Collaborative Filtering Recommendation Algorithm Based on Contextual Information

Jia GUO*, Jian-Jing SHEN
The Information Engineering University, Henan, China
soap_guo@163.com
*Corresponding author

Keywords: Contextual Information, Collaborative Filtering, Curricula-variable.

Abstract. Contextual information refers to all the factors affecting the recommendation system and supporting recommendation except “user - item” assessment information. To make effective recommendation to the user against a specific context, it is very important to integrate contextual information into recommendation. Based on contextual information, this paper improves collaborative filtering recommendation algorithm, applying the algorithm to the curricula-variable recommendation context so as to improve the accuracy of recommendation results and provide certain guidance and suggestions to curricula-variable process.

Introduction

Research on early user behavior analysis shows that the user presents different preferences for different goods against different environments, that is, the user has his interest variable and change with the environment [1]. It is not proper to consider the user and the item only. In order to make an effective recommendation to the user against certain situations, it is necessary to integrate contextual information into recommendation. Dourish pointed out different applications should consider different types of contextual information, which have different impacts on different recommended persons. The recommendation system should produce recommendations apt to the environment as much as possible based on user’s different environments. This paper applies collaborative filtering recommendation algorithm based on contextual information to the curricula-variable system in colleges and universities, analyzing students’ specific contextual information based on their personal interests, development needs and quality, integrating contextual information into the course recommendation, making the recommendation results more consistent with their needs, improving their study motivation, reducing their blindness and bigotry, and providing better guidance for their curricula-variable.

Collaborative Filtering Recommendation Algorithm

Collaborative Filtering Recommendation

With the development of information technology and the Internet, people gradually approach the information-overloading age from the information-lacking age. Information overloading makes information in large amounts, of poor quality and low value, which has become a serious problem. In response to information overloading, a lot of information filtering tools have emerged, such as portals and search engines, which, however, are based on
people-oriented mainstream demands, and the way users access information is “passive”. The recommendation system is an intelligent software tool providing proposals to users, serving as a very promising solution to information overloading. It actively provides useful information to users, filtering information and in line with their individual needs [2]. Different from search engines, the recommendation system does not require user’s clear demands. Through analysis of user’s historical behavior, it takes the initiative to have interest modeling and consequently meet their interests and needs.

The collaborative filtering model has been the most famous recommended model so far, typically applied in the world’s largest B2C e-commerce website Amazon, web resource excavator Digg, Stumble Upon and social music service Last.fm and ilike [3]. The collaborative filtering recommendation method is mainly used to predict the current users’ favorite or interest based on their previous behavior.

The user-based collaborative filtering algorithm relies on similarity. User’s preference similarity is used to have data analysis. Item resources the user might be of interest are recommended to the target user. User-based collaborative filtering algorithm procedure is shown in Figure 1.

![Figure 1. User-based Collaborative Filtering Algorithm Procedure.](image)

**Improvement Based on Context Information**

The recommendation system calls factors explaining contextual environment contextual information, which will be determined according to specific application needs and user’s requirements. There has been no unified definition [4]. In the e-commerce recommendation field, the contextual information may include multiple factors such as user’s purchase intention, season, holiday, location and companion. Wider contextual information may also include device type, social networks and emotion. Therefore, contextual information can be understood as all the factors affecting the recommendation system and supporting recommendation except “user - item” assessment information.

Integrating contextual information into the recommendation model is vital to situation recommendation. Contextual information integration can be divided into two categories: 1. Recommendation based on contextual information retrieval; 2. Preference extraction and prediction based on contextual information. The former searches resources according to contextual information, and provides recommendation with the traditional recommendation model, which is widely applied in mobile tourism recommendation system. The latter divides the situation into 3 categories: context pre-filtering, context filtering and context modeling, based on contextual information study and outline of user’s interest.

Before recommendation is made, context pre-filtering uses contextual information to filter irrelevant “user - item” assessment data to construct a data set in line with the current situation, and then uses the traditional recommendation model and algorithms to filter the data set to produce recommendation results for input.

