Book Recommendation Model based on Collaborative Filtering

【Abstract】The paper has proposed five models in total on the basis of collaborative filtering. By full understanding of the questions and full use of the data provided, we optimized the model one by one, and computer program results show that the model indeed reflects the actual situation.

For question 1, two factors are taken into consideration in the evaluation of a book: the preference for book tags of the users, the grade provided by the friends of the users. Linear fitting model has been proposed to study the weight of any tag when the users are giving their scores, and the model is solved with **gradient decent** algorithm. On the basis of this, **Preference Similarity** is defined to study the similarity in choosing books in a quantitative way, and it's regarded as another weight index to study the impact of friends' score on the evaluation of one book. Then, the **influence index** is defined to reflect the overall influence of user's friends on his evaluation. Results of MATLAB program shows that the influence index of the user's friends is high enough. Ultimately, the conclusion is deduced: **the book preference and the friend's scores have an significant impact on the scores.**

For question 2, one significant fact is taken into consideration: costumers with similar preference will grade in a similar way. And the model of collaborative filtering has been built on the precondition of this hypothesis. Firstly, Pearson correlation coefficient is used to describe the similarity of two users and the prediction model of collaborative filtering based on users is built afterwards. Second, the situation that the intersection of the book sets of two users is not large enough to reflect the fact is taken into account. And Jaccard-Pearson correlation coefficient is proposed to further describe the similarity of two users, and the corresponding model is then proposed.

Subsequently, another hypothesis that similar books will get similar scores from the same user is employed to optimize the former model. PCC is again used to describe the similarity of two books, and the prediction model of collaborative filtering based on items is proposed. Solving the model with MATLAB program, 20 percent of the data in file'user_book_score.txt' is picked out to serve as the testing set, and the results show that the prediction accuracy of three models are 0.8347,0.9151 and 0.9319, respectively. Model 3 is eventually used for the prediction.

For question 3, the model is built based on two significant points: the preferred tags and the predicted scores. Taking full use of the users' historical data of reading, finding out the frequency of one certain tag appearing in the books the users once used to read, **preference index** is defined to describe quantitatively the level how the users prefer one certain kind of books. Based on the definition of preference index, the Top5 tags are selected from all the tags, and the candidate item set which contains all the possible recommended books is put forward. The elements of the candidate item set are books which has one or more Top5 tags. With the former discussion of problem 2, the predicted scores of the books in the candidate item set are calculated. Finally, the product of the overall preference index and the predicted score is regarded as the final reference of which book is to be recommended. 3 books with the top 3 final value are to be recommended to the user.

At last, objective evaluation is given on the strengths and weaknesses of all the models, and optimization proposals are provided.

【Key Words】 Collaborative filtering; Pearson correlation coefficient; Linear fitting; Preference index; Gradient decent

目录

1. Question Description

The rapid development of internet technology brings us into a time of information erosion, meaning that a host of messages are full of our lives. For the information collectors, how to find large amount of information they are interested in or they really need has become a extremely difficult thing; However, for information publishers, how to make their messages stand out has already become the necessary conditions for occupying market share, profitability. As a result, recommendation mechanism becomes a vital tool to solve the problem of information redundancy, and it's also widely used in the recommendations of searching keywords ， subjects ， e-commerce products , social network data and so forth.

Subject offers users' behavior information on a famous online bookstore, including rating data for books, label information of books and people's social relationships. According to the data required, we need to answer the following questions:

(I)We are required to analyze factors influencing books' score which users made;

(II)By establishing the model, we should forecasts scores users make for the books which they don't read in the file called 'predict.txt;

(III)For users in the file of "predict.txt", we should recommend them 3 books they don't read.

2. Question Analysis

Every individual is linked with the help of Information Ages, and universal access to computer network makes it possible to share information. Also, network recommendation mechanism provides a convenient for people to obtain the required information. Therefore, website makers are able to set recommendation projects according to keywords users used to search, concerned topics of themselves and their friends. What's more, users usually select important information by these projects.

Firstly, we need to process the data. The data provided can be used to analyze, including users - books - scores data, books - tag data, users-social data and users - books read data. And we can make a series of matrix based on these data and then analyze. For question1 , we suppose that users' scores are influenced by personal preference. Tags of some books can reflect books' types, though they are showed by ID figures. But a figure stands for a type of book. And the total number of books is certain. Therefore, we can consider users' scores weights from each tags. Second, when selecting books, users may also refer to their friends' choice. For analyzing various factors which have an influence on scores, we can use principal component analysis and multivariate linear fitting method. Here the method of principal component analysis is adopted. Here, we adopt the method for multiple linear fitting to analyze the impact of books' tags firstly.

As for question 2, we need to predict users' scores and we can use the collaborative filtering method to find the statistical information in the attachment of 'user_book_score.txt'. Nowadays, many algorithms are used to calculate the correlation coefficient between samples, including collaborative filtering algorithm based on users, the improved collaborative filtering algorithms, and collaborative filtering algorithm based on the project. The first two algorithms are based on an assumption that if two users with similar preferences, when one user evaluates a commodity, the other users shall have similar comments on the goods. When two users with the highest similarity, we can use a user to approximately estimate the books' evaluation of another user for the same books. And Pearson algorithm on the basis of projects is based on another assumption: when two projects have high similarity, evaluations of the same user should be relatively close for two projects. Therefore, the collaborative filtering based on projects can be used to estimate similar projects of the same user. In problem 2, we can use three algorithms to predict respectively, and select the part of the rating data as the test set. Finally we are able to assess the accuracy of the algorithm, using the algorithm of the highest accuracy as the final prediction algorithm.

As for question 3, when recommending books for users, we are required to consider the history reading books and find the tags of uses' favorite books. Also we need to screen the top10 books. Combing with the model of question 2, we should analyze two factors of books' tags and book predicted scores. Finally, we can recommend 3 books whose comprehensive scores are highest.

3. Basic Assumption

(I)Data given by the subject is true and reliable;

(II)Users only estimate the books they read;

(III) All evaluations are rational;

(IV)All data are selected randomly;

(V)When selecting books, users will refer to recommendation information and scores.

