A Personalized Recommender System Based on Library Database

https://doi.org/10.3991/ijet.v12.i12.7964

Chunyan Xu
Chongqing College of Electronic Engineering, Chongqing 400000, China
hongxiuwuyue@163.com

Abstract—In recent years, the university libraries in China have acquired increasingly abundant electronic resources. However, the information silo phenomenon appears due to the lack of connection between university IT system and the community. Based on the book borrowing, favourite collection, comments and social relationship of students, this paper digs into the personalized interests of students, and promotes the design and implementation of a personalized recommender system. Specifically, the overall framework and recommender engine of the system were created based on the library data services. The modules in the system were also elaborated, and the recommendation results were verified by an offline test.

Keywords—recommender system, personalized service, digital library.

1 Introduction

With the explosive development of the IT industry, the information tidal wave has swept across universities in China. The shortage of teaching resources and hardware facilities have also been alleviated, thanks to techniques like office automation, information management, curriculum management and digital libraries [1-3]. Despite the increasingly abundant electronic resources in universities, the information silo phenomenon appears due to the lack of connection between university IT system and the community. Thus, it is imperative for the universities to provide students with personalized services through integration of their E-campus information resources [4-6].

To answer the call, this paper introduces the relatively mature concept of recommender system in E-commerce and library database into the design and implementation of a personalized recommender system for university students [7]. The overall framework and the recommender engine reflect the modularity principle in software engineering, which greatly improves the flexibility and scalability of the system. Besides, the recommended modules can satisfy the demands of different users. Experimental results show that the proposed system, coupled with the library database, achieved desirable recommendation results [8-10].

2 System design

2.1 Framework of personalized recommender system

As shown in Figure 1, a recommender system acts as an intermediary between students and library resources. The students can be associated with the resources based on their favourite items or the interests shared with other students. The association lays the basis for the design of a feature-based recommender system (Figure 2).

Upon the entry of a student, the system generates the features of each student, finds out the resources associated with him/her, and eventually forms a list of recommended items. Therefore, the system has two core tasks: the generation of features for a given student, and the search for suitable resources based on the features.

