ON THE RECOMMENDER SYSTEM FOR UNIVERSITY LIBRARY

Shunkai Fu, Yao Zhang and Seinminn
Computer Science and Technology, Huaqiao University, No.665 Ave. Jimei, Xiamen, China

ABSTRACT

Libraries are important to universities, and they have two primary features: readers as well as collections are highly professional. In this study, based on the experimental study with five millions of users' borrowing records, our discussion covers: (1) the necessity of recommender system for university libraries; (2) collaborative filtering (CF) technique is applicable and feasible; (3) user-based CF technique is preferred over item-based; (4) the performance of applying classical used-based collaborative filtering algorithm; (5) the effectiveness of local recommendation and the great saving of computing resource it may bring potentially. Since the data size used in our experiments is the largest one among similar studies, it is believed a valuable reference on this specific direction.

KEYWORDS

Recommender system, university library, collaborative filtering, user modeling

1. INTRODUCTION

In most universities, libraries are well equipped with modern IT infrastructure, and large amount of usage data have chance to been collected and stored. As compared with public libraries, university libraries have two primary features, (1) large volume of abstruse collection on science and engineering, and (2) readers have in-depth knowledge on related fields. Meanwhile, due to the explosion of knowledge, the scale of collection increases quickly as well.

For a long time, we depend on the search engine to retrieve items (or books) in the collection. To do so, we may have to iteratively try different keywords, and adjust them given the results if necessary. It repeats until satisfactory items are found. During this procedure, those who know how to represent their requests clearly, especially in a manner "liked" by the engine, may reach their goals soon. Besides, users' carefully made decision and options could not benefit others who have similar requests, although the system has helped to record who has borrowed which books at what time. Hence, in conclusion, search engine is not enough for us to effectively find what we want in university library, and we desire a smarter assistant which could make use of peers' options. In this project, we work to build a suitable recommender for university library, but, in this submission, we only share what we have gained from some early experiments. The discovery may guide our future work.

This paper is structured as follows. In Section 2, related works are reviewed and our contributions are listed. In Section 3, we analysis the necessity, feasibility and a potential solution based on the specific requirements of university libraries. Then, in Section 4, we introduce two primary collaborative filtering algorithms, and discuss the cause of our selection. In Section 5, a series of experiments are conducted, using about five millions of records from one Chinese university, Huaqiao University. We conclude in Section 6.

2. RELATED WORK AND OUR CONTRIBUTION

There are some published works on the recommender system for university library. According to whether there are reported experiments with real data as contained, we filtered out with only trustable ones left, including contributions based on clustering [1, 2], association analysis [3, 4] and collaborative filtering (CF) [5, 6] respectively. Though our discussion focuses on CF as well, it has some obvious differences:

1. We analysis the necessity in a quantitative way (see next section), while [1-6] only provide qualitative analysis;
2. Like [1, 3, 4], we directly use the original borrowing records like <*user_id, book_id, timestamp*>, i.e. each one indicates someone borrowed some book at some time. However, in [5], it uses the circulation log of the reading room. In [6], since it provides online reading, a score is estimated as the preference on behalf of the user, based on how many pages viewed. Borrowing log is used in [1] as well, but a score is also calculated according to the borrowing length (i.e. how long the reader keep the book) and whether renewal happens;
3. For the first time, we discovered that local search is effective to construct top-N recommendation list while applying CF-based recommender for university library (see Section 5). This would bring great reduction on computing complexity as compared with conventional global search way;
4. Previous studies only used tiny or small scale of usage log in their experiments. For example, only 38,078 records were used in [5]; data about 2,358 users and 4,352 books were used in [4]; 7,090 loan records between January and March, 2003 in [3]; records collected between September 2006 and May 2007 were used in [1]; two semesters of records in [2]. In our study, totally 4,932,579 borrowing records are used, with a span of 10 years, and it is at least ten times of the largest one found in previous studies. Hence, it is closer to the real application scenario.

3. NECESSITY AND FEASIBILITY

How to bridge the ‘gap’ between the professional readers and involved scholarly collections is one of the most challenging tasks faced by university libraries. Recommender system, due for its success in many fields in the past two decades, is believed an economic and workable solution.

