PROCEEDINGS OF SCIENCE

Collaborative Filtering Recommendation System based on User Potential Associated Information

Huali Shen
Anhui University of Science and Technology
Huainan, 232000, China
E-mail: 759896846@qq.com

Kelei Sun
Anhui University of Science and Technology
Huainan, 232000, China
*E-mail:*klsun@aust.edu.cn

In order to solve data sparsity and inaccurate recommended item lists in traditional collaborative filtering algorithms, a collaborative filtering recommendation algorithm is proposed in this paper on the basis of information related to user potential. By applying user attributes data to the scoring matrix, the algorithm integrates the project clustering method and the time sign of user scoring into the project recommendation process, reducing the sparseness of scoring matrix and improving the recommendation accuracy to a certain extent. For illustration, this paper also uses samples of movie recommendations to test the feasibility of the algorithm. Empirical results show that the proposed methods can effectively improve the overall performance of the proposed system, provided that the data related to user attributes, the time stamp and the project clustering method are properly applied. Therefore, this algorithm system provides an effective solution to the problems of data sparsity and low accuracy in collaborative filtering algorithms.

ISCC2017
16-17 December 2017
Guangzhou, China

1. Introduction

With the continuous progress of modern science and technology, devices have become more and more intelligent, generating explosive amount of information. Recommendation system[1][2] mainly filters out a lot of useless information on the network, so as to provide users with tailored recommendations. However, since the current recommendation system contains a large number of users and projects which continue to expand, making a scoring matrix becomes more and more sparse. Along with the cold-starting problem[3] of new users, they will lead to a significant reduction in the accuracy of the recommended items.

For the problems above, researchers have made attempts from different angles. In the literature[4], a singular value decomposition technique is proposed to reduce the dimension of the proposed system database, which indirectly reduces the dimension of the input matrix, reduces the sparseness of the data, and reduces the dimension of the matrix, increasing the complexity of the algorithm. Yifan Wu et al.[5] proposed to incorporate users' background information into the calculation of similar users to improve the accuracy of the users' pre-score. Yan Yang[6] proposed the k-means method of clustering the project, and then select the target project with the most similar to a number of projects recommended to effectively alleviate the user project scoring matrix sparseness.

2. Collaborative Filtering Recommendation System based on User Potential Associated Information

This section will be a detailed description of the methods in this article, and these methods are combined for the implementation of the algorithm to form a whole ideological framework.

2.1 Fill the Sparse Matrix with User Association Information

This paper will use the three important types of user potential information, gender, age, and occupation, to find similar users and fill in the score matrix.

The next step is to use the user's attribute characteristics to calculate the similarity between users. First, find the user ***i*** and user ***j*** in the gender similarity, the result is represented by ***S(i , j)***. the calculation is as follows: ***if*** $S_i \neq S_j$ ***then*** $S(i\ ,\ j)=0;$ ***else*** $S_i=S_j$ ***then*** $S(i\ ,\ j)=1.$, where S_i represents the gender of user ***i***, and S_j represents the gender of user ***j***.

Second, find the user ***i*** and the user ***j*** in the age of similarity, the results are represented by ***A(i , j)***. After several experiments, the user age difference within 10 years within the definition of similarity for **1** is more appropriate for the user ***i*** and ***j*** in the age of similarity. The calculation is as follows: ***if*** $A_i\text{-}A_j \leq 10$ ***then*** $A(i\ ,\ j)=1$ ***else*** $A_i\text{-}A_j>10$ ***then*** $A(i\ ,\ j)=\frac{5}{|A_i-A_j|}$, where A_i represents the age of user ***i***, and A_j represents the age of user ***j***.

Finally, for the user ***i*** and user ***j*** in the professional similarity, the results are represented by ***P(i , j)***. The calculation is as follows: ***if*** $P_i \neq P_j$ ***then*** $P(i,j)=0$ ***else*** $P_i=P_j$ ***then*** $P(i,j)=1.$, where P_i represents the career of user ***i***, and P_j represents the career of user ***j***.

