TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 01/17 pp1-8
Volume 12, Number 1, February 2007

Personalized Service System Based on Hybrid Filtering for Digital Library*

GAO Fengrong (高凤荣), XING Chunxiao (邢春晓)**,
DU Xiaoyong (杜小勇)†, WANG Shan (王 珊)†

Web and Software Technology R&D Center, Research Institute of Information Technology,
Tsinghua University, Beijing 100084, China;
† School of Information, Renmin University of China, Beijing 100872, China

Abstract: Personalized service systems are an effective way to help users obtain recommendations for unseen items, within the enormous volume of information available based on their preferences. The most commonly used personalized service system methods are collaborative filtering, content-based filtering, and hybrid filtering. Unfortunately, each method has its drawbacks. This paper proposes a new method which unified partition-based collaborative filtering and meta-information filtering. In partition-based collaborative filtering the user-item rating matrix can be partitioned into low-dimensional dense matrices using a matrix clustering algorithm. Recommendations are generated based on these low-dimensional matrices. Additionally, the very low ratings problem can be solved using meta-information filtering. The unified method is applied to a digital resource management system. The experimental results show the high efficiency and good performance of the new approach.

Key words: personalized service system; content-based filtering; collaborative filtering; user preferences model; category-based collaborative filtering; meta-information filtering

Introduction

With the rapidly increasing amount of information available, the problem of information overload is becoming increasing acute. We all have experienced the feeling of being overwhelmed by the number of new books, journal articles, and conference proceedings coming out each year. Many researchers pay more attention on building a proper tool which can help users obtain personalized resources. Personalized service systems are one such software tool in which information filtering, collaborative filtering, and data mining techniques are used to help users obtain recommendations for unseen items based on their preferences. For example, a personalized service system on http://www.amazon.com suggests books to each customer based on other books that the customer has bought from Amazon. Another personalized service system on http://www.cdnow.com helps customers choose CDs to purchase as gifts based on previously indicated preferences of the gift received.

A number of web-based personalized service systems have been proposed in diverse fields[1-3]. Two main technologies are usually adopted in personalized service systems: content-based filtering and collaborative filtering. Content-based filtering methods[4-7] provide recommendations by comparing items which interest the user. The items and the user profile are represented in the same feature space, for example, in a

Received: 2005-05-17; revised: 2006-01-13
* Supported by the National Natural Science Foundation of China (No. 60473078)
** To whom correspondence should be addressed.
E-mail: xingcx@mail.tsinghua.edu.cn; Tel: 86-10-62798998

keyword vector space. The similarity of any two items is computed via a standard distance metric (e.g., term frequency/inverse document frequency (TFIDF) similarity)[4]. A major drawback of content-based filtering is that many resources (e.g., multimedia) cannot be represented by a set of keywords selected from the resource content. In contrast, collaborative filtering methods[8-10] offer recommendations to a user based on both the user's previous item ratings and the ratings of same items by other similar users. For a given user, the method recommends items which are highly rated by other users with similar interests. Collaborative filtering systems can deal with large numbers of people and with many different items. A problem, however, is 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. To solve these problems, we present a partition-based recommendation algorithm. A related problem is that if a user does not rate anything, he/she will receive no recommendation from the system. Similarly, an item cannot be recommended if no user has rated it. A meta-information filtering method can solve this problem. The unified method is applied to a practical digital resource management system (DRMS).

1 Domain Description

1.1 Learning item characteristics

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. In a digital library the attribute "category" of items can be used for classification. In general, item description does not follow a uniform standard. So we must first standardize the description of all resources according to a public schema. A program is then used to extract, transport, and load the descriptive data of each item to the DRMS database. Librarians in China rely mainly on Chinese library classification (CLC) for resources and building coincidence relations between the CLC and other classification standards. Meta-information filtering mentioned in this paper is mainly based on CLC. Figure 1 illustrates the multi-level hierarchical categories of the CLC.