3.1 Necessity

The necessity of building recommender system for university library can be explained with, but not limited to, the following points:

- Fundamental information system in library is mature, including easy-to-use text retrieval function. However, we have to transform our own abstract and/or abstruse requests into one or few keywords, facing the traditional search engine. This step demands intensive brain work, and it may greatly influence the quality of retrieval. Worse thing is that it only returns text-related results, preventing us from accessing books unknown but may interested us [7];
- Meta information of each book in the database is rare, which furthermore restricts retrieval function. For example, most books in our university’s collection only contain basic sections like titles, authors and publishers, but no abstract, saying nothing of contents. The shortage of these rich information will make current text retrieval function work poorly;
- Low penetration rate due to the fact that our collection is abstruse. For example, in our library, totally there are around two millions of volume and 450 thousands unique items. However, only 47.51% of (unique) items once be borrowed. Can we declare that those remaining items are just not interesting? Or can we say that they are not known by readers? I think the second is more likely, so we need a ‘smart’ recommender system to help readers in university library find what they want, especially those they don’t know yet but may feel interested;
- We hope such a system won’t disturb us, e.g. we need to inform it what we want now and then explicitly, but just use the recorded our usage data and work ‘silently’. By assuming that we borrow a book only when we like it, existing recommender technique is an ideal candidate since it could use these implicit data to produce something useful.

3.2 Feasibility

There are two primary information filtering techniques: content-based [8, 9] and collaborative filtering-based [10, 11]. Most methods used in content-based come from information retrieval, and it has limits in, at least, three aspects:

- It works by comparing if content features are matched with user profiles. To do so, we need some way to extract features of the target content, and furthermore, we want the features to be complete as well as concise. However, there is no effective approach to do feature selection, especially when we face multimedia resource, and extracted features alone cannot tell us the quality of information, which may influence users' satisfaction level. Finally, our fact is that we only have rare information about the book, as discussed above;
- Even though we have enough information on content, constructing users' profile is again a challenging task since it may also rely on the feature extractions of contents;
- Content-based filtering technique may disturb users since their privacy would be 'touched', due to the inborn mechanism of this approach.

Hence, collaborative filtering (CF) based recommendation is the remaining option. The most feature of this technique is its independence of content, therefore it is applicable to not only text but multimedia. Its basic assumption is that users similar to each other may have similar interest, i.e. borrowing similar books here. Some observations and features of this application are listed as below, and they are solvable by CF technique, or useful to the success of this technique:

- Meta data about books is rare. Besides, on most conventional library system, readers have no way to comment or give explicit feedback online, though this is quite common on Internet today;
- Readers' borrowing records can be viewed as their profile. Since readers normally make the selection only after careful checking and consideration, circulation record could be a strong indication that whether one user 'like' a book;
- Enough readers' borrowing records are collected. Totally we have around five millions of borrowing records. Because of the relatively high frequency of circulation in university, new records emerge and are collected every day, which could be used to update users' profile;
- Borrowing behavior is of implicit type of feedback, without having to modify the existing system, and bring no interrupt during users' interaction with system;
- Implementation based on CF technique may minimize the influence on existing three-tier system (Web/Application/Database). Because what data required by the recommender can be retrieved from database directly, we need not intervene with application layer. For security and performance consideration, we may create an independent database on the recommender side, and synchronize it with online database according to acceptable strategy, e.g. incremental updating during low transaction volume period;
- Outputs by the recommender can be exposed with REST (Representational State Transfer) interface, and easily be plugged into existing application systems (see Figure 1). For example, what produced by our recommender may be presented as "Users who borrow this book also borrow them";

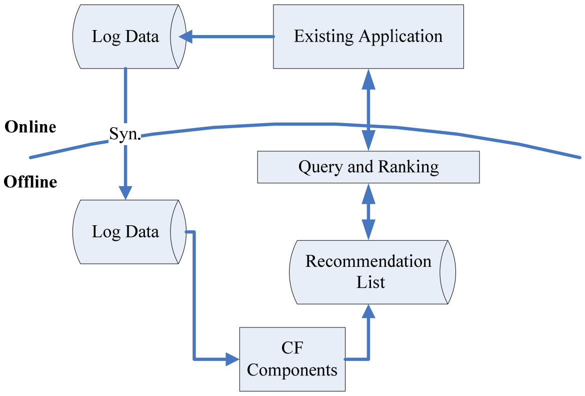

Figure 1. Overall logical view of the existing application and the recommender

- Research on collaborative filtering has been lasted for over 15 years, and there are many successful applications [12-14].