In reality, the user's attributes reflect that the degree of interest to the project is not the same, which can be set by the weight of each attribute to represent. Setting the gender weight w_1, age weight w_2, professional weight w_3, this weight is generally generated by the statistical data or domain experts. This paper gives the final user ***i*** and ***j*** total similarity formula as follows:

$$Simi(i,j)=w_1*S(i,j)+w_2*A(i,j)+w_3*P(i,j)\quad,\quad \sum_{k=1}^{3}w_k=1 \tag{2.1}$$

Through the calculation of the above formula, the user similarity matrix $\boldsymbol{A_{n,m}}$ is produced**.** And then according to the matrix $\boldsymbol{A_{n,m}}$ to calculate the user has not evaluated the project pre-score, the initial pre-score matrix ***Init_R****w,v* is produced. The pre-score calculation method uses the following formula[5]:

$$P_{ai}=\bar{r}_i+\frac{\sum_{j\in NBS_i\cap U_a} simi(i,j)*(r_{aj}-\bar{r}_j)}{\sum_{j\in NBS_i\cap U_a} simi(i,j)} \tag{2.2}$$

Where $\bar{r}_i$ denotes the mean value of user ***i***, ***simi(i , j)*** denotes the similarity between user ***i*** and user ***j***, r_{aj} denotes the score of user ***a*** to item ***j***, $\bar{r}_j$ denotes the mean value of user ***j.***

There is a special case when all neighbor users of the user do not score the item. It is necessary to use the user's score average to fill the pre-score of the item to ensure that the user rating information can be maximized. The calculation is as follows:

$$P_{ai}=\frac{\sum_{i=1}^{N} r_i}{N} \tag{2.3}$$

Where $\boldsymbol{r_i}$ represents the score of user ***a*** for item ***i*** and ***N*** is the number of items.

2.2 Generate a Recommended List with Project Clustering and Timestamp Properties

The initial filling matrix ***Init_R****w,v* can be obtained from Section 2.1, combined with the traditional user-based collaborative filtering algorithm, after calculations of the final project pre-score matrix ***End_R****w,v*. The specific process is as follows:

1） The first step is to calculate the similarity ***sim (i , j)*** of any two users by using the initial project pre-score matrix ***Init_R****w,v*. The common methods for calculating the similarity between users ***i*** and ***j*** are Cosine Similarity, Pearson Correlation Coefficient and Modified Cosine Similarity[7]. This paper uses the modified cosine similarity[8], the formula is as follows:

$$\sin(i,j)=\frac{\sum_{U_{ij}}(r_{iu}-\bar{r}_i)*(r_{ju}-\bar{r}_j)}{\sqrt{\sum_{U_{ij}}(r_{iu}-\bar{r}_i)^2}*\sqrt{(r_{ju}-\bar{r}_j)^2}} \tag{2.4}$$

And then the user similarity matrix $\boldsymbol{B_{n,m}}$ can be obtained. The r_{iu} represents the score of user ***i*** for item ***u***, r_{ju} the score of user ***j*** for item ***u***, $\bar{r}_i$ the mean value of user ***i***, $\bar{r}_j$ the mean value of user ***j***, $\boldsymbol{U_{ij}}$ and the item set of user ***i*** and ***j*** common score.

2） The second step is based on the user similarity degree matrix $\boldsymbol{B_{n\,,m}}$, which can be calculated via the pre-score $\boldsymbol{P_{ai}}$ of a user ***a*** for the unrated item ***i***. The formula is as follows[9]:

$$P_{ai}=\bar{r}_i+\frac{\sum_{j\in NBS_i\cap U_a} \sin(i,j)*(r_{aj}-\bar{r}_j)}{\sum_{j\in NBS_i\cap U_a} \sin(i,j)} \tag{2.5}$$

Finally, the final project pre-score matrix is ***End_R****w ,v* produced. Where $\bar{r}_i$ denotes the mean value of user ***i***, ***sin(i , j)*** denotes the similarity between user ***i*** and user ***j***, r_{aj} denotes the score of user ***a*** to item ***j***, and $\bar{r}_j$ denotes the mean value of user ***j.***

3） Next, all the items in the system are clustered, using ***I*** to indicate a collection of items, $\boldsymbol{I=\{I_1,I_2,I_3,\ldots,I_n\}}$, where ***n*** represents the number of items, ***T*** to represent the item type attribute set, $\boldsymbol{T=\{T_1,T_2,T_3,\ldots,T_m\}}$, where ***m*** represents the number of attributes, and $\boldsymbol{S_{i\,,j}=[0|1]}$, $\boldsymbol{S_{i\,,j}}$ to indicate whether item $\boldsymbol{I_i}$ and item $\boldsymbol{I_j}$ belong to the same class. When $\boldsymbol{S_{i\,,j}=1}$, it means that item $\boldsymbol{I_i}$

