3. Summarized bar charts are shown in Figs. 2–5. Table1 is the comprehensive comparison about the precision, recall F-measure and coverage between ARC, PC and COS algorithms.

In the aspect of the precision, the ARC increased by 18.40% compared with PC and 33.58% compared with COS. In the aspect of the recall, the ARC increased by 17.65% compared with PC and 66.68% compared with COS. In the aspect of the F-measure, the ARC increased by 18.39% compared with PC and 34.13% compared with COS. In the aspect of coverage, the ARC increased by 4.66% compared with PC and 24.78% compared with COS. From the results, we can see that there are greatly improved in the aspect of the precision, recall, F-measure and coverage. But from the above data, we find that the COS is worst in the situation of the sparsity. Otherwise, in the aspect of coverage, the ARC increased by only 4.66% compared with PC. We also make another experiment, the results show that the coverage can increase more than 10% when the M equals 5, the overhead of the computing have great increased, but the increase was very little in the recommendation precision. This paper considers that a low coverage rate increase for two reasons, on the one hand, it is because the value of the M is 3; on the other hand, maybe the sparse degree of the experiment data set is not enough.

TABLE1.
COMPREHENSIVE COMPARISON TABLE

| | | PC | | | | COS | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F-measure | Coverage | Precision | Recall | F-measure | Coverage |
| ASS | D-value | 0.00256 | 0.1429 | 0.0503 | 0.0378 | 0.00414 | 0.381 | 0.0824 | 0.1685 |
| | The percent of the improving | 18.40% | 17.65% | 18.39% | 4.66% | 33.58% | 66.68% | 34.13% | 24.78% |

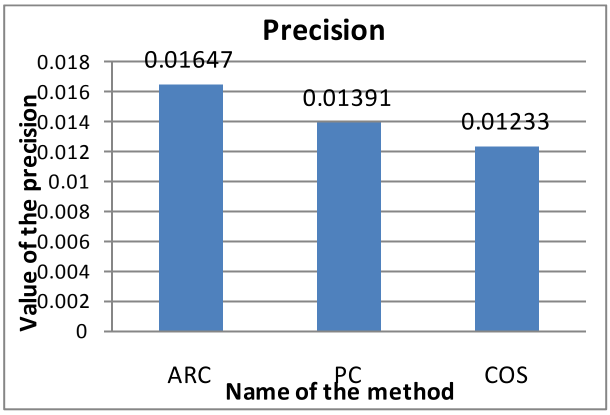


Figure 2. The comparison of the predictive precision

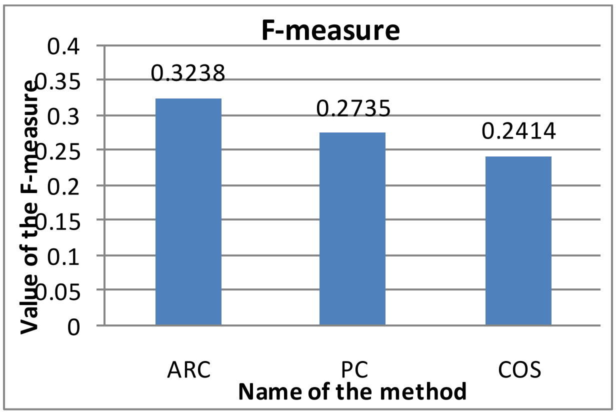


Figure 4. The comparison of the F-measure

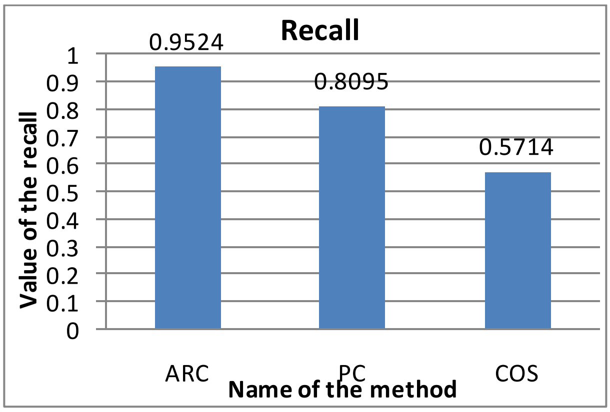


Figure 3. The comparison of the recall

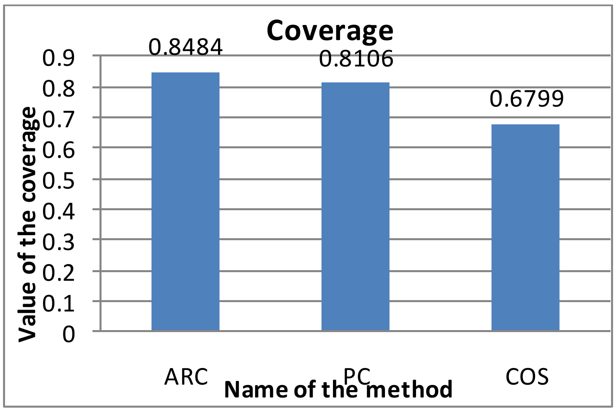


Figure 5. The comparison of the coverage

## VI. CONCLUSION

In this paper, we aimed to alleviate the sparsity problem and improve the recommendation precision in collaborative filtering systems. We use the association retrieval technology to alleviate the sparsity problem and proposed a new collaborative filtering algorithm to increase the recommendation precision. The effectiveness of the approach was evaluated experimentally using data from the movielens data set. The experiment indicated that our approach alleviated the sparsity problem and achieved significantly better recommendation quality than the standard collaborative filtering approaches

Meanwhile, there is a great problem for the proposed approach in this paper. The volume of data these systems utilize will continue increasing over time. In this situation, our approach will cause the data overload problem. As a result, it will present a significant challenge for the scalability of collaborative filtering recommenders. So, the next research, we will consider the scalability problem of collaborative filtering recommenders.