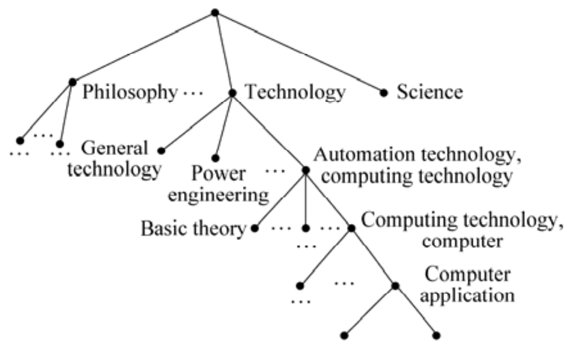

Fig. 1 Multi-level hierarchical categories of CLC

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. Table 1 shows examples of the descriptive information of items.

Table 1 Descriptive information of items

Title	Category	Subject	Authors	Published time
Data Warehouse and OLAP	TP311	Data warehouse; OLAP	Wang Shan	1999
Database System	TP311.13	Database system	Palmer I. R.	1986
Database System Conspectus	TP311.1	Database	Wang Shan and Sa Shixuan	1999
Programming Data Driven Web Applications with ASP.NET	TTP311 TP311.13	Web database	Mack D. and Seven D.	2001
Database Structure Design	TP311.13	Database	Teorey T. J.	1986
Analysis of DBMS	TP311.13	DBMS	Larson J. A.	1986
Web Database Construction Kit	TP311.133	Web database	Khurana G. S. and Khurana B. S.	1997

1.2 Information about items ratings

Without the description of items there is a collection of historical judgments of m users on n items, usually represented as an $m \times n$ user-item matrix, $\boldsymbol{R}$:

$$\boldsymbol{R}_{m\times n}=\begin{bmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{a1} & \cdots & r_{aj} & \cdots & r_{an} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mj} & \cdots & r_{mn} \end{bmatrix}_{m\times n} \quad (1)$$

where r_{aj} denotes the score of item j rated by an active user a. If user a has not rated item j, then $r_{aj}=0$. The symbol m denotes the total number of users, and n denotes the total number of items.

In practice, the amount of items is very large. Users are often reluctant to provide judgments on items. That is to say, the matrix, $\boldsymbol{R}$, is very sparse. In Section 3 we propose an effective method to partition the matrix $\boldsymbol{R}$ based on item categories and a matrix clustering algorithm to produce better recommendations.

1.3 Information about users

The preferences of each user are of primary importance to the quality of a personalized service system. The DRMS system is mainly aimed at researchers in the university. All users are required to register and to provide demographic information including age range (under 22, 23-30, 31-45, 46-60, above 60), title (Professor, Associate Professor, Lecturer, Student), profession (Teacher, Graduate, Undergraduate), department, specialty, research area, etc. The research area information is very useful for generating recommendations. Demographic information can be collected from other intramural information systems or from the homepage of each user. This mode guarantees data authenticity and reduces manual input. The demographic information of each user can be used to classify users that like similar categories or subjects of items. Table 2 shows examples of demographic information for some users.

Table 2 Example of demographic information about the users

Name	Profession	Age range	Title	Department	Specialty	Research area
Joan	Teacher	46-60	Professor	Information school	Computer science	Database, digital library
John	Teacher	31-45	Professor	Information school	Computer science	Information retrieval
Mike	Graduate	23-30	Student	Information school	Computer science	Information retrieval
Jill	Graduate	23-30	Student	Information school	Computer science	Digital library, information system
Chris	Teacher	46-60	Professor	Information school	Library science	Library science
…	…	…	…	…	…	…

The specialty field refers to the research field of the user. The specialty can be closely related to particular categories of books or articles. Therefore, if we know the specialty to which a user belongs, we can partially know which categories the user will be interested in. This relationship can be used to initialize the user preferences of a new user. Another type of useful information is the user rating for each item. When a user reads a book or an article, they can give a score for the item (generally, a score between 1 and 5, from dislike to like).

Figure 2 gives an example of the relation between a user and various categories of items.