Context filtering applies contextual information to improve recommendation results. In recommendation generation stage, the impact of contextual information is not considered.
Top-K recommendation list is generated based on the traditional recommendation model. Top-K recommendation list is filtered or adjusted to generate the final recommendation result in line with the context.

Context modeling refers to the entire process to integrate contextual information into the recommendation generation, redesign models and algorithms to deal with multi-dimensional context user’s preference. It is necessary to deal with high-digit data, which is the most complex and the most effective to manage the relationship between the user, item and context, and applicable to the case that the contextual information and user’s preferences are tightly coupled.

**Contextual Information-based Collaborative Filtering Curricula-variable Recommendation**

**User Based Collaborative Filtering Recommendation Algorithm**

First of all, it is necessary to establish the student – foundation course ratings matrix model, select the basic course student achievement as the evaluation matrix. It is understood that user based collaborative filtering recommendation algorithm [6] is applicable when the number of users is much larger than that of items and the calculation result is better. And it is easier to obtain the result data and at the same time then inaccurate results caused by scoring matrix sparse is avoided.

Student: $U = \{u_1, ..., u_n\}$
Foundation course: $F = \{f_1, ..., f_m\}$
Curricula-variable course: $S = \{s_1, ..., s_n\}$

Student - foundation course ratings matrix $R$: $\forall r_{ij} \in R \quad i \in \{1, ..., n\} \quad j \in \{1, ..., m\}$

In $r_{ij}$, i: student, j: foundation course. Under normal circumstances, the results range [0-100] is $v_{ij}$ value range, student - foundation course ratings matrix is taken as ratings matrix. In the ratings matrix, senior students of the same major as that of target students, collaborative filtering algorithm is used to compare and analyze data information of target students and all the students in ratings matrix to find similarity highest group K, and establish relevant nearest neighbor set.

Pearson correlation coefficient: student $u_a$ and student $u_i$ similarity $sim(u_a, u_i) \quad [7]$

$$sim(u_a, u_i) = \frac{\sum_{y \in R_a \cap R_i} (v_{ay} - \bar{v}_a) (v_{iy} - \bar{v}_i)}{\sqrt{\sum_{y \in R_a \cap R_i} (v_{ay} - \bar{v}_a)^2} \sqrt{\sum_{y \in R_i} (v_{iy} - \bar{v}_i)^2}}$$

(1)

The above formula refers to UCF similarity specific algorithm, where $sim(u_a, u_i)$ is target student and matrix student similarity; y is their common rated item, namely their common foundation course; $r_{ay}$ and $r_{iy}$ are student a and student i item assessment, namely student foundation course result; $\bar{v}_a$ and $\bar{v}_i$ are average item assessment value, namely student average result.

**Curricula-variable Recommendation Characteristic Context**

(1) Total quantity control. Curricula-variable recommendation total quantity control involves two aspects: total quantity control of credits students need and limit to the number of
students of curricula-variable courses. The credits of curricula-variable courses students need are regulated. Each student’s energy is limited. More courses does not mean better. It is proper to optimize the quantity of curricula-variable courses. Each curricula-variable course has a limit to the number of students. In case of a very popular curricula-variable course, it is impossible to recommend it to every student. Once the limit is reached, the course shall be no longer variable, that is, curricula-variable course recommendation should completely eliminate “Matthew Effect” [5].

(2) Higher real-time requirements. Curricula-variable recommendation relies on the curricula-variable system. A large number of students simultaneously access the curricula-variable system within a certain period of time. It is necessary to have real-time judgment about whether each course reaches the limit to the number of students, whether each student curricula-variable course conflicts with other selected courses in school time. Similar recommendation shall be made for unqualified courses. So, the real-time requirements of algorithm are higher.

(3) Accurate information. Recommendation curricula-variable basis requires accurate student details, such as basic information, historical curricula-variable record, and achievement information, and detailed information of curricula-variable courses, such as the creation time, creation site, class credits, and course type and other course properties. It should be clear that after the school began to implement new curriculum standards in 2011. In order to develop students’ comprehensive abilities, the school regulated basic credits for curricula-variable courses, that is, in terms of curricula-variable courses, it is necessary to consider the limit to the credit of curricula-variable course. The information shall be accurate so as to personalize the curriculum recommendation service.