Therefore, CF is chosen in this project. Like clustering algorithms, CF also depends on the search of near neighbors, but CF can provide more fine-grained results. Besides, classical k-means algorithm also requires users to specify the choice of parameter k. As compared with association analysis, CF could tell us more previously unknown knowledge from similar users, and in a faster way. To avoid generating too many rules, association analysis has to increase the support and confidence level, which will influence the discovery of some new items.

4. COMPARISON OF CF ALGORITHMS

4.1 Overview

There are two kinds of CF algorithms, user-based [10] and item-based [11]. Both use user-item score matrix and some active user as input, and produce an item recommendation list for this active user.

The so-called user-item score matrix is the core data structure in CF algorithm, and it is of size $m \times n$, i.e. m users' preferences/options on n items. It is denoted as $S_{m \times n}$. The value of each cell $S(u, i)$ expresses user 's preference on item i, and it can either be boolean or numeric. In our application, $S(u, i)$ is boolean, with 1 indicates that user u once borrowed item i and 0 for not. Similarly, $S(u,\cdot)$ refers to user u's circulation vector, and $S(\cdot, j)^T$ for the circulation vector about item j.

Given $S_{m \times n}$, the kernel step in CF is to determine the similarity between users/items. Widely used measures include cosine similarity, Pearson's correlation, various revised cosine similarity and *Tanimoto* coefficient (also called *Jaccard* index/coefficient often) for binary data [15], which can easily be found in nearly all statistics books. In our application, *Tanimoto* coefficient is used, and its equation in user-based scenario is shown below:

$$sim(u, v) = \frac{S(u,\cdot)S(v,\cdot)^T}{\|S(u,\cdot)\|^2 + \|S(v,\cdot)\|^2 - S(u,\cdot)S(v,\cdot)^T} \quad (1)$$

The corresponding equation given item-based calculation is similar

$$sim(i, j) = \frac{S(\cdot,i)^T S(\cdot,j)}{\|S(\cdot,i)^T\|^2 + \|S(\cdot,j)^T\|^2 - S(\cdot,i)^T S(\cdot,j)} \quad (2)$$

4.2 User-based CF

It is the earliest CF algorithm being proposed. Based on the calculation of user similar (see equation (1)) , we are able to find k users closest to the active user a. Then, for each item $x \in \bigcup_{u=1}^{k} \{j | S(u, j) = 1\} \setminus \{y | S(a, y) = 1\}$, we can predict user a's possible preference on x as below:

$$S(a, x) = \frac{\sum_{i=1}^{k} (sim(a,i) \times S(i,x))}{\sum_{i=1}^{k} sim(a,i)} \quad (3)$$

By sorting all such $S(a, x)$ in descending order, we could produce a recommendation list containing N items with the highest score, i.e. the known top- N recommendation.

4.3 Item-based CF

For some online recommender, the 'long tail' effect of users is more prominent than item's. For example, there are hundreds of millions of users, but millions of items, on Amazon.com. Furthermore, their users' amount and profiles changes all alone, so we have to calculate the similarity of users online if we hope to catch users' updates in time, which will slow down the response speed of recommender built upon user-based CF technique. Differently, similarities between items are relatively stable, so their similarity could be calculated offline in item-based CF, and be updated periodically (and also offline), which ensures the speed of online response. Hence, item-based CF is preferred on most large scale Web sites when construct their own recommender systems [13, 16].

4.4 Our Choice

In our project, used-based CF algorithm is finally selected based on the following causes:

- There are around 70 thousands of users and 500 thousands of books (items) in database, and more items are added to the collection than newly enrolled students per year;
- Item-based CF will compute the prediction on an item *i* for a user u by computing the sum of the ratings given by the user on the item similar to *i*, and each ratings is weighted by the corresponding similarity *sim(i,j)* between items *i* and *j*. However, the 'interest' of readers in university is NOT stable – it may be completely different between adjacent semesters. Hence, what one read in this semester may have little influence on what s/he would select in the next semester, which limits the 'power' of item-based CF algorithm;
- Based on our experimental results, readers' interest is rather stable (see the next section) in macroscopic perspective, i.e. limited to their profession. This means that readers from the same major will have share on reading selections, and this is the basis of user-based CF.