The collection of user demographic information is not in conflict with the protection of personal privacy. All users login to the DRMS anonymously. Moreover, the demographic information is only used to produce better recommendations, and is not used for any commercial purpose.

The generation of recommended items is performed by partition-based collaborative filtering based on the user-item matrix. If a new user has not yet rated any items, or no user has rated a new item, the matrix is supplemented to generate recommendations using the relation between the user demographic information and each item.

2 Partition-Based Collaborative Filtering

Both the number of items and the total number of users are very large. In general, each user usually rates a small percentage of the total amount of items, and focuses on a few categories of items (Table 3).

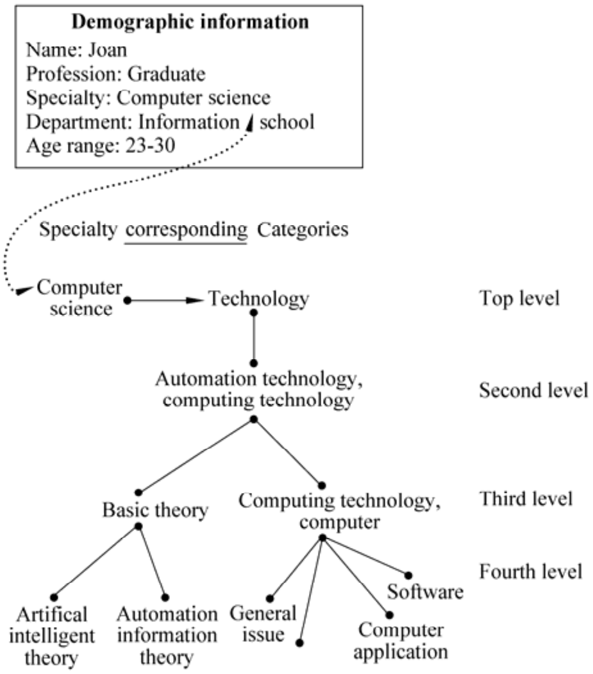

Fig. 2 The relationship of a user and categories

Table 3 User-item ratings

User	Item 1	2	3	4	5	6	7	8
1	5	3	4					
2	4	2		5				
3		3	5	3				
4	4		5	4				
5						3	5	2
6						4	5	
7			2	4			4	5
8			3	5	4	5	4	3
9			5		4			

Using $\boldsymbol{R}_{m\times n}$ in Eq. (1), the prediction of user a to an unseen item j, i.e., p_{aj} is calculated based on the average ratings of user a and a weighted sum of co-rated items between user a and all similar users, i.e.,

$$p_{aj} = \overline{r_a} + k\sum_{i=1}^{m_a} w(a,i)(r_{ij} - \overline{r_i}) \tag{2}$$

where m_a is the number of users who are similar to user a and have rated item j. The weight $w(a,i)$ expresses the similarity between user a and user i. A normalizing factor k is used such that the absolute values of the weights sum to unity. For a target user (in subsequent sections, user a always denotes the target user) the collaborative filtering method provides a list of unseen items in descending order by predicted values.

The high-dimensional sparsity of the original matrix $\boldsymbol{R}_{m\times n}$ provides a motivation to explore methods to partition the matrix into low-dimensional dense matrices. Two partition methods, category-based partition and clustering-based partition, are used in this paper.

2.1 Category-based partition

The category-based partition method can easily classify items and users into several sub-matrices. Each sub-matrix $\boldsymbol{R}^p$ is comprised of the items belonging to category p and the users who rate at least one of the items belong to p. If there are P categories, the original matrix $\boldsymbol{R}$ will be partitioned into P sub-matrices.

$$\boldsymbol{R}^p_{m_p\times n_p} = \begin{bmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1n_p} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{a1} & \cdots & r_{aj} & \cdots & r_{an_p} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{m_p1} & \cdots & r_{m_pj} & \cdots & r_{m_pn_p} \end{bmatrix}_{m_p\times n_p} \tag{3}$$

where n_p is the total number of items, and m_p is the total number of users. Generally, $n_p<n$ and $m_p<m$.