and item $\boldsymbol{I_j}$ belong to the same class; When $\boldsymbol{S_{i,j}=0}$, it means that item $\boldsymbol{I_i}$ and item $\boldsymbol{I_j}$ do not belong to the same class. Each type in the attribute project corresponds to digital numbering. If a project has one or more of these types of features, it is marked with "1" at the corresponding position. If it does not have this type of features, "0" is used to mark it. For example, supposing the number of item type attributes is $\boldsymbol{m=10}$, and a project with T1, T4, T8 these three types of features, then the project corresponding to the 10 types of attribute value is "1 0 0 1 0 0 0 1 0 0". The attribute values of the project are used to calculate the similarity between items, then similar or identical items are clustered. Here, we need to set the threshold $\boldsymbol{\varepsilon}$ of similarity between items, that is, when the similarity of two items item sim ***item_sim(I**$_i$ **,I**$_j$**)≥ε***, the two items are clustered into the same class ***Si, j = 1***. In this paper, the similarity of the project is given as follows:

$$items(I_i, I_j)=\sum_{k=1}^{t} r_k*(1-|a_{I_i k}-a_{I_j k}|) \quad , \quad \sum_{k=1}^{t} r_k=1 \qquad (2.6)$$

Where $\boldsymbol{t}$ is the number of type features of the item; $\mathbf{r_k}$ represents the weight of the ***k-th*** type feature. Setting type feature weights is because some types of features tend to better reflect the main types of items. The proportion of the more important types of features will be larger, on the contrary, the proportion of other types of features will be smaller. $a_{I_i k}$ represents the value of the ***k-th*** attribute of the $\boldsymbol{I_i}$ item; $a_{I_j k}$ represents the value of the ***k-th*** attribute of the $\boldsymbol{I_j}$ item.

4） Next, the known clustering results are used to generate initial recommendation list. First of all, Statistics of the target user all pre-evaluation is divided into 5 points of the project. The scoring criteria used in this paper are [1-5] score, with a maximum score of 5, followed by the clustering of these projects, the statistics of the same or the same project where the most project class. Finally, in the corresponding project category in accordance with the user pre-score sub-production of former ***N*** non-rated initial recommendation. The recommended effect of a recommendation system is not only reflected in the accuracy of the pre-score, but also in whether to meet the recommended bit more recommended before the higher hit rate. The use of timestamp properties can satisfy the above requirements to some extent.

5） This article also uses the timestamp attribute[10] to reorder items in the initial recommendation list. First, the target user in the training set has been evaluated in the project in the recent evaluation of the project number. Then the initial list of projects recommended by the similarity is compared with recently evaluated the project, reordering[11] recently evaluated in accordance with the level of similarity of projects. After the reordering of the project list to a certain extent to meet the recommended bit more forward hit rate higher rules, it can not only improve the accuracy of the entire recommendation list, but also make each recommended bit hit rate.

2.3 Recommended Steps for the Algorithm

Algorithm: Collaborative Filtering Recommendation System Based on User Potential Associated Information

Input: Target users to be recommended ***ID***; Number of items to be recommended ***N***

Output: Recommended list of items ***List(N)***

Step 1 According to the user attribute matrix $\boldsymbol{Attr_{x,y}}$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ represent the dimensions of the matrix. The formula **(2.1)** is used to calculate the similarity between users via the user similarity matrix $\boldsymbol{A_{n,m}}$ **.** Then, according to $\boldsymbol{A_{n,m}}$, The formula **(2.2)(2.3)** is used to pre-score the item that the user has not score, and get the initial pre-score matrix $\boldsymbol{Init\_R_{w,v}}$.

Step 2 Use ***Init_R$_{w,v}$*** to calculate the similarity between all users, using the formula **(2.4)** to calculate and obtain the user similarity matrix ***B$_{n,m}$***.

Step 3 Select ***k*** nearest neighbor users for user ***u***, and then pre-score the item ***i*** which is not rated by user ***u*** according to formula **(2.5)** to obtain pre-score ***R$_{u,i}$***.

Step 4 Loop execution step3, until all users have completed a pre-evaluation of the unrated items, getting the final pre-score matrix ***End_R$_{w,v}$***.

Step 5 The items in the system are clustered, using the equation **(2.6)** to calculate the degree of similarity between the item, followed by a given threshold ***ε***, the similar or identical items clustered into the same class.