5. EXPERIMENTS: DESIGN, DISCOVERY AND DISCUSSION

There are two kinds of evaluation for recommender systems, online and offline. Online evaluation requires the recommender to work online with existing application to collect actual interaction log. Its cost is high, and it is not suitable for repeating test, especially when the work is not mature since it may sacrifice the experience of users. Most research work on recommender takes offline experiments, i.e. feed recommender system with partial history data to train it, and compare the predicated value with those parts left but observed in history log.

5.1 Dataset

Dataset used in this project is from the library of Huaqiao University, ranging from 1998 to 2011. There is various information as contained, but we only use the users' borrowing records, basic profile (including which department s/he belongs to) and brief description of books. Related summary statistics can be found in Table 1.

Table 1. Summary statistics of dataset

Total # of borrowing records		4,932,579	
Total # of books in system	445,722	**Total # of books with borrowing record**	211,776
Total # of users in system	76,524	**Total # of users with borrowing records**	65,483
Avg. # records per users	75	**Avg. # records per books**	23

Regarding books, we only count the number of different books, ignore multiple copies

In the current dataset, due to some unknown reason (probably early users' borrowing records were not saved), there are 11,041 users with no borrowing records. Even we remove these users and those books with no records from consideration, the user-item matrix is still very sparse, about 0.036% (=4,932,579/(65,483 211,776)).

We also extract summary statistics about two typical colleges, Materials Science and Engineering (MSE) and Public Administration and Service (PAS), as Table 2.

Table 2. Summary statistics of two colleges: MSE and PAS

	College of MSE	**College of PAS**
# of lending records	548,998	270,351
# of users with records	7,277	3,377
# of books covered	87,745	63,925
Avg.# of records per user	75	80

5.2 Experimental Design and Discussion

It is mentioned (in last section) readers from the same college/department would be similar one another on reading interest, so we could only refer to these users' preferences during the decision making in used-based CF. If this could be confirmed, we need only conduct a local search of neighbors, instead of looking for similar users from the whole user set. To do so, we select MSE and PAS colleges as examples. We compare the recommendation effect on MSE and PAS given a recommender trained by the whole dataset (see Table 1) and dataset related to these two colleges(see Table 2) respectively.

Given the dataset, we take the 10-folder strategy, i.e. dividing the data into ten groups, selecting one for testing and the remaining ones for training. By repeating the experiments for ten times, we measure the average precision and recall. Although there are various measures proposed to evaluate one recommender [17], such as coverage and serendipity, accuracy is still the most concerned index.

As mentioned in last section, we implement the recommender as user-based, and Tanimoto coefficient is applied to measure the similarities. We select K neighbors, and study the top-N recommendation list. By differentiating K and N, we have the chance to compare their influence on recommendation accuracy.

5.3 Recommender trained with whole Dataset

Firstly, we train the recommender with whole dataset, and use it to provide recommendation list for MSE and PAS colleges. This is the typical mechanism in conventional CF-based recommender. In experiments, K is set as 10, 20, 30, 40 and 50 respectively; N is set 10, 20 and 30.

It is observed from Figure 1 that, (1) Both precision and recall increase with larger K, e.g. the precision of MSE has 34% increase when K increases from 10 to 50 given $N = 10$; (2) Given the same K , smaller N results with higher precision but lower recall. For example, the precision of PAS when N =10 is about 38% higher when N =30 given K=10; while the corresponding recall when N =10 is only 46% of that when N =30.