2.2 Clustering-based partition

Although some researchers have proposed that data mining techniques can be applied to collaborative filtering systems[11], there are still only a few published studies describing the integration of these data mining techniques with collaborative filtering. We use the matrix clustering algorithm proposed in Ref. [12] to extract dense sub-matrices from the original matrix.

First, all the ratings in matrix $\boldsymbol{R}$ are translated into (0,1) values. The translation rule is r_{ij}>0 if r_{ij}=1; otherwise, r_{ij}=0. We denote the converted matrix as $\boldsymbol{R}_{m\times n}$. In the following we give two definitions:

Definition 1 For a matrix $\boldsymbol{R}$, its square $S_{\boldsymbol{R}}$ is equal to the number of rows multiplied by the number of columns. That is $S_{\boldsymbol{R}} = m \times n$ where m is the number of users and n is the number of items.

Definition 2 For a matrix $\boldsymbol{R}$, its density $d_{\boldsymbol{R}}$ is defined as the percentage of the number of cells with value equal to 1.

Given a density threshold ϖ, we extract the sub-matrix for which the maximum density is greater than ϖ. Reference [12] gives details of how to perform such sub-matrix extraction.

For convenience, we suppose the original matrix to be partitioned into P sub-matrixes. The value P is in general not equal to that in Section 2.1. Figure 3 gives an example of one sparse matrix divided into 3 low-dimensional sub-matrices.

Original matrix

User	Item 1	2	3	4	5	6	7	8
1	1	1	1	0	0	0	0	0
2	1	1	0	1	0	0	0	0
3	0	1	1	1	0	0	0	0
4	1	0	1	1	0	0	0	0
5	0	0	0	0	0	1	1	1
6	0	0	0	0	0	1	1	
7	0	0	1	1	0	0	1	1
8	0	0	1	1	1	1	1	1
9	0	0	1	0	1	0	0	0

Sub-matrices

User	Item 1	2	3	4
1	1	1	1	0
2	1	1	0	1
3	0	1	1	1
4	1	0	1	1

User	Item 6	7	8
5	1	1	1
6	1	1	0
7	0	1	1
8	1	1	1

User	Item 3	4	5
7	1	1	0
8	1	1	1
9	1	0	1

Fig. 3 Example of an original matrix and its sub-matrices

In the next section we present how to generate recommendations with $\boldsymbol{R}^p_{m_p\times n_p}$ ($p=1,\cdots,P$) for the users.

2.3 Computing similarity

In our algorithm, we use the Pearson correlation coefficient[13,14] to compute the similarity weight between user a and user i $(i=1,\cdots,m_p)$ within the sub-matrix $\boldsymbol{R}^p_{m_p\times n_p}$ ($p=1,\cdots,P$), i.e.,

$$w^p(a,i)=\frac{\sum_j(r_{aj}-\overline{r_{ap}})(r_{ij}-\overline{r_{ip}})}{\sqrt{\sum_j(r_{aj}-\overline{r_{ap}})^2\sum_j(r_{ij}-\overline{r_{ip}})^2}} \tag{4}$$

Here, $j\in I_{ap}\cap I_{ip}$, I_{ap} is the set of items rated by user a within class p, and I_{ip} is the set of items rated by user i within class p. Thus the summations over j are over items for which both user a and user i have given ratings within class p. $\overline{r_{ap}}$ is the average rating of user a for items, which belong to class p, defined as

$$\overline{r_{ap}}=\frac{1}{|I_{ap}|}\sum_{j\in I_{ap}}r_{aj} \tag{5}$$

If the total number of items rated by both user a and user i is below a certain threshold value, it is considered that the two users have no common preferences for class p, i.e., $w^p(a,i)=0$. In detail the similarity is computed as follows.

```
For class p=1 to P
{
  Read R_ap from user preferences model of user a;
  I_ap is the set of items in R_ap;
  If R_ap ≠ 0 and |I_ap| exceeds a threshold
  {
    For each user i=1 to m_p
    {
    Read R_ip from user preferences model of user i;
      I_ip is the set of items in R_ip;
      Compute w^p(a,i) using Eq. (4);
    }
  }
}
```