4. Variables

Variables	explanation
T	Matrix of book' tags;
$t_{i,j}$	The number of tag j owned by book i;
$x_{i,j}$	The weight of user i to tag j;
$r_{i,j}$	Scores user i makes for the book j;
$\hat{r}_{i,j}$	Estimated scores of book j to user i;
J_i	Mean square root values of users i;
$\overline{r}_i$	The average scores of user i;
$r^*_{i,j}$	The predicted scores user i makes for the book j;
I_{U_k}	The collection of books user U_k scores;
U_i	The collection of neighbor users for user i;
P_j	The collection of neighbor projects for project j;
$U_{i,j}$	The collection of common users scoring projects i and j;
$sim(u,v)$	Pearson correlation coefficient of user v and u;
$\cos(u,v)$	Cosine similarity between user v and u;
u_i	The collection of any user u_i;
u	The goal collection;
$C_{i,j}$	The collection of books user i and j both like;

$J(u,u_k)$	The Jaccard coefficient between u and u_k ;
$\lambda_{i,j}$	The preference factor of tag j to user i ;
$F_{u,i}$	The synthetically rating index of book i to user u ;
I_{rec}	The collection of recommended books;

5. Model

5.1 Data Processing

Firstly, the data provided has to be preprocessed. The data can be processed with JAVA programming, and the data of different tags of the same book, different books one user already read together with different one-way friends of the same user can be abstracted into a series of vectors from the data files ‘book_tag.txt’, ‘user_read_ history.txt’ together with the file ‘user_social.txt’. At the same time, the one-direction friends of the 6 users in the file ‘predict.txt’ and the books they have already read can be find out for the good of further analysis. Then, find out the books which have its tag information in ‘book_tag.txt’ from the historical data, and abstract these tags of the corresponding book into another vector. Afterwards, the large sums of data can be well-ordered, and there is some important information we get, and the detail information can be find in the attachment. There are 4071 users and 7840 books in the file ‘user_book_score.txt’, and 8369 books and 1129 tags in the file ‘book_tag.txt’.

5.2 Model of Question 1

5.2.1 Data Screening

Users grade books depends on many factors, including the degree that users like the type of books, the user's own evaluation scale, scores the user's concerned makes, and even to the public for a consensus view of books , all will affect the user's scores. Users can either depend on their true preferences for books, or referring to the concerned persons, and the public opinion.

Based on data provided by the subject information, we first make screening on these data. The data file ‘book_tag. txt’ gives tag data of some books , and this data can reflect the books types , so it has some impact on the process of user ratings; Secondly, ‘user_social. txt’ reflect the data of concerned users, so it also considered useful information; While ‘user_read_history. txt’ just reflect the data on the user's reading history, but it does not give users’ direct scores of history-reading books ,so that the data has no obvious use; ‘user_book_score. txt’ can reflect the user's rating scale, therefore, it can be considered to be useful data. Therefore, we select type of books, social relations and the rating scale as the factors affecting user ratings.

5.2.2 Model 1 Linear fitting model

One of the main factors influencing the user to select books and scores are types of books. In order to find out how the book tags influence the user’s scores, we can use the gradient descent method to calculate the proportion of each tag in user ratings, and analyze the same tag in the score of the user's focus. Here by linear fitting method and gradient descent method to calculate the minimum mean square error, the specific process as follows:

(I) Matrix Forming

We can make all books graded and their corresponding tag in the file of ‘user_book_score.txt’ into a book $m \times n$ matrix of 0-1, namely

$$T = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ t_{m,1} & t_{m,2} & \cdots & t_{m,n} \end{bmatrix}$$

Where row represents the total number of books graded, and line means the number of tags. And $t_{i,j} = 0$ or 1. When book i has tag j, $t_{i,j} = 1$. Otherwise, $t_{i,j} = 0$.

Similarly, we can get a $s \times m$ matrix of user-scores

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{m,1} & r_{m,2} & \cdots & r_{m,n} \end{bmatrix}$$

Where

$r_{i,j}$: Book j 's scores of user i ;

Notice: users all join evaluation and the matrix can be got from the subject's data.

(II)Coefficient Setting

When work out the status of book tag in the users' heart, we only need to consider the influence that the book's tag has for scores to further analyze the book' tag weights in the social relationship. Therefore, we are able to find the impact in 2 users by discussing the connection between social relationship and users' preference.

We set $x_{i,j}$, meaning that weights tag j have for user i , and form a $s \times n$ weighted matrix:

$$X = \begin{bmatrix} x_{11}\ x_{12} \cdots x_{1n} \\ x_{21}\ x_{22} \cdots x_{2n} \\ \text{.........} \\ x_{s1}\ x_{s2} \cdots x_{sn} \end{bmatrix}$$

Where row represents users, and line means tag, scoring weights is variable to be worked out.

(III)Model

Each tag has different weights for users. When there are books users are fond of in the book already read, this project tag has influence to improve users' scores to some extent. By using Linear fitting method, we can calculate the users' score weight value of each project. Gradient descent method is a kind of the most optimal iterative search algorithm using function in the negative gradient direction decreasing fastest principle. In this model, we can user the gradient descent method to solve the residual error minimum. The mathematical model of linear fitting is formed:

$$\hat{R} = XB^{T}$$

Namely
$$\begin{bmatrix} \hat{r}_{1,1} & \hat{r}_{1,2} & \cdots & \hat{r}_{1,m} \\ \hat{r}_{2,1} & \hat{r}_{2,2} & \cdots & \hat{r}_{2,m} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{r}_{s,1} & \hat{r}_{s,2} & \cdots & \hat{r}_{s,m} \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{s,1} & x_{s,2} & \cdots & x_{s,n} \end{bmatrix} \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ t_{m,1} & t_{m,2} & \cdots & t_{m,n} \end{bmatrix}^{T}$$

While

$$\hat{r}_{i,j} = \sum_{k=1}^{n} x_{i,k} t_{i,k}$$

Where

$\hat{R}$:User rating estimate matrix;

$\hat{r}_{ij}$:Rating estimation user i makes for book j ;

At this point, the mean square error root with n weight values of user i is

$$J_i = \sqrt{\sum_{j=1}^{m} (r_{i,j} - \hat{r}_{i,j})^2} = \sqrt{\sum_{j=1}^{m} (\sum_{k=1}^{n} x_{i,k} t_{i,k} - r_{i,j})^2}$$

Where

J_i =rating estimate mean square root values about user i .we give each weight $x_{i,j}$ an original value $x^{(0)}{}_{i,j}$ and work out the minimum of J_i by iterating, so

$$\nabla J_i(X_i^{(k)}) = \left[\frac{\partial J_i(X_i^{(k)})}{\partial x_{i,1}}, \quad \frac{\partial J_i(X_i^{(k)})}{\partial x_{i,2}}, \quad \cdots \quad \frac{\partial J_i(X_i^{(k)})}{\partial x_{i,n}} \right]$$

Where

$X_i^{(k)}$ = the weighted vector of user i after n iterations,

$$X_i^{(k)} = \left[x_{i1}^{(k)}, x_{i2}^{(k)}, ..., x_{in}^{(k)} \right];$$

The process of iterating:

$$X_i^{(k+1)} = X_i^{(k)} - a^{(k)} \frac{\nabla J_i(X_i^{(k)})}{\left\| \nabla J_i(X_i^{(k)} \right\|}$$

When $\left\| X_i^{(k)} - X_i^{(k+1)} \right\| \le \varepsilon$ or $\left\| \nabla J_i(X_i^{(k)}) \right\| \le \varepsilon$, the process ends. And $a^{(k)}$ means iterative step length and ε means the iterative precision.

Using the above methods, we solve the various users' weighted rating values for different tags with MATLAB programming. WE list some weight values calculated as shown below.