## ACKNOWLEDGMENT

This work was supported in the National Natural Science Foundation of China under Grant No. 60672051. The Fundamental Research Funds for the Central Universities (3105005). Wuhan science and technology plan projects (201010621209)

## REFERENCES

[1] Liu Jianguo, Zhou Tao, et al. Progress of the personalized recommendation systems. *Progress of Nature and Science*, 200919(1):1-15
[2] Zan Huang, Hsinchun Chen, et al. Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in

Collaborative Filtering. *ACM Transactions on Information Systems*, Vol. 22, No. 1, January 2004, 116–142.
[3] Liu Jianguo, Zhou Tao, et al. Overview of the Evaluated Algorithms for the Personal Recommendation Systems. *Complex System and Complexity Science*. 2009, Vol.6, No.3, 1-10.
[4] Rong Jin, Luo Si, et al. Collaborative Filtering with Decoupled Models for Preferences and Ratings. *CIKM '03, New Orleans, Louisiana, USA*, November 3-8, 2003.
[5] Rich E. User modeling via stereotypes. *Cognitive Science*, 1979, 3(4):329-354.
[6] Goldberg D, Nichols D, et al. Using collaborative filtering to weave an information tapestry. *Comm ACM*, 1992, 35(12):61-70.
[7] Konstan JA, Miller BN, et al. GroupLens: Applying collaborative filtering to usenet news. *Comm ACM*, 1997, 40(3):77-87
[8] Shardanand U, Maes P. Social information filtering: Algorithms for automating 'Word of Mouth'. *Proc Conf Human Factors in Computing Systems Denver*, 1995, 210-217.
[9] Linden G, Smith B, et al. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*. 2003, 7(1):76-80.
[10] Goldberg K, Roeder T, et al. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*. 2001, 4(2):133-151.
[11] Terveen L, Hill W, et al. PHOAKS: A system for sharing recommendations. *Comm ACM*, 1997, 40(3):59-62.
[12] J. S. Breese, D. Heckerman, et al. Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proceeding of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI)*. 1998.
[13] Chen YL, Cheng LC. A novel collaborative filtering approach for recommending ranked items. *Expert Systems with Applications*, 2008, 34(4):2396-2405.
[14] Yang MH, Gu ZM. Personalized recommendation based on partial similarity of interests. *Advanced Data Mining and Applications Proceedings*, 2006, 4093:509-516.
[15] Breese JS, Heckerman D, et al. Empirical analysis of predictive algorithms for collaborative filtering. *Proc 14th Conf Uncertainty in Artificial Intelligence Madison*, 1998, 43-52
[16] Getoor L, Sahami M. Using probabilistic relational models for collaborative filtering. *Proc Workshop Web Usage Analysis and User Profiling, San Diego*. 1999.
[17] Billsus, D., Pazzani, M. J. Learning collaborative information filters. *In Proceedings of the 15th International Conference on Machine Learning*, 1998, 46–54.
[18] Schein, A. I., Popescul, A., et al. Methods and metrics for coldstart recommendations. *In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002). (Tampere, Finland)*, 2002, 253–260.
[19] Sarwar, B., Karypis, G., et al. Item-based collaborative filtering recommendation algorithms. *In Proceedings of the 10th International World Wide Web Conference*. 2001, 285–295.
[20] Sarwar, B., Karypis, G., et al. Application of dimensionality reduction in recommender systems: A case study. *In Proceedings of the WebKDD Workshop at the ACM SIGKKD. ACM, New York*.2000.
[21] Good, N., Schafer, J., et al. Combining collaborative filtering with personal agents for better recommendations. *In Proceedings of the 16th National Conference on Artificial Intelligence*, 1999, 439–446.
[22] Huang, Z., Chung, W., et al. A graph-based recommender system for digital library. *In Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (Portland, Ore.). ACM, New York*, 2002, 65–73.
[23] Chrsistian Desrosiers, George Karypis. Solving the Sparsity Problem: Collaborative Filtering via Indirect Similarities. *Technical Report*. Department of Computer Science and Engineering University of Minnesota 4-192 EECS Building 200 Union Street SE Minneapolis, MN 55455-0159 USA. 2008.
[24] Sarwar, B., Karypis, G., et al. Analysis of recommendation algorithms for e-commerce. *In Proceedings of the ACM Conference on Electronic Commerce. ACM, New York*, 2000, 158–167.

**Yibo Chen,** born in 1982, Ph.D. candidate. The research interests include personalization recommendation and semantic web.

**Chanle Wu**, born in 1945, professor, The interests include computer networks, e-Learning, grid computing and semantic web.

**Ming Xie,** born in 1978, Ph.D. candidate. The research interests include data mining and semantic web.
Xiaojun Guo, born in 1984, Ph.D. candidate. The research interests include semantic web and e-Learning.