2.4 Personalized list generation

2.4.1 Predicting unseen items

After the similarity weight is computed between user a and each user in class p, a prediction for the value of unseen items can be computed using the following equation:

$$\mathrm{PR}^p_{aj}=\overline{r_{ap}}+k\sum_{i=1}^{m_{ap}}w^p(a,i)(r_{ij}-\overline{r_{ip}}) \tag{6}$$

where PR^p_{aj} represents the prediction for the target user a for item j within sub-matrix $\boldsymbol{R}^p_{m_p\times n_p}$, and m_{ap} is the number of users similar to user a who have rated item j in class p. Synthetically, we compute the prediction for user a for unseen item j whenever the item belongs to more than one sub-matrix or belongs to just one sub-matrix.

If item j belongs to more than one class, the prediction for user a and item j, i.e., PR_{aj}, is assigned to the maximum value among the classes that item j belongs to

$$\mathrm{PR}_{aj} = \max_{j \in p} \mathrm{PR}_{aj}^{p} \tag{7}$$

If item j belongs to just one class, $p_{aj} = p_{aj}^{p}$.

2.4.2 Recommendation for a list of items

In partition-based collaborative filtering, unseen items for user a are sorted in descending order by their predicted values. The algorithm then provides a list of the highest predicted value items for user a. This is commonly known as top-N recommendation. If user a exists in C sub-matrices, for each sub-matrix p about $\lceil N/C \rceil$ the highest predicted value items are selected for the recommendation.

3 Meta-Information Filtering

Partition-based collaborative filtering is well suited for solving the problem of sparsity. However, there is still an assumption that each user has rated at least one item. When a new user logins the personalized service system, how does one recommend items to the new user? In the context of the digital library, we use the type of item meta-information, not item content, to compute similarity of new items and users. This method is labeled as "meta-information filtering", to distinguish it from content-based filtering. The following describes the two main steps in meta-information filtering.

Step 1 Use the user demographic information to model the user profile as some categories of items.

For example, in Fig. 2 the specialty of user "Joan" is "computer science". The system then considers the items belonging to category "TP31" as being of interest for user "Joan".

Step 2 Compute the weighted value of items based on other users' ratings.

The meta-information filtering method has been shown to be a useful complement for generating recommendations in the DRMS. More details can be found in Ref. [15].

4 Experiments and Evaluation

In this section we experimentally evaluate the performance of partition-based collaborative filtering using either the category-based partition method or the matrix clustering-based method. The performance of our algorithms is also compared against traditional collaborative filtering algorithms. Main parameters of algorithm implementation: Windows XP, VC++ 6.0, CPU AMD Athlon 1.2 GHz, and memory 256 MB.

For convenience, we abbreviate collaborative filtering using category-based method as category-based CF, collaborative filtering using matrix clustering-based method as MCA-based CF, and the collaborative filtering algorithm as traditional CF.

4.1 Data sets

As the DRMS system has not been running for a long time, the ratings at present are not sufficient to test the performance of the various algorithms. Instead, we use a standard data set MovieLens (http://www.grouplens.org/data/) which contains 100 000 ratings of 1682 movies rated by 943 users. Each movie has an attribute "genre". Eighteen genres are included in the data set, and each movie belongs to one or more genres. Each user has rated at least 20 items. The data set is divided into two parts with 80% as a training set and 20% as the test set. We use MovieLens to compare the results of different partition-based collaborative filtering methods.