Book	Book Tags				
ID	8427	6391	5942	9230	7628
817168	0. 1953	0. 1024	0. 0931	0. 0834	0. 0416
616799	0. 0151	0. 2102	0. 0082	0. 1021	0. 0321
489646	0. 1032	0. 0092	0. 2103	0. 0023	0. 1055
391403	0. 2256	0. 0036	0. 1013	0. 0032	0. 0102

Table 1.The weight value of some book tags.

5.2.3 Model 2 Social Relationship Model

(I) Preference Similarity

Users' social relationship has some influence to the process of rating. For example, when concerned users grade generally high or low, it can impact the judgment users have for books directly, which origins the phenomenon of "following".

On the other hand, the fundamental reason that the user concerns is that there are the same preference between them, namely they like the same type of book.

Therefore, we are able to user the result of model 1 to analyze the influence that social relationship have to the users' scores.

We define the preference similarity between goal users u and u_k :

$$H(u,u_k)=\frac{\sum_{j=1}^{n} x_{u,j}\cdot x_{u_k,j}}{\sqrt{\sum_{j=1}^{n} {x_{u,j}}^2}\cdot\sqrt{\sum_{j=1}^{n} {x_{u_k,j}}^2}}$$

Where $x_{u,j}$ is user u 'scores weight of tag j ;

(II)Social Relationship Model

According to model 1, we know that $X_u=\lfloor x_{u,1},\ x_{u,2},\ \cdots\ x_{u,n}\rfloor$ means the weighted vector of goal user u to each tag, while concerned users' weighted vector is $X_{u_k}=\lfloor x_{u_k,1},\ x_{u_k,2},\ \cdots\ x_{u_k,n}\rfloor$ .we can build the model:

$$Inf(u,u_k)=\frac{1}{5}\left|\frac{\sum_{i=1,i\neq p}^{n} x_{u,i}\overline{r_{u,i}}}{\sum_{i=1,i\neq p}^{n} x_{u_k,i}\overline{r_{u_k,i}}}\cdot\frac{\overline{r_{u_k,p}}}{\overline{r_{u\ ,p}}}\right|$$

Where

$\inf(u,u_k)$: Influence coefficient of user u to user u_k ;

$\overline{r_{u,i}}$ Weighted average values of book whose tag is i to user u ;

$\overline{r_{u,p}}$ The highest weight between and closest to the average rating scores between user u and u_k ;

The meaning of this model is that it can first rule out the evaluations of the books of the two individual's preference closest to the label and then consider the ratio of rating scale, which stands for other rating standards of two people. And the significance of multiplying with the right fraction is to calculate the average score goal users should give first, and then divide actual average points. Last, multiplying one fifth is to ensure that impact factor is between [0, 1].

Through screening the appropriate test set, and using MATLAB to prove the validity of this model, we find that the concerned users can influence goal users' scores to a certain degree. Therefore, social relations and books tags are the key factors influencing the user ratings.

5.3 Model of Question 2

5.3.1 Collaborative Filtering

Combined with the actual situation, when the two users u and v have the relatively similar scores for books collections C and user u has a certain score for another books' collection C_1 or another books' element $i \notin C$, therefore the user v also has similar grades for set C_1 or elements i . Collaborative filtering is based on the premise of this assumption, and put forward the measure of the similarity between two individuals. Collaborative filtering is a very efficient algorithm for business recommend. By mining similarity between two individuals, it can do further prediction and recommendation. Commonly used collaborative filtering algorithm in computers includes cosine similarity algorithm and Pearson correlation coefficient methods.

We assume scoring vector of users is $u \notin R^m$, and another's is $v \notin R^n$ $(m \neq n)$. In order to analyze the similarity of 2 users' in the same vector space, we make use of $v, u \notin R^{\min(m,n)}$, which means that we can map these two into the vector space which includes the common elements' value of these 2 vectors.

The cosine similarity of 2 scoring vectors:

$$\cos(u,v) = \frac{\sum_{i \in I_u \cap I_v} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i \in I_u \cap I_v} {r^2}_{u,i}} \cdot \sqrt{\sum_{i \in I_u \cap I_v} {r^2}_{v,i}}} \quad (1)$$

Where

$r_{u,i}$: Scores user u makes for product i ;

$r_{v,i}$: Scores user v makes for product i ;

However, Pearson correlation coefficient considers the relative problems of personal scoring rules, so it's better than the method of cosine similarity. The Pearson correlation coefficient between user u and v :

$$sim(u,v)=\frac{\sum_{i\in I_u\cap I_v}(r_{u,i}-\overline{r_u})\cdot(r_{v,i}-\overline{r_v})}{\sqrt{\sum_{i\in I_u\cap I_v}(r_{u,i}-\overline{r_u})^2}\cdot\sqrt{\sum_{i\in I_u\cap I_v}(r_{v,i}-\overline{r_v})^2}} \tag{2}$$

Where

$sim(u,v)$: The similarity between users u and v , and $sim(u,v)\in[-1,1]$;

I_u : The book scored by user u ;

I_v : The book scored by user v ;

$\overline{r_u}$: The average scores of user u ;

$\overline{r_v}$: The average scores of user v ;

$r_{u,i}$: Scores user u makes for the book i ;

$r_{v,i}$: Scores user v makes for the book i ;

From the Pearson correlation coefficient, we are able to conclude that different individuals due to different scoring rules can affect the similarity, because some tend to score highly to each book, while the others don't; As a result, the way can eliminate the influence to the similarity between 2 users.

5.3.2 Prediction Model based on Pearson Correlation Coefficient

As for problems for predicting users' scores, we can consider the basic assumptions of Collaborative filtering, which means 2users or more with the same reading tendency have similar evacuation for different stuffs. Collaborative filtering method (KNN) based on users can be used to offer recommend and scoring prediction to products for various goal users.

Predicted model based on Pearson correlation coefficient:

(I)The forming of neighbor users:

$$U=\{u_1,u_2,u_3\cdots\cdots u_k\} \tag{4}$$

$$sim(u,u_1)=\max\{sim(u,u_1)\} \qquad 0<i<N \tag{5}$$

$$sim(u,u_1)>sim(u,u_2)>\cdots>sim(u,u_k) \tag{6}$$

Where

u_i : User i ;

$sim(u,u_i)$: The similarity between users u and u_i ;

U : The neighbor users of goal user u ;

N: The number of all users;

We set the first k users with the highest Pearson correlation coefficient as the neighbor users, and order them by their values of similarity from the highest to the lowest, which can also prepare for the latter predication.