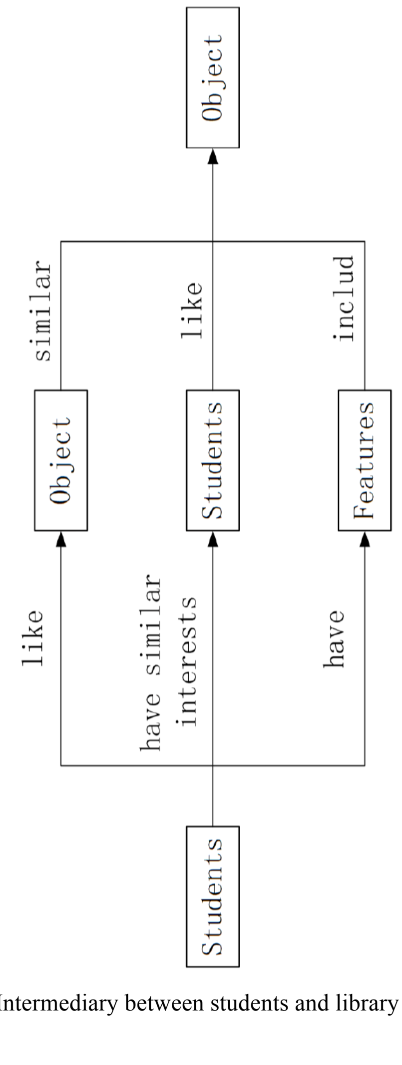

Fig. 1. Intermediary between students and library resources

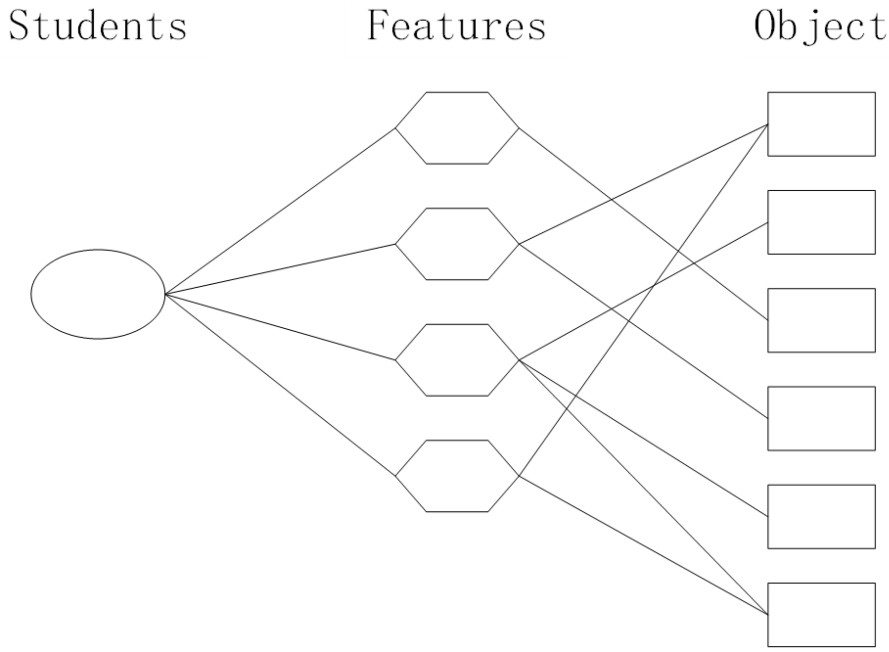

Fig. 2. Structure of feature-based recommender system

The user features mainly fall into the categories below:

Demographics. The age, gender, ethnicity and other information provided by the user at the time of registration.

User behaviours. The books borrowed, the resources browsed, and the comments left by the user; the user behaviours can be classified into near-term behaviours and long-term behaviours.

User themes. The computer network and the data structure can be converged into different topics by the topic model based on the historical behaviours of the user, so as to calculate the interested topic of each student.

The recommender system covers various tasks, namely the addition of new resources into the library, the publication of resources as required by the university, and the recommendation of different library resources to students. It would be too complex to include all these features and tasks in one system, not to mention weighting the features and tasks one by one. Therefore, multiple engines are required for the recommender system, each of which is responsible for a class of features and a type of tasks. In this way, the recommender system only needs to merge the results of recommender engines by a certain weight or priority, and return the sorted results to end users (Figure 3).

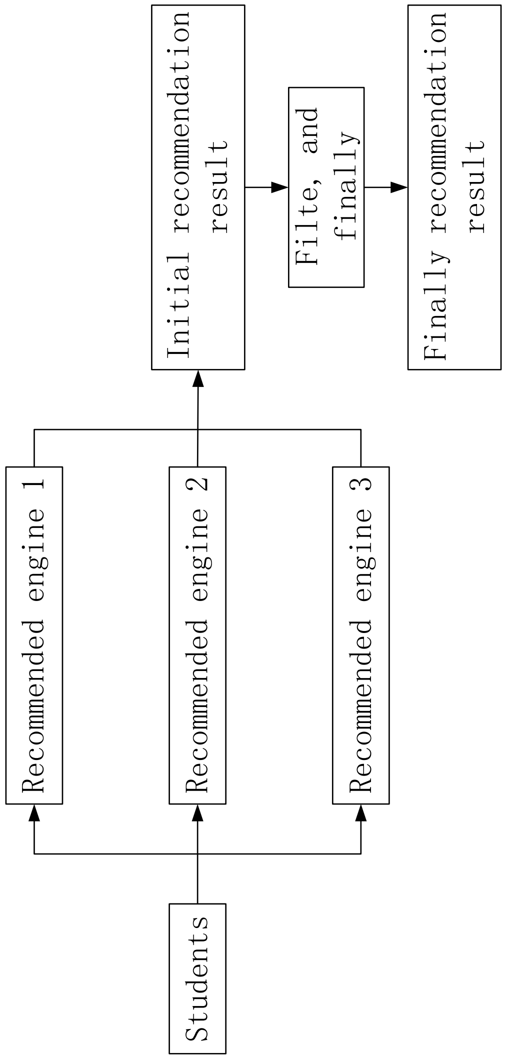

Fig. 3. Structure of recommended system

2.2 Design of recommender engines

Based on one or more user features, a recommender engine generates a list of recommended items according to the recommendation strategy. The engine consists of three parts (Figure 4): Part A, Part B and Part C.

Part A fetches and analyses student behaviour data from the database or cache, and then generates the eigenvector of the current student; Part B transforms the eigenvector into a list of initial recommended items based on a correlation matrix of feature items; Part C filters and ranks the items in the list to form the final recommendation list.