Contextual Information-based Curricula-variable Recommendation Algorithm

N nearest neighbors of the target student can be determined through application of UCF and Pearson correlation coefficient calculation (with 5% of the number of selected matrix students as the value of TOP-N). Historical curricula-variable records of N students shall be extracted in the current recommendation semester, and shall be arranged according to the curricula-variable course cumulative number of students in the descending sequence so as to generate recommendation list TOP-K={K_1,K_2,...,K_n}. But it is clear that the accuracy of the sequence is not high, and the recommendation value is relatively low, and therefore it is necessary to add contextual information to adjust the recommendation list TOP-K.

There are 2 context filtering ways: First, items irrelevant to the context shall be filtered from the recommendation list. Second, the sequence of recommendation list TOP-K shall be adjusted according to contextual information, and the item closely related to the context shall be recommended to the user with priority. Recommendation according to curricula-variable courses requires 2 contextual features for total quantity control and TOP-K: the curricula-variable course number of students is full, curricula-variable course of the target student are filtered, and curricula-variable courses of required credits are full. The remaining curricula-variable courses to be recommended are divided into two categories: the curricula-variable course of the target student with required credits unachieved, and the curricula-variable course not for the target student and consequently with required credits unachieved. The curricula-variable course of the target student is of the same type as that of the target student shall be regarded as the student preference. The curricula-variable course of the target student with required credits unachieved shall be regarded as the student demand.
Based on students preference and demand, the sequence of TOP-K recommendation list shall be rearranged so as to generate TOP-K'={K'_1,K'_2,...,K'_k}, and recommended to target student.

```c
#define MAX_COURSE_SIZE 30
typedef struct {
  string optional course name;
  BOOL condition 1=False; // condition 1 is the curricula-variable course number of students is full
  BOOL condition 2=False; // condition 2 is curricula-variable course of the target student are filtered
  BOOL condition 3=False; // condition 3 is curricula-variable courses of required credits are full
  BOOL condition 4=False; // condition 4 is the curricula-variable course of the target student with required credits unachieved
  BOOL condition 5=False; // condition 5 is the curricula-variable course not for the target student and consequently with required credits unachieved
  int weight=1
} Sort Infor;

void Context filtering () {
  Sort Infor infor[MAX_COURSE_SIZE];
  void SearchDatabase()
  {....... //Search the database to evaluate the “infor”
  }
  int i=0;
  while (i< MAX_COURSE_SIZE) {
    if (condition 1=True || condition 2=True || condition 3=True)
      infor [i].weight=0 //evaluate the “weight”
    if (condition 4=True)
      infor [i].weight++;
    if (condition 5=True)
      infor[i].weight+=2;
    i++;
  }
  Quick Sort (infor)
  {....... //Take the “infor [i].weight” as the key to sort by quick sort arithmetic, print out “infor[i]” which weight is not 0
  }
}
```

In addition, taking into account that the recommendation algorithm shall meet real-time requirements, N nearest neighbors of the target student can be calculated before curricula-variable course is on-line on the basis of accurate data, which will greatly reduce the computation time of curricula-variable recommendation algorithm.

**Solution to the Cold Start Problem**

Collaborative filtering algorithms witnesses cold start problems. In case of curricula-variable of freshmen, there is no foundation course student result record or historical curricula-variable record, it is impossible to find similar N nearest neighbors from matrix students, and therefore it is necessary to deal with freshmen curricula-variable
recommendation. Considering the feature that freshmen have less courses in the first semester, it is proper to have the first semester curricula-variable courses of the previous enrolment of the same major rearranged in the descending order so as to generate recommendation list TOP-K={K_1,K_2,...,K_{2n}}, and then adjust TOP-K through context filtering. In filtering, it is necessary to consider whether the number of curricula-variable students is full and whether there is any conflict with students required course time. In adjusting, the following 2 contextual factors should be considered: the distribution of curricula-variable course semester and requirements of the required credits of curricula-variable courses. The curricula-variable courses with fewer semesters and courses with higher required credits shall be put forward.