(II) Goal Scoring Predication

To score to the goal projects, we are able to do it by the prediction method of the neighbor users' weights. We assume that in the neighbor users U of goal users u , we can find the collection of neighbor users scoring the goal project i .

$$U^*{}_i = \{u_1, u_2, u_3 \cdots u_k\} \qquad (7)$$

Goal scoring predicted model of user i making for project i

$$r_{u,j^*} = \overline{r_u} + \frac{\sum_{u_k \in U^*{}_i} sim(u,u_k) \cdot (r_{u_k,i} - \overline{r}_{u_k})}{\sum_{u_k \in U_i^*} (|sim(u,u_k)|)} \qquad (8)$$

Where

$\overline{r_u}$: The average scores of user u ;

$r_{u_k\, i}$: Scores user u_k makes for the book i in the collection of neighbor users $U^*{}_i$;

$\overline{r_{u_k}}$: Average scores of books scored both by users u_k and u ;

$|\,sim(u,u_k)\,|$: Correlation coefficient' absolute values between 2 users;

Also

$$\overline{r_{u_k}} = \frac{1}{N_{I_{u_k}}} \sum_{I_u \cap I_{u_k} \neq \Phi} r_{u_k,j} \qquad \textit{(9)}$$

$$\overline{r_u} = \frac{1}{N_{I_u}} \sum_{j \in I_u} r_{u,j} \qquad \textit{(10)}$$

Where

$r_{u,j}$: Scores goal user makes for book j ;

I_u : The collection of books scored by user u ;

N_{I_u} : The number of elements of collection I_u ;

$I_u \cap I_{u_k} \neq \Phi$ means that when users u and u_k both score some book, we need to work out the average of books scored by user u_k.

Explanation for this model:

a）Firstly, we need to screen out the users scoring book i from the neighbor users of goal user u. Because only when the neighbor users of goal user read and score this book, the scores' branch is significant for scores made by user u.

b）For formula (8), denominator means that the absolute sum of Pearson correlation coefficient between all neighbor and goal users screened out, and numerator represents we should subtract the scores for book i made by neighbor users and existing. And then we can use the weighted sum of these differences as numerator. The purpose of taking difference is that we can consider the scoring scale difference of different users, while by taking difference it can impact whether the book is good in his scoring scale. And the aim to take weighted sum is that the weighted factor-- $sim(u,u_k)$ can reflect the similarity of the reading preference. Meanwhile, users who are more similar to the goal user have higher weight.

c）$\overline{r_u}$ is also set based on the scoring scale of goal user. It is got by working out the average scores made by user u and acts as the scoring scale of user u.

d）As for formula (9), when getting the average scores $\overline{r_{u_k}}$ of user u_k, we can only work out the average ones of books read by both users u_k and u, because the common books can reflect the same preference of these 2 users.

（III） The result of the model

a）The result of the correlation coefficient

We can put the attachment "user_book_socre.txt" into software of MATLAB, and we can use u to represent the graded vector read by user.

$$u = (r_{u_1}, r_{u_2}, r_{u_3} \cdots r_{u_i} \cdots r_{u_n}) \tag{3}$$

We can work out the Pearson correlation coefficient between each user and other users by method of Pearson correlation coefficient, and then we can get a correlation coefficient matrix（see attachment 1）.

We can take the users in the attachment of 'predict.txt' as goal users, and the ID of books as predicted objects. By MATLAB, neighbor users are filtered. In order to avoid the disorder of data, we can choose coefficient $k = 200$, meaning that electing the first 50 users who are more similar to the goal user as the neighbor users.

b) Score prediction

According to the formula (10) to edict MATLAB procedure, the specific calculating flow is as follow figure 1-1:

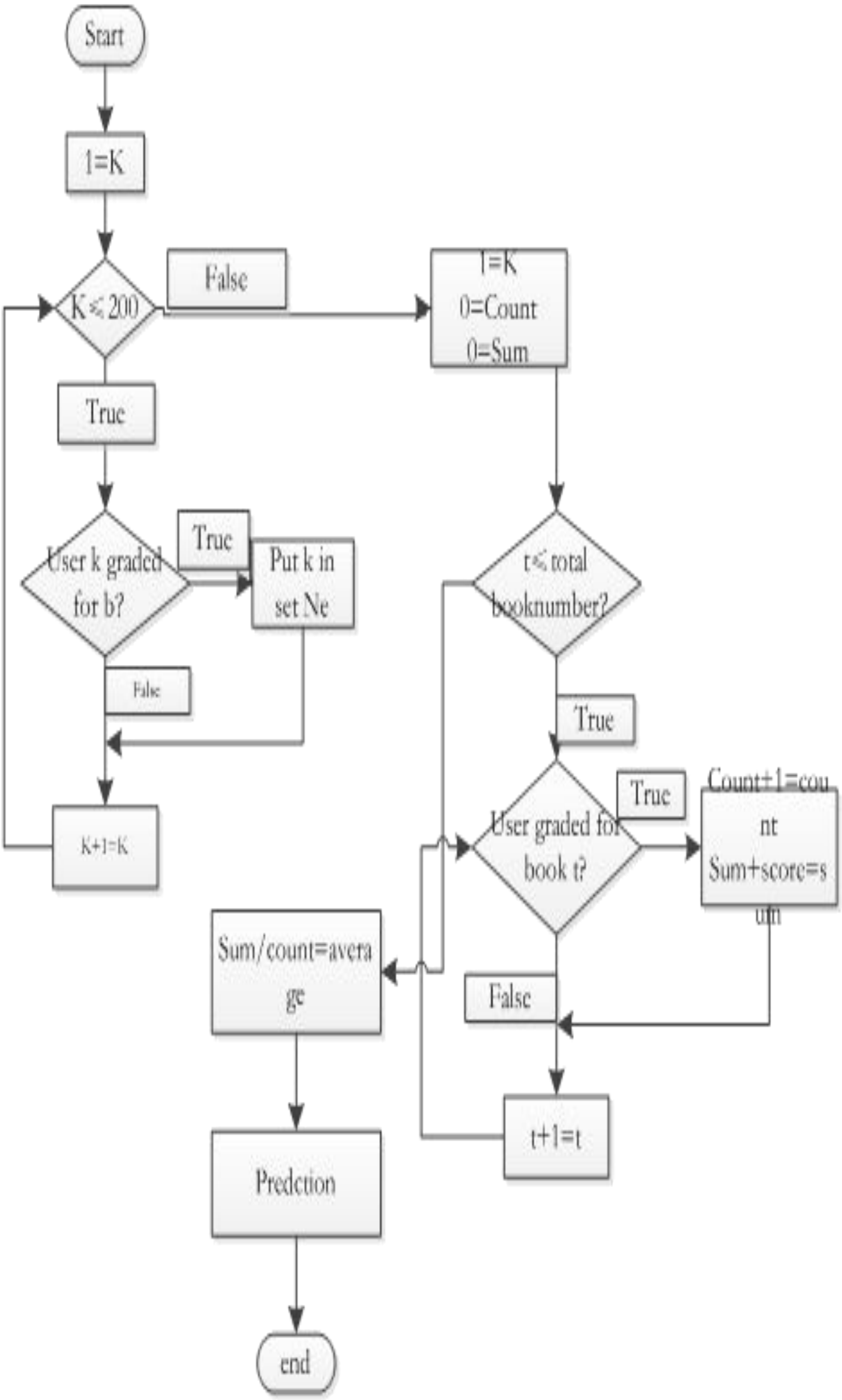

Figure 1.The calculating flow of model 1

In order to prove the validity of the model, we can take the data in the file of 'user_book_score.txt' as test set. And we get the curve of error as shown bellow.