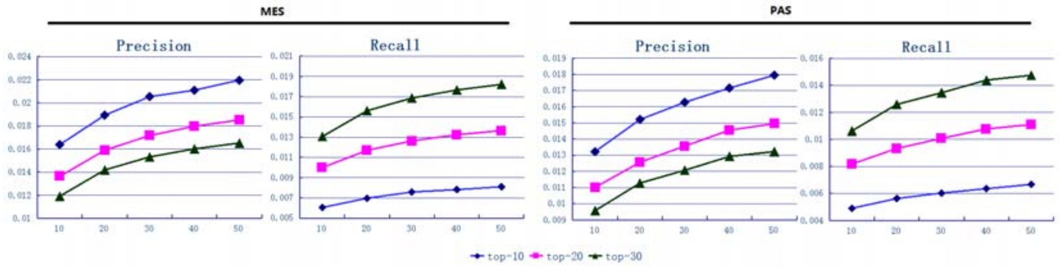

Figure 2. Average precision and recall about MSE and PAS by recommender trained with the whole dataset

5.4 Recommender trained with Partial Dataset

Secondly, we train the recommender with partial dataset. For instance, a recommender is trained by the MSE dataset, and used to provide recommendation list for readers belonging to MSE. This is different from the conventional solution, and the remaining settings are kept the same.

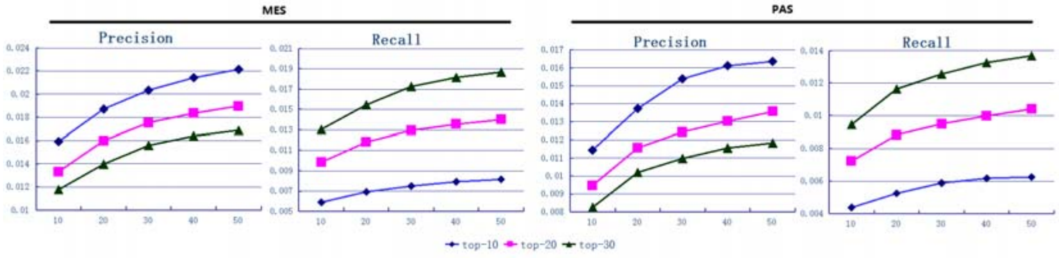

Figure 3. Precision and recall about MSE and PAS by recommender trained with only MSE and PAS dataset respectively

Similar trend as in Figure 2 is observed from Figure 3, (1) both precision and recall increase with larger *K*; (2) given same *K* , smaller *N* results with higher precision but lower recall.

5.5 Global vs. Local Recommendation

For easy reference, we call recommender trained by whole dataset as global recommendation, and that by partial dataset as local recommendation.

Table 3. Precision (P1) and Recall(R1) achieved by whole data VS. Precision(P2) and Recall (R2) achieved by partial data (P2) about MSE's users: (P1/P2) and (R1/R2)

N K	***10***		***20***		***30***	
	P1/P2	***R1/R2***	***P1/P2***	***R1/R2***	***P1/P2***	***R1/R2***
10	*1.03*	*1.03*	*1.02*	*1.02*	1.01	1.00
20	*1.01*	*1.01*	*0.99*	*0.99*	1.02	1.01
30	*1.01*	*1.01*	*0.98*	*0.98*	0.98	0.98
40	*0.99*	*0.99*	*0.98*	*0.98*	*0.98*	*0.97*
50	*0.99*	*0.99*	*0.98*	*0.98*	*0.98*	*0.98*

Table 4. Precision (P1) and Recall(R1) achieved by whole data VS. Precision(P2) and Recall (R2) achieved by partial data (P2) about PAS's users: (P1/P2) and (R1/R2)

N K	***10***		***20***		***30***	
	P1/P2	***R1/R2***	***P1/P2***	***R1/R2***	***P1/P2***	***R1/R2***
10	*1.16*	*1.12*	*1.16*	*1.13*	1.16	1.12
20	*1.11*	*1.07*	*1.09*	*1.06*	1.11	1.08
30	*1.06*	*1.03*	*1.09*	*1.06*	1.10	1.07
40	*1.06*	*1.03*	*1.12*	*1.08*	*1.12*	*1.09*
50	*1.10*	*1.06*	*1.10*	*1.07*	*1.12*	*1.08*

Table 3 and Table 4 are about the comparison of precision and recall achieved by global and local recommendation on MSE's and PAS's readers respectively. It can be observed and concluded that:

- Local recommendation may even achieve higher precision and recall than global recommendation, which can be observed from the gray cells in Table 3. However, this is NOT observed in Table 4, which may be explained that engineering readers' (e.g. college of MSE here) interest is NOT as diverse as that of non-engineering readers (e.g. college of PAS here);
- Regarding MSE's readers, by average, the global recommendation achieves 10.9% and 7.6% higher precision and recall respectively than local recommendation. Considering the great computing complexity saved by local recommendation, this cost is affordable;
- *K*=30 and *N*=20 are the suggested parameters observed in our experiments.