4.2 Evaluation metrics

The effectiveness of collaborative filtering algorithms has traditionally been measured by statistical accuracy and decision-support accuracy metrics. Statistical accuracy metrics evaluate the accuracy of a system by comparing the numerical recommendation scores against the actual user ratings for the user-item pairs in the test dataset. A popular statistical accuracy metric is the mean absolute error (MAE). The MAE is defined as the average absolute difference between predicted ratings and actual ratings: the lower the mean absolute error, the more accurate the algorithm.

$$\mathrm{MAE} = \frac{\sum_{j=1}^{N} \left| \mathrm{PR}_j - r_j \right|}{N},$$

where $\{\mathrm{PR}_1, \cdots, \mathrm{PR}_N\}$ are predicted values in the target set, and $\{r_1, \cdots, r_N\}$ are all the real values for the same items. N is the total number of items in the target set.

Decision support accuracy metrics evaluate how effective a prediction engine is at helping a user select high-quality items from the set of all items. The receiver operating characteristic (ROC) sensitivity is an example of a decision-support accuracy metric. The metric indicates how effectively the system can steer users towards highly-rated items and away from low-rated ones. We use ROC-4 measure as the evaluation metric.

$$\text{ROC-4} = \sum_{j=1}^{N} w_j \Big/ \sum_{j=1}^{N} u_j ,$$

where

$$w_j = \begin{cases} 1, & r_j \geqslant 4 \text{ and } p_j \geqslant 4, \\ 0, & \text{otherwise}; \end{cases}$$

$$u_j = \begin{cases} 1, & p_j \geqslant 4, \\ 0, & \text{otherwise}, \end{cases} \quad \{\mathrm{PR}_1, \cdots, \mathrm{PR}_N\}$$

are also predicted values in the target set, and $\{r_1, \cdots, r_N\}$ are all the real values for the same items. N is the total number of items in the target set. The larger the ROC-4 value, the more accurate the algorithm.

4.3 Experimental results

4.3.1 Statistical accuracy

When a user rates an item, the number of nearest neighbors affects the MAE for the algorithm. Figure 4 shows the dependence of the MAE on the number of nearest neighbors.

From Fig. 4 we can see that the MCA-based CF algorithm performs better than both traditional collaborative filtering and category-based collaborative filtering algorithms. The category-based CF algorithm performs better than the traditional collaborative filtering algorithm. On average, the MCA-based CF algorithm performs 8.3% better than the traditional CF algorithm, and the category-based CF algorithm performs 5% better than the traditional CF algorithm.

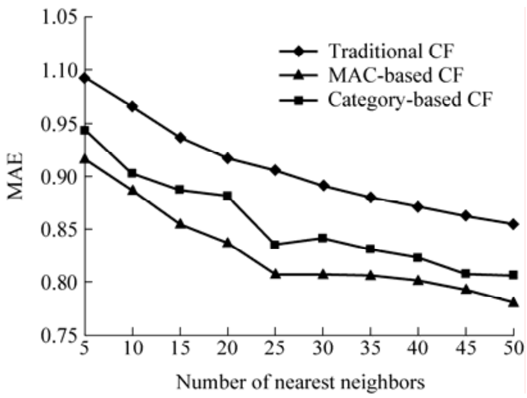

Fig. 4 MAE results

4.3.2 Decision support accuracy

The system works by selecting the top-N items from the predicted item set sorted by descending order in predicted value. The performance of the 3 filters in terms of the ROC-4 measure is shown in Fig. 5, as a function of N, the number of recommended items.

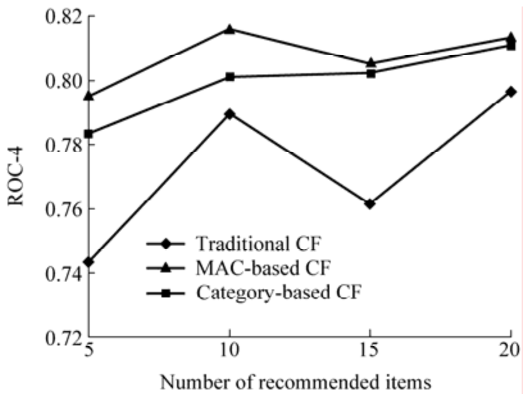

Fig. 5 ROC-4 results

The results in Fig. 5 show that partition-based collaborative filtering using both partition methods performs better than non-partitioned collaborative filtering. On average, MCA-based CF performs 5.7% better than traditional CF, and category-based CF performs 5.3% better than traditional CF. Only small differences are seen between MCA-based CF and category-based CF. As is already shown, the original matrix can be more simply partitioned by category-based CF than by MCA-based CF. So, if items can be classified by their attribute "category", category-based CF is suggested. As the number of recommended items N increases, the performance advantage decreases. If N is very large, traditional CF is as useful as partition-based CF.