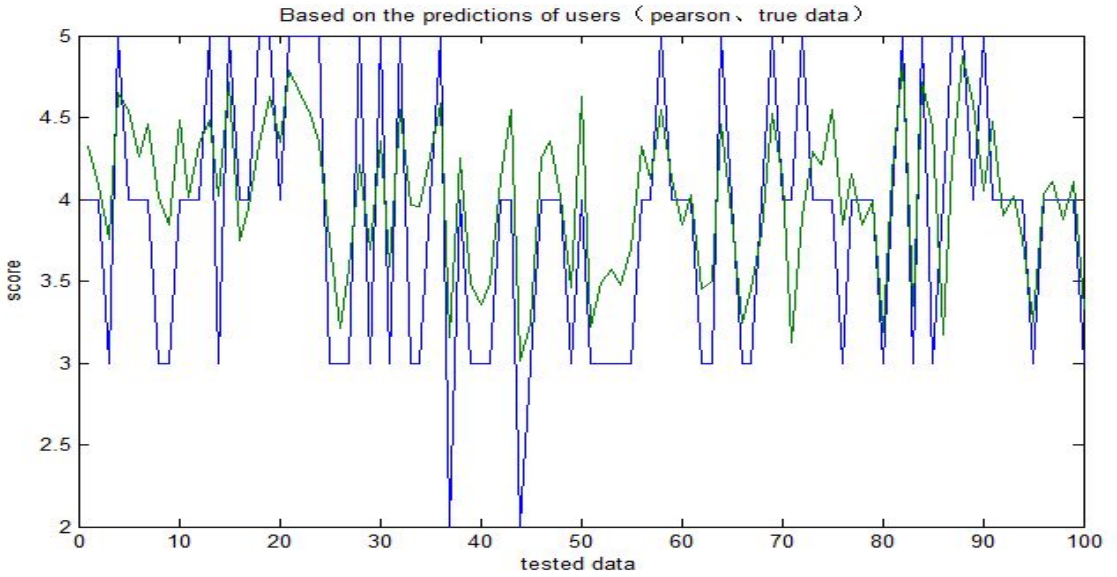

Figure 2.predicted value with PCC model and the true value

From the above image, we can directly see that there is a certain error between predicted and actual values in process. Through the analysis of this prediction model, when calculating the correlation coefficient between two users, we only consider the two users for the same point approximation degree of books. And in some extent it can cause larger error, because it may occur that two people only read a few books at the same time and have the similar scores. In this case, by using this model it can lead that the correlation coefficient between two people is near to 1, which will produce certain effect to the whole forecast . So, we put forward prediction model based on the following Jaccard - Pearson correlation coefficient (JacPCC) algorithm.

5.3.3 Model 2 JacPCC Prediction Model

(I) Jaccard - Pearson correlation coefficient

Jaccard - Pearson correlation coefficient (JacPCC) based on Jaccard correlation coefficient is an improved correlation algorithm. Pearson correlation coefficient algorithm is based on common objects of two sets, and it only considers two individuals with the evaluation of things with the same property, and ignores the two spaces' dimension, namely the 2 individual concern users respectively. Significant downside of this kind of algorithm is the misjudgment for some special cases, namely when the intersection between two individuals is very small, there may be two individuals that don’t have strong correlation, but the Pearson correlation coefficient is always close to1. Therefore, improved Pearson correlation coefficient, namely Jaccard - Pearson correlation coefficient, considers the two overall evaluations in the influence of the correlation coefficient between the two, by Jaccard - Pearson coefficient to reflect the impact on the intersection as a whole.

Jaccard coefficient:

$$J(u,u_k)=\frac{|I_u \cap I_{u_k}|}{|I_u \cup I_{u_k}|}$$

Where

I_{u_k} : Rating collection of user k ;

$|I_u \cap I_{u_k}|$: The number of intersection of I_u and I_{u_k} ;

$|I_u \cup I_{u_k}|$: The number of union of I_u and I_{u_k} ;

Formula represents the weight of the intersection of goal set and user k among the books graded by these 2 users. We can use Jaccard coefficient to multiply Pearson coefficient to get the Jaccard - Pearson correlation coefficient:

$$sim(u,u_k)_J=\frac{|I_u \cap I_{u_k}|}{|I_u \cup I_{u_k}|}\frac{\sum\limits_{i\in I_u \cap I_v}(r_{u,i}-\overline{r_u})\cdot(r_{u_k,i}-\overline{r_{u_k}})}{\sqrt{\sum\limits_{i\in I_u \cap I_{u_k}}(r_{u,i}-\overline{r_u})^2}\sqrt{\sum\limits_{i\in I_u \cap I_{u_k}}(r_{u_k,i}-\overline{r_{u_k}})^2}}$$

So, when the intersection between the two individuals is small, the correlation coefficient between the users set k to the goal set is relatively small. As a result, prediction of impact on the goal user is relatively small.

(II) Improved prediction Model

We can build new model on the basis of JacPCC model. The predicted value of goal user to book i :

$$r_{u,j^*}=\overline{r_u}+\frac{\sum\limits_{u_k\in U^*{}_i} sim(u,u_k)_J\cdot(r_{u_k,i}-\bar{r}_{u_k})}{\sum\limits_{u_k\in U_i^*}(|sim(u,u_k)_J|)}$$

Where

$sim(u,u_k)_J$: Jaccard - Pearson correlation coefficient between goal user u and u_k .

(III)Result of the Model

We need to compile the MATLAB procedure to calculate the Jaccard – Pearson correlation coefficient first. Basing on the above improved the prediction model of MATLAB and using the model of testing method for testing the same test set again, we need to compare error curve in these two kinds of models shown in the folliowing figure.