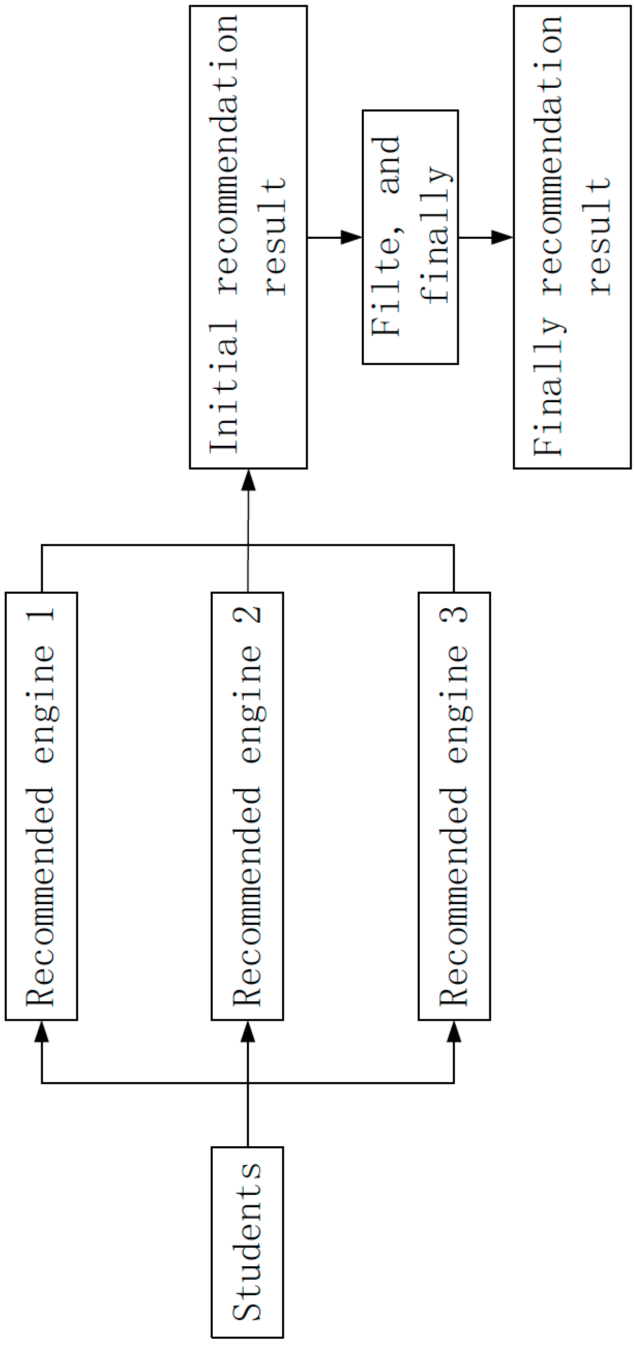

Fig. 4. Structure of recommender engine

2.3 Generation of user eigenvector

The eigenvector, consisting of feature and weight, was calculated considering the following factors.

Type of user behaviours. The user behaviours include but not limited to book browsing, borrowing, favouring, rating, evaluating, labelling and sharing. These behaviours have different impacts on the weight of item features in the resources. It is often difficult to determine the relative importance of each behaviour. A general standard is that a behaviour weighs heavier if the user pays more cost.

Time of behaviour. In general, the recent behaviours are more important than those occurred a long time ago. If a book was borrowed recently, its features should be assigned relatively high weights.

The number of user behaviours. Sometimes, the students make repeated attempts on the same item. For example, a book or the books of the same type may be borrowed for many times. The number of user behaviours concerning the same item demonstrates their interests in the item. In other words, the number of relevant behaviours is positively proportional to the weight of the features.

The popularity of items. The borrowing of a popular book does not necessarily represents the feature of a student. There is a high possibility that the behaviour arises out of conformity. On the contrary, the borrowing of a less popular book can reflect the individual demand of a student. Hence, the recommender engine should assign greater weight to student features concerning unpopular items.

3 Design and test of personalized recommender system

3.1 Design

The resource-student track-down list was presented below:

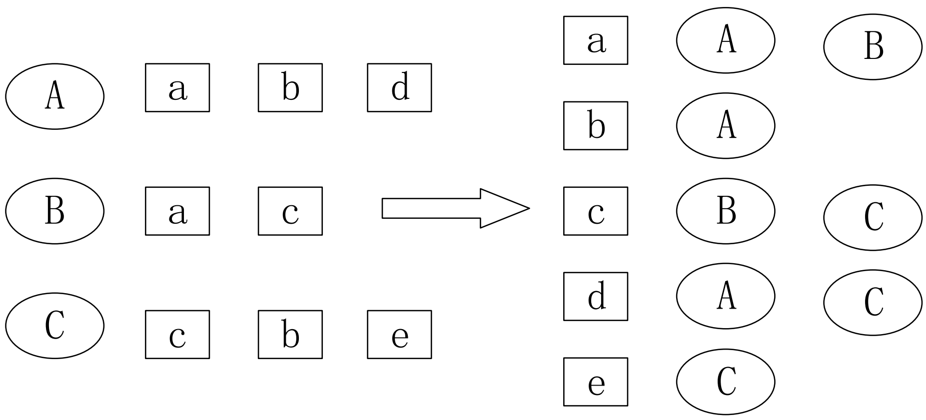

Fig. 5. Resource-student track-down list

In Figure 5, each resource corresponds to a list of students who have ever acted on it. Let $C[u][v] = |N(u) \cap N(v)|$ be a sparse matrix, where u and v are the attributes of the students corresponding to resource k, i.e. $C[u][v] = k$. It is possible to obtain the list of students corresponding to each resource in the track-down list by adding 1 to $C[u][v]$. In this way, the $C[u][v]$ on which none of the students act can be generated.