Taking into account real-time of the system, it can conduct off-line arrangement for curricula-variable courses of different majors. It is proper to conduct context filtering during the student curricula-variable period, which can greatly improve the efficiency of recommendation. Different from non-freshmen curricula-variable course recommendation, matrix students choose the previous enrolment target students of the same major rather than the recommendation matrix senior students with recommendation semester records. Different composition of matrix students is mainly caused by the fact that the courses of the same training plan are different in different semesters. The more distant from the recommendation semester is, the more curricula-variable courses may change. So it is rational to have the latest students as the matrix students for recommendation.

Curricula-variable Recommendation Algorithm Implementation and Testing

Curricula-variable recommendation can serve as a function in the curricula-variable system with data coming from the database of the academic system. Not all the data in the database will be used. In terms of security, only a part of data of curricula-variable recommendation in the curricula-variable system is open, which is designed as the following view:

Student basic information: {student number, name, admission date, place of origin, major}

Foundation course: {foundation course ID, foundation course name, open time, hours, credits, major}

Curricula-variable course type information: {curricula-variable course type ID, curricula-variable course type name}

Curricula-variable course: {curricula-variable course ID, curricula-variable course name, open time, curricula-variable course type ID}

Students - Foundation course result information: {student number, foundation course ID, results}

Student - curricula-variable course result information: {student number, curricula-variable course ID, results}

Curricula-variable recommendation for students can be regarded as an attempt to predict whether the student likes the curricula-variable course. It is not necessary for the recommendation algorithm to give the predicted scores. It is simply necessary to predict whether the student likes it, that is 0 or 1, and make it a classification model based on historical data. The evaluation system established by data exploration for classification can be directly used to evaluate recommendation algorithms. It is proper to construct a confusion matrix and calculate the accuracy according to the classification results and actual marker information.

Confusion matrix, #TP and #TF respectively stand for the amount of recommendation curricula-variable course of the student and other recommendation curricula-variable courses
in the recommendation curricula-variable courses #FN and #TN. P(precision) stands for the proportion of recommendation curricula-variable course of the student in the recommendation curricula-variable course list. R (recall) stands for the proportion of recommendation curricula-variable course of the student in the recommendation list of the curricula-variable courses.

\[
P = \frac{\#TP}{\#TP + \#FP}
\]

\[
R = \frac{\#TP}{\#TP + \#FN}
\]

It is obvious that the bigger the two evaluation indexes are, the better the results will be.

Coverage refers to the proportion of the coverage in all the items which the algorithm recommended for the user [8]. A lower coverage recommendation system means it will evaluate and recommend only a limited number of items and it is difficult for it to make the user have sound experience. Coverage is defined as:

\[
Coverage = \frac{|U \cup R(u)|}{|I|}
\]

I stands for set of all curricula-variable courses, U stands for set of students.

Curricula-variable course recommendation mainly aims at target student with historical curricula-variable records. Take the curricula-variable courses of 50 enrolments of 2013 of the system engineering major of the first semester of 2014-2015 academic year as example, referring to the data of curricula-variable courses of 20 enrolments of 2011 and 2012 of the same major, the accuracy, recall and coverage were calculated and algorithm results were analyzed. In calculating, nearest neighbor student K value was taken as 10, the number of curricula-variable courses N value was taken as 10. The evaluation indexes before and after the contextual information was added:

<table>
<thead>
<tr>
<th></th>
<th>Accuracy %</th>
<th>Recall %</th>
<th>Coverage%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaborative filtering</td>
<td>25.36</td>
<td>9.87</td>
<td>41.74</td>
</tr>
<tr>
<td>Context filtering</td>
<td>32.95</td>
<td>10.98</td>
<td>40.82</td>
</tr>
</tbody>
</table>

Experimental results show that after the contextual information was added, the accuracy and recall were higher while the coverage changed little, that is, through context filtering, the recommendation results meet students’ interests and needs better, contextual information-based collaborative filtering curricula-variable course recommendation algorithm meets the requirements.

Reference