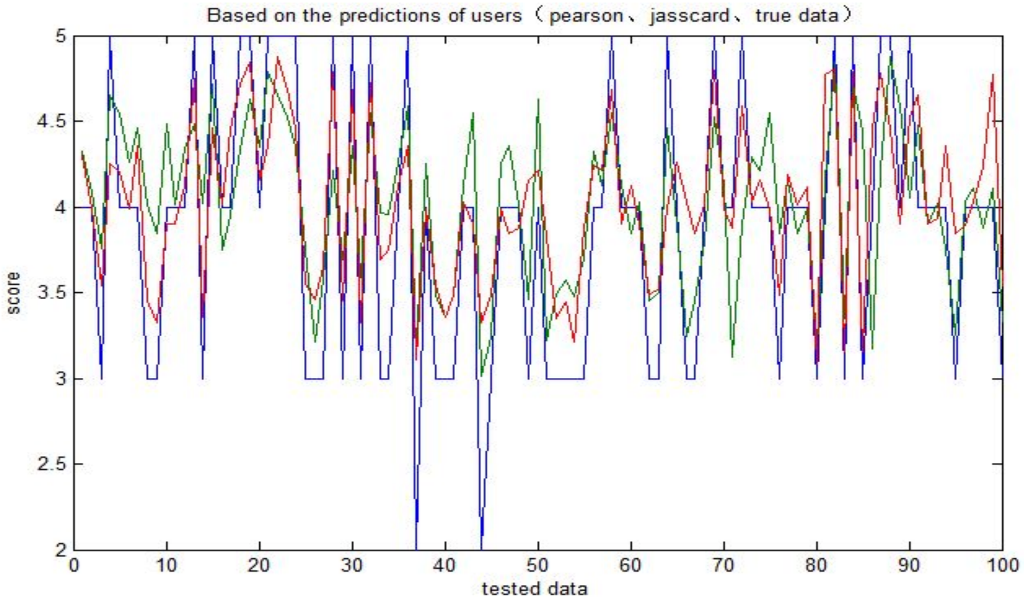

Figure 3.The predicted value of PCC model and JacPCC model

Compared to two groups of graphics, it can be seen that using the improved model for predicting the test set data has the higher precision. Therefore, we can use the improved model to predict the books' scores user grade for the file of 'predict.txt'. The data of prediction is shown bellow.

USER	ORDER OF THE BOOKS(The same as the data file)					
ID	1	2	3	4	5	6
7245481	4.1078	4.1918	4.3630	4.1985	4.0263	4.1938
7625225	3.5010	3.8133	3.7479	3.9250	3.7723	3.9030
4156658	4.1901	4.1763	3.9922	4.2538	4.0076	4.1294
5997834	4.1540	4.5587	4.2154	4.3694	4.3966	4.0877
9214078	4.3938	4.3729	4.2099	4.2088	4.2094	4.2362
2515537	3.8842	3.7179	3.9764	3.9427	3.7618	3.5544

Table 2.The predicted value of JacPCC Model

5.3.4 Model 3 Prediction Model based on Items

The above two models are based on the user similarity model. They calculate 2 users' correlation coefficient in the common projects' collection. In fact, a collection of intersection between two users may only have a few elements, which will lead to rating matrix with high sparsity.

We define the sparsity of a collection as $S_c = 1 - \frac{N_r}{N_u \cdot N_I}$,

Where

N_r : The total scores;

N_u :The number of total users;

N_I :The number of total books.

Obviously, when each user makes a comment on all the books, we can get $S_C = 0$ .With computer program, we are able to calculate the sparsity of the file called 'predict.txt' which is as high as 0.9232, explaining the data's content is highly sparse. Based on the relativity analysis of item problems and the principle of correlation analysis based on the user, we just turn the research of preference similarity between two users into a certain degree of similarity between two items. We, below, establish the prediction model.

(I)Forming the set of neighbor projects

The Pearson correlation coefficient between any book i and j :

$$sim(i,j) = \frac{\sum_{u \in U_{i,j}} (r_{u,i} - \overline{r_i}) \cdot (r_{u,i} - \overline{r_j})}{\sqrt{\sum_{u \in U_{i,j}} (r_{u,i} - \overline{r_i})^2} \sqrt{\sum_{u \in U_{i,j}} (r_{u,i} - \overline{r_j})^2}}$$

Where

$U_{i,j}$: The common collection of rating projects between project i and j ;

$\overline{r_i}$: The average scores of project i ;

$\overline{r_j}$: The average scores of project j .

Similarly, we can get the neighbor collection of project i $P_i = [I_1, I_2, \cdots, I_k]$, in which $I_1, I_2, \cdots, I_k$ are the top k neighbor projects which are closest to project I .

(II)Rating prediction model based on the item

After Generating neighbors projects, we are able to predict scores of item i. Prediction model is as below:

$$r_{u,i}^{*} = \overline{r_i} + \frac{\sum_{j \in P_i} sim(i,j) \cdot (r_{u,j} - \overline{r_j})}{\sum_{j \in P_i} |sim(i,j)|}$$

Where

$r_{u,i}^{*}$: Predicted score of project i to user u ;

P_i : The collection of neighbor users of project i ;

Formula represents the scores which we get by using a weighted average of the neighbor users set to approximately estimate goal project i .

(III)Result of the model

Simulating the methods for model 1 and 2, we write the MATLAB program to solve the model. Finally we can get the forecast of each user to review books. To compare the accuracy prediction based on the user model and item model, we can choose two the same test sample of model 2 and comparing prediction based on JacPCC model shown bellow:

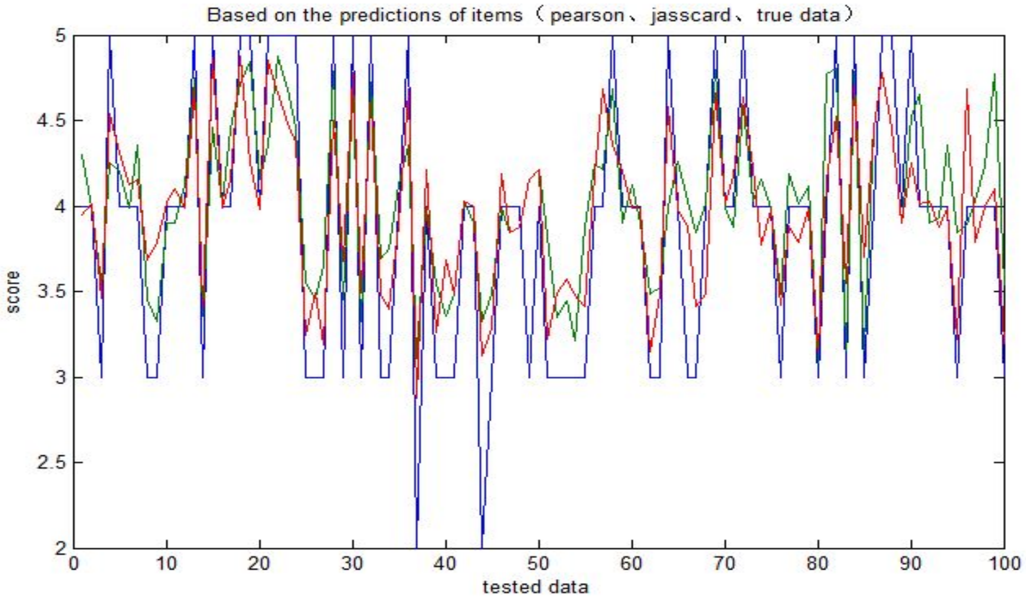

Figure 4.The Predicted value with PCC based on items and value with JacPCC based on user

Comparing the predicted situation of the 2 models, we find that the accuracy of the predicted model based on items is better than that of one based on JacPCC, while the accuracy of JacPCC model is better than that of other users. Obviously, model based on items is the best.

According to the model, we are able to estimate scores of items waiting to be graded and then we can get the predicted value shown as attachment.