5 Conclusions and Future Work

This paper presents an overall solution to the problem of providing an active, personalized information service for every user in a digital library. The techniques described in the paper were compared using a personalized service system and our solution performs quite well on an experimental data set. The newly proposed algorithms overcome the disadvantages of traditional collaborative filtering, i.e., the problem of sparsity, the "new user" problem, and the "new item" problem.

Although unifying partition-based collaborative filtering and meta-information filtering works well in the

DRMS system, more research is needed in particular for the update algorithm used when the data both of users and items change. Future work is also required in generating recommendations from various kinds of resources dispersed over multiple digital libraries. The ultimate objective of a digital library is data sharing and a personalized active information service over a wide area.

References

[1] Claypool M, Gokhale A, Miranda T. Combining content-based and collaborative filters in an online newspaper. In: Proceedings of the ACM SIGIR Workshop on Recommender Systems — Implementation and Evaluation. Berkeley, USA, 1999.

[2] Delgado J. Agent-based information filtering and recommender systems on the Internet [Dissertation]. Nagoya Institute of Technology, 2000.

[3] Lawrence R D, Almasi G S, Kotlyar V, et al. Personalization of supermarket product recommendations. IBM Research Report, IBM T. J. Watson Research Center, New York, 2000.

[4] Mooney R, Roy L. Content-based book recommending using learning for text categorization. In: Proceedings of the Fifth ACM Conference on Digital Libraries. San Antonio, TX, USA, 2000.

[5] Melville P, Mooney R J, Nagarajan R. Content-boosted collaborative filtering. In: Proceedings of the 2001 SIGIR Workshop on Recommender System. New Orleans, LA, USA, 2001.

[6] Pazzani M. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 1999, **13**(5-6): 393-408.

[7] Zeng Chun, Xing Chunxiao, Zhou Lizhu. A personalized search algorithm by using content-based filtering. *Journal of Software*, 2003, **14**(5): 997-1002. (in Chinese)

[8] Sarwar B M, Karypis G, Konstan J, et al. Application of dimensionality reduction in recommender system — A case study. In: Proceedings of ACM WebKDD 2000 Web Mining for E-Commerce Workshop. Bostom, USA, 2000.

[9] Popescul A, Ungar L H, Pennock D M, et al. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-2001). Morgan Kaufmann, San Francisco, USA, 2001.

[10] Rashid A M, Albert I, Cosley D, et al. Getting to know you: Learning new user preferences in recommender systems. In: Proceedings of IUI'02. San Francisco, California, USA, 2002.

[11] Aggarwal C C, Yu P S. Data mining techniques for personalization. *IEEE Bulletin of the Technical Committee on Data Engineering*, 2000, **23**(1): 4-9.

[12] Oyanagi S, Kubota K, Nakase A. Application of matrix clustering to web log analysis and access prediction. In: Proceedings of the ACM WebKDD Workshop on Mining Log Data Across All Customer Touch Points. San Francisco, CA, USA, 2001.

[13] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithm for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. Madison, WI, USA, 1998.

[14] Herlocker J L, Konstan J, Borchers A, et al. An algorithmic framework for performing collaborative filtering. In: Proceedings of the 1999 SIGIR Conference on Research and Development in Information Retrieval. Berkeley, USA, 1999: 230-237.

[15] Gao Fengrong. Research on the key techniques of personalized recommender systems [Dissertation]. Beijing: Renmin University of China, 2004. (in Chinese)