6. CONCLUSION AND FUTURE WORK

In this article, we discuss the necessity of building recommender system for university library firstly. Then, we compare and propose that user-based CF algorithm is more suitable for this application. We conduct experiments with five millions of lending records collected in one Chinese university, National Huaqiao University. The results confirm that we could train individual recommender for each college (or even department), or at least, show recommendations from these individual recommenders with higher priority than those from global recommender. This finding could bring great reduction on computing complexity, with affordable cost on recommendation accuracy.

Although we focus only on accuracy here, we are interested to study the serendipity in future study, aiming at better service. However, till now, there is no mature definition and measure on serendipity.

Meanwhile, how to tradeoff the accuracy and serendipity is an interesting topic as well. More experiments will be designed and conducted to help us gain more knowledge and experience about the application of recommender in university library, before we start to develop the real system.

ACKNOWLEDGEMENT

This research is supported by Fujian Engineering Technology Research Center on Green Communication and Intelligent Information Service (2012H2002), Huaqiao University Grant 11Y0274, 12HJY18 and the Fundamental Research Funds for the Central Universities (11J0263).

REFERENCES

Sun, S.-y. and W. Wang, 一种基于用户聚类的协同过滤个性化图书推荐系统. Modern Information, 2007. 27(11).

Wu, J.-w., X.-h. YU, and W.-q. CHEN, Density-based Dynamic Collaborative Filtering Books Recommender Algorithm. Application Research of Computers, 2010. 27(8).

Zhao, L., The Design and Implementation of the Bibliographic Recommendation System Based on Maximal Frequent Patterns Mining Algorithm. New Technology of Library and Information Service, 2010. 5: p. 23-28.

Chen, K., Simulation of Book Recommender System Based on Eclat Algorithm. Computer Simulation, 2010(9): p. 311-314.

Kun, D., Research of Personalized Book Recommender System of University Library Based on Collaborative Filter. New Technology of Library and Information System, 2011. 11.

Zeng, Q.-h. and Y.-h. Qiu, An E-Book Recommender System with Collaborative Filtering. Computer Science, 2005. 32(6).

马张华, 论主题检索系统中先组词的选择和使用. The Journal of The Library Science in China, 1997. 23(2): p. 15-19.

Mooney, R.J. and L. Roy. Content-Based Book Recommending Using Learning for Text Categorization. in SIGIR Workshop on Recommender Systems: Algorithms and Evaluation. 1999. Berkeley, CA, USA.

Meteren, R.V. and M.V. Someren, Using Content-Based Filtering for Recommendation. 2000.

Resnick, P., et al. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. in ACM Conference on Computer Supported Cooperative Work. 1994. Chapel Hill, NC: ACM.

Sarwar, B., et al. Item-Based Collaborative Filtering Recommendation Algorithms. in 10th International Conference on World Wide Web (WWW). 2001. ACM.

Goldberg, D., et al., Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM, 1992. 35(12): p. 61-70.

Linden, G., B. Smith, and J. York, Amazon Recommendation: Item-to-Item Collaborative Filtering. Internet Computing, 2003. 7(1): p. 76-80.

Shardanand, U. and P. Maes. Social Information Filtering: Algorithms for Automating 'Word of Mouth'. in ACM CHI Conference on Human Factors in Computing Systems. 1995. ACM Press.

Mild, A. and T. Reuterer, An Improved Collaborative Filtering Approach for Predicting Cross-category Purchases Based on Binary Market Basket Data. Journal of Retailing and Consumer Services, 2003. 10(3).

邓爱林, 朱扬勇, and 施伯乐, A Collaborative Filtering Recommendation Algorithm Based on Item Rating Prediction. Journal of Software, 2003. 14(9).

L.Herlocker, J., et al., Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information System (TOIS), 2004. 22(1): p. 5-53