After obtaining the similarity of user preferences, the author introduced the user-based collaborative filtering algorithm (CFA) to recommend the resources most favoured by k students to the end users.

3.2 Test

The user-based CFA has a key parameter k, which represents k students sharing similar interests in the same item. The books, groups, tags, etc. favoured by these students are recommended to the end users. Based on the datasets synchronized for local use, this subsection carries out an offline test to measure the performance indices for the algorithm at different k-values.

Table 1. Performance at different K-values

k	Accuracy	Recall rate	Coverage	Popularity
5	15.7	8.9	51.9	6.4
10	19.4	9.2	43.1	7.1
20	22.1	11.1	33.9	6.9
40	23.8	11.3	24.8	7.2
80	25.4	13.2	19.2	7.1
160	23.5	12.8	16.1	7.3

As shown in the Table 1, the variation in k-value has a certain impact on the performance of the algorithm:

Accuracy and recall rates: The accuracy rate of the recommender system is not linearly correlated with parameter k. In the local dataset, a relatively high accuracy and recall rates were achieved at about k=80, indicating that k value is the key to high system accuracy. Of course, the accuracy of recommended results is not particularly sensitive to k. The results were rather accurate as long as k fell within a certain area.

Popularity: The greater the k in the dataset, the more popular the recommended result. This is because the k-value determines the number of students sharing similar interests in an item recommended to target users. Therefore, the k-value is positively correlated with the number of students for reference, and the chance of recommending the global hot resources.

Coverage: The coverage of recommended results decreased with the increase in k-value. The decrease is attributed to the fast spread of popular resources, which lowers the hit rate of long tail resources by recommender engines and thus narrows down the coverage.

4 Conclusion

In e-commerce, the recommender system is often relied on to link up users with goods. Recent years has seen the popularity and progress of the personalized recommender system. Based on the book borrowing, favourite collection, comments and social relationship of students, this paper describes the overall architecture of a personalized recommender system, and introduces the structural design of recommender engines. Several modules were also created to generate user eigenvectors. Finally, the algorithm of these modules were introduced and verified by an offline test.

5 References

[1] Guo, K. (2016). Empirical study on factors of student satisfaction in higher education. Revista Iberica de Sistemas e Tecnologias de Informacao, E11: 344-355.

[2] Jamie, L., Mark, B., Emily, K.F. (2017). Using virtual environments to investigate wayfinding in 8- to 12-year-olds and adults. Journal of Experimental Child Psychology, 166: 178-189.

[3] Mohammed, K., Ahmed, A.H., Sadiq, A. (2017). Concept model for the second life cycle of vehicles in Palestine. Procedia Manufacturing, 8: 707-714. https://doi.org/10.1016/j.promfg.2017.02.091

[4] Manuel, B., Andreas, H.J., Miriam, A.L. (2016). Interaction and space in the virtual world of Second Life. Journal of Pragmatics, 101: 83-100. https://doi.org/10.1016/j.pragma.2016.05.009

[5] Mashrura, M., Jennifer, S., Faisal, K. (2017). Human performance data collected in a virtual environment. Data in Brief, 15: 213-215. https://doi.org/10.1016/j.dib.2017.09.029

[6] Thomas, D.P., Timothy, M. (2017). An initial validation of the virtual environment grocery store. Journal of Neuroscience Methods, 291: 13-19. https://doi.org/10.1016/j.jneumeth.2017.07.027

[7] Dolores, M.G., Salvador, B., Jan, N. (2016). Second life adoption in education: A motivational model based on uses and gratifications theory. Computers & Education, 100: 81-89. https://doi.org/10.1016/j.compedu.2016.05.001

[8] Ahmed, M.A., Krishna, K.S. (2015). Interactive e-learning through second life with blackboard technology. Procedia - Social and Behavioural Sciences, 176: 891-897. https://doi.org/10.1016/j.sbspro.2015.01.555

[9] Amber, M. (2014). Sensemaking in second life. Procedia Technology, 13: 107-111. https://doi.org/10.1016/j.protcy.2014.02.014

[10] Adam, S.S., Anna, M.L. (2017). Virtual intimacy: Propensity for physical contact between avatars in an online virtual environment. Computers in Human Behaviour, 78: 1-9.

6 Author

Chunyan Xu is with Chongqing College of Electronic Engineering, Chongqing 400000, China.

Article submitted 23 October 2017. Published as resubmitted by the author 15 december 2017.