5.3 Model 3 Recommendation Model

According to the discussion of question 2, as for question 3, we mainly consider the favorite books' types and the predicted rating values. When users' preference doesn't change too much, the favorite books of recommended users are relatively suitable. Meanwhile, we need to synthetically consider the predicted rating values of some books graded by the user. The predicted rating value is higher, the favorite degree is better.

(I) $Top5$ Tags

The subject gives the history reading sources of users, including ones of 6 users needing to be recommended. By screening the data, we discover that the most history reading books' tags are able to be found in the file of "book_tag.txt". So it's possible to find the users' preference information from the history data.

We set H_i represents the collection of history books of user i, the element of this collection is the books user i read and $i \le 6$. We can structure a tag matrix combining with all history books of user i and tags.

$$T = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ t_{m,1} & t_{m,2} & \cdots & t_{m,\mathrm{n}} \end{bmatrix}$$

Where

m : The total number of history books of user i;
n : The number of all tags;

$$t_{i,j} = \begin{cases} 0 & tag_j \in Tag_{i,m} \\ 1 & tag_j \notin Tag_{i,m} \end{cases}$$

Where

tag_j : The tag j;

$Tag_{i,m}$: The tag collection of book m read by user i;

Now, we are able to give the definition of preference factors. The preference factors of tag tag_j to user i:

$$\lambda_{i,j} = \frac{\sum_{r=1}^{m_i} t_{r,j}}{\sum_{j=1}^{m_i}\sum_{r=1}^{m_i} t_{r,j}}$$

Where

$\lambda_{i,j}$: The tag j ’ preference to user i;

m_i: The total number of books read by user i;

The meaning of the formula is: by researching the ratio of total numbers of some tag in the history books and the total number of all tags we are able to quantitatively describe the preference to some tags of users. So we can get $\sum^{n} \lambda_{i,j} = 1$.

Calculate the 6 users’ preference factors with MATLAB program，we select the $Top5$ tag as candidates to prepare for the next recommendation. The six users’ preferences’ for $Top5$ tags is shown in the following table.

USER	TOP5 TAG ID/WEIGHT				
ID	1	2	3	4	5
7245481	6067/0.0317	6391/0.0263	3924/0.0175	6449/0.0171	7336/0.0136
7625225	6391/0.0237	6067/0.0191	7515/0.0186	2099/0.0186	5380/0.0169
4156658	6067/0.0253	6391/0.0252	3924/0.0175	6449/0.0161	9230/0.0156
5997834	6067/0.0264	6391/0.0264	6449/0.0215	4528/0.0182	5380/0.0149
9214078	6391/0.0248	3924/0.0232	9230/0.0205	7336/0.0192	6067/0.0167
2515537	3924/0.0295	7736/0.0259	6067/0.0222	5896/0.0197	6391/0.0192

able 3.The top5 tag ID and its corresponding book ID

(II)The recommendation model based on reading history

With the discussion in question (1), we are able to find out t user i ’ $Top5$ labels, but there still exists a lot of books with these tags, and even favorite types of books also will not necessarily make users satisfied. As a result, we can consider score prediction model combining with model 2 to solve books recommended problems.

By analyzing problem 2, we find that the accuracy of predicted model based on Jaccard - Pearson correlation coefficient is better than that of Jaccard - Pearson correlation coefficient, while the accuracy of JacPCC model is better than that of other users. As a result, when recommending to users in the file of “predict.txt”, we can use the value got by the predicted model based on the projects in problem 2 and model 3 as the recommended reference.

Because we need to consider the influence of books’ tags to recommend books, so we can use nooks in the file of ‘book_tag. txt’ as a recommended collection. For user I, we first need to screen out one or more books whose tags are $Top5$. Then, we set that Tag_i^5 means that the collections of $Top5$ tags, and B_d means the books already read. Also B_c means the collection of screened books, and Tag_j stands for the tag collection of book j, B is the collection of all books.

So, we have $B_c \subset B, B_c \cap B_d = \Phi, Tag_j \cap Tag_i^5 \neq \Phi$

After screening the books of $Top5$ tags, we build the recommendation model based on history.

a) Prediction of the Targeted Set based on Project

Following the process of model 3 in problem 2, the neighbor set of the project has to be deduced when predicting the scores of the chosen set of books. The correlation coefficient of book i and book j is as follows :

$$sim(i,j) = \frac{\sum_{u \in U_{i,j}} (r_{u,i} - \overline{r_i}) \cdot (r_{u,i} - \overline{r_j})}{\sqrt{\sum_{u \in U_{i,j}} (r_{u,i} - \overline{r_i})^2} \sqrt{\sum_{u \in U_{i,j}} (r_{u,i} - \overline{r_j})^2}}$$

Its neighbor set is $P_i = \left[I_1, I_2, ...I_k \right]$, $I_1, I_2, ...I_k$ denote the top k neighbor project closest to the target project, and the predicted value of the target project is

$$r_{u,i}^* = \overline{r_i} + \frac{\sum_{j \in P_i} sim(i,j) \cdot (r_{u,j} - \overline{r_j})}{\sum_{j \in P_i} | sim(i,j) |}$$

b) Recommendation Model

When taking the data of reading history into consideration, define the following overall evaluation index

$$F_{u,i} = \lambda_{\Sigma}^{i} r_{u,i}$$

$$\lambda_{\Sigma}^{i} = \sum_{j=1}^{k} \lambda_{u,j}$$

Among them, λ_{Σ}^{i} means the sum of all the preference index of the $Top5$ tag book i contains regarding the relevant user. k denotes the number of $Top5$ tag book i contains, as one book may contain more than one tag, and when some of these tags are the Top5 tags the user are interested in, the book will be more attractive to user, and the probability for the book to get a higher overall evaluation. The reason of taking the prediction value of book i into account is the consideration of the quality of the book itself, as even if the theme of the book is what the user is interested in, its quality may be disappointing due to low level of the author.

The process of screening is shown as follows:

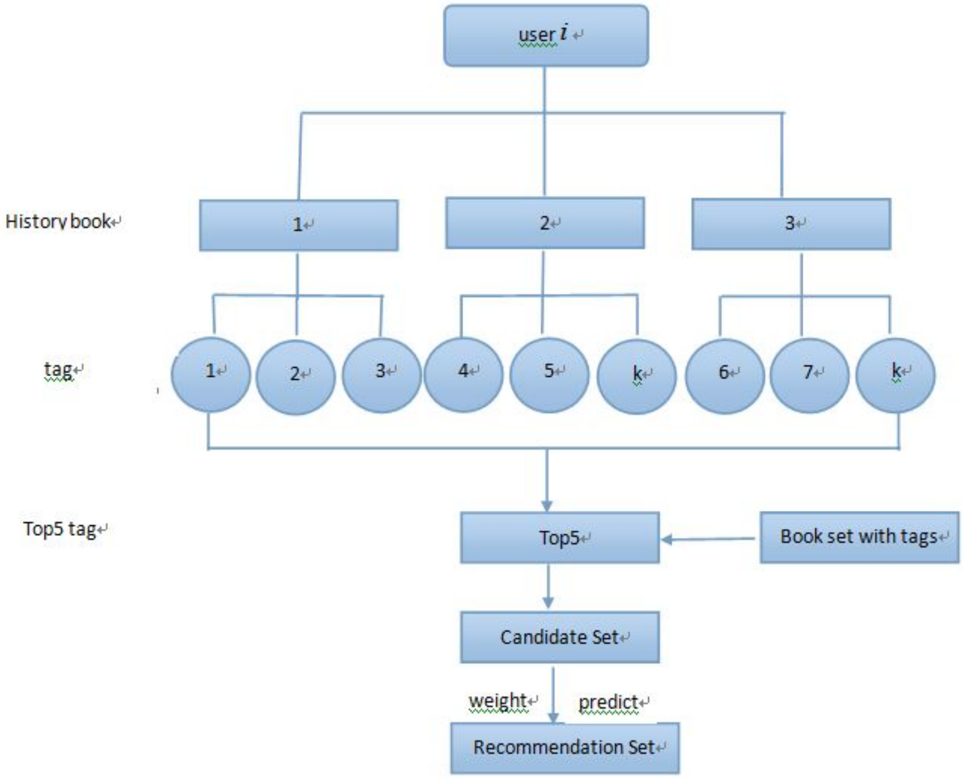

Figure5. The steps of recommendation

According to the above definition, the overall evaluation index of the elements in set B_c are calculated with MATLAB programming, and 3 books with the top 3 value are selected to be the recommended books. The recommended books of the 6 users are shown below.

6. Assessment of the model

6.1 Question 1

6.1.1 Strengths

Two models are proposed for problem 1. One model is the linear fitting model, studying the impact of book tags on the customer's appraise, deducing the minimum value of the root-mean-square error with gradient descent algorithm, and ultimately get the weighs of all the tags regarding the relevant user. The model has a appropriate consideration, together with an excellent algorithm and better theoretical background. The other model take the social collection of the user into account, and it calculated the similarity of the interests of two users with a unidirectional relationship by giving the definition of preference similarity. The model takes full advantage of the data provided, and has a full consideration with the support of quantified data, and it's also original proposed.

6.1.2 Weaknesses

Too many variations are concluded in model 1, and the time complexity is too high, which in turn causes the process of calculation of the model is deeply influenced by the performance of the computer. And the initial condition sensitiveness problem causes the misconvergence of the iteration process.

6.2 Question 2

6.2.1 Strengths

Three models are proposed with collaborative filtering which are widely used in recommendation systems. Model 1 calculated the correlation of two users with Pearson correlation coefficient, on the basis of which PCC prediction model based on users is proposed. Model 2 is proposed on the basis of model 1, taking use of Jaccard-Pearson correlation coefficient, and make up for the weakness of the potential error caused by the neglect of the size of the intersection of two users in model 1. And model 3 make a further evolution on model 2, turning to the model of prediction based on items rather than users. Three models all have an excellent theoretical background and all are well practiced. In addition, the models are optimized one by one, making the prediction precision higher, and they all have a good generation.

6.2.2 Weakness

Model 1 and model 2 are prediction models based on the users, which only consider the intersection of two users and neglect the situation of small intersection, and false prediction may occur frequently. Model 3 is proposed on the basis of project. Although it has made some improvement on the precision of prediction, it ignored the reading history of the users and the influence of the preference for the tags of the users on the evaluation of a book, and there still exists error on the prediction.

6.3 Question 3

6.3.1 Strengths

The model makes a direct analysis on the preference for the tags of the users by taking the reading history of the readers into consideration, and measure the level of preference with the definition of preference index, taking out the set of books which have one or more Top10 tags. Combining the prediction model proposed in model 2, the predicted values are putting forward. The product of the preference index and the predicted value will be regarded as the final evaluation of the book. This model has taken full consideration of the reading history of the users and it has a good originality. And the outcome of the model shows that it has the highest prediction accuracy among all.

6.3.2 Weaknesses

The result still shows some error, yet the recommendation is not perfect. More factors have to be taken into consideration and more experiences remain to be conducted.

Reference

[1]Huifeng Sun. Individual Web recommendation based on collaborative filtering [D].Beijing: Beijing University of Posts and Telecommunication,2012:
42-46.

[2]Fei Cheng. Study of collaborative filtering recommendation algorithm based on user similarity [D].Beijing: Beijing University of Posts and Telecommunication, 2012:19-27.

[3]Qingwen Liu. Study of recommendation algorithm based on collaborative filtering [D].Beijing: University of Science and Technology of China, 2013:46-
51.

[4]Huixuan Gao. Applied Multivariate Statistical Analysis [M]. Beijing: Peking University Press, 2007:249-258.

[5]Ailin Deng, Yangyong Zhu. Collaborative filtering algorithm based on project evaluation prediction [J]. Journal of Software,2003, 14(09):2-4.

[6]Chuangguang Huang, Jian Yin, Jing Wang, Yubao Wang, Jiahai Wang. Collaborative filtering recommendation algorithm based on uncertain neighbors [J]. Journal of computer science, 2010, 33(8):2-5.

[7]Goldberg D ，Nichol D ，Oki B ，Terry D. Using collaborative filtering to weave an information tapestry. Communications of the ACM ，19921 ，35(12):54-67.

[8]Breese J ，Hecherman D ，Kadie C. Empirical analysis of the predictive algorithms for collaborative filtering. Processing of the 14th Conference on Uncertainty in Artifical Intelligence. 1998:29-47.

[9]Sarwar ，Karypis G ，Konstan J ，Riedl J. Item-based collaborative filtering recommendation algorithms. Proceeding of the 10th International World Wide Web Conference. 2001. 279-284.

[10]Hu Wu, Yongji Wang, Zhe Wang. Two stage co-clustering collaborative filtering algorithm [J]. Journal of software, 2010, 21(5):139-181.

[11]Xiaobo Zeng, Zukuan Wei, Zaihong Jin. Study of matrix sparsity problem in collaborative filtering [J].Journal of computer science, 2010, (004):123-134.

[12]Chun Li, Zhenming Zhu, Xiaofang Gao. Collaborative filtering recommendation algorithm based on neighboring decision [J]. Computer Engineering, 2010, 36(13):27.

[13]Billsus D ,Pazzani M J. Learning collaborative information filters. Proceedings of the fifteenth international conference on machine learning ，volume 54 ，1998. 54.

[14]Pearson K.LIII. On lines and planes of the closest fit to systems of points in space. The London ,Edinburgh ,and Dublin Philosophical Magazine and Journal of Science ，

1901，2（11）:421-435.

[15]Frank R.Giordano，William P.Fox，Steven B.Horton. Mathematical Modeling [M].Beijing: China machine press, 203-245.

[16]Jinwu Zhuo, Yongsheng Wei, Jian Qin. Application of MATLAB in Mathematical Modeling [M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 154-167.