Step 6 Cluster the highest pre-rated items of user. Statistics of the similar or the same project where the most items class, in the corresponding project class in accordance with the user pre-score produce initial recommendation of the former ***N*** non-rated.

Step 7 The initial recommendation list is reordered in conjunction with the time stamp attribute of the user's rating score to produce the final recommended list ***List (N)***.

The key step of this algorithm is to use the user's potential information in the search nearest neighbor, and integrate that into the project clustering method and timestamp attribute. To a certain extent, this system reduces the sparseness of data and the score error at the same time, as well as effectively improves the recommendation accuracy.

3. Experimental Results and Analysis

This experiment uses a public data set: MovieLens data set. The data set includes a user information table, an item information table, a user item score information table, a movie type information table, and a user occupation information table etc.. In order to ensure the rationality of the experiment, at least 10 users are selected to evaluate the film and at least 10 movies from each user were evaluated as test data, a total of 100000 user-item score data. 100,000-strong data is divided, 80% is used as training data, 20% used as test data. The number of users in the training data is 943, the number of items is 1682, and the number of reviews is 80,000. The number of users in the test data is 462. Two different training sets and test set data are selected, in which the test set data were are intersecting with each other.

3.1 Evaluation Index

1) The average absolute error (MAE) evaluation refers to the deviation value of the user's pre-score data and the actual score data of the user. MAE formula[12] is as follows:

$$MAE=\frac{\sum_t^n |P_t-R_t|}{N} \tag{3.1}$$

Where ***n*** is the number of target user's pre-rated items; ***P$_t$*** represents the user's pre-score for the ***t-th*** item; ***R$_t$*** represents the user's true score for the ***t-th*** item.

2) Accuracy evaluation is one of the important indexes to measure the performance of the algorithm. It is the intuitive standard that the system recommends to meet the user's needs to a certain extent. The accuracy formula[13] is as follows:

$$precision=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|R(u)|} \tag{3.2}$$

Where ***R(u)*** represents the item in the recommended list made by the system for the target user ***u***; ***T(u)*** indicates that the target user ***u*** of the test set is item set on a project with a score of

4 or more; ***U*** is the set of users in the test set, and the user's score is 1 to 5 points.

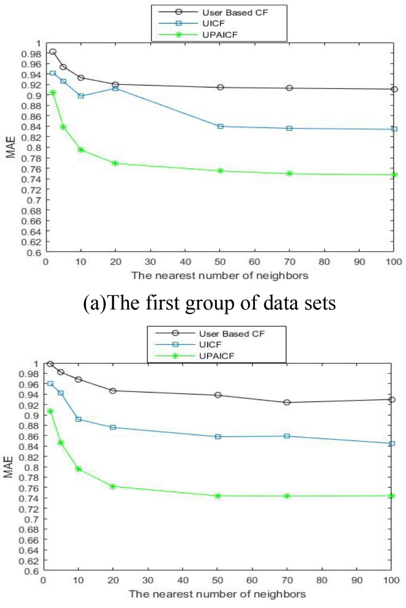
3.2 Experimental comparison results

In this paper, the improved algorithm (UPAICF) proposed in this paper is compared with User-Based CF(UCF)[14] algorithm and A Collaborative Filtering Recommendation Algorithm Incorporated with User Interest Change(UICF)[15] for MAE value and Accuracy.

Experiment 1. The UPAICF algorithm proposed in this paper is compared with the User Based CF and UICF algorithms on the MAE standard.

Many experiments have proved that, the average absolute deviation of the system is lower when the neighborhood $\boldsymbol{M=25}$ and the corresponding gender, age, occupation of the three user characteristics of the weight are $\boldsymbol{w_1 = 0.3}$, $\boldsymbol{w_2 = 0.2}$, $\boldsymbol{w_3 = 0.5}$.

At the above values, the MAE value of the three Kinds comparison algorithm in the two groups of data sets is shown in Fig. 1 below:



(a)The first group of data sets

(b)The second group of data sets

Figure 1: Comparison of MAE Values of Three Algorithms on Two Groups of Data Sets

(1) In case of the improved UPAICF algorithm, when the nearest neighbor number ***k*** takes different values, the MAE value is always lower than UCF and UICF; when the nearest neighbor number ***k=100***, MAE hits its minimum at 0.7473. In cases of the UCF and UICF algorithms, the MAE minimum is MAE=0.9110 and MAE=0.8341 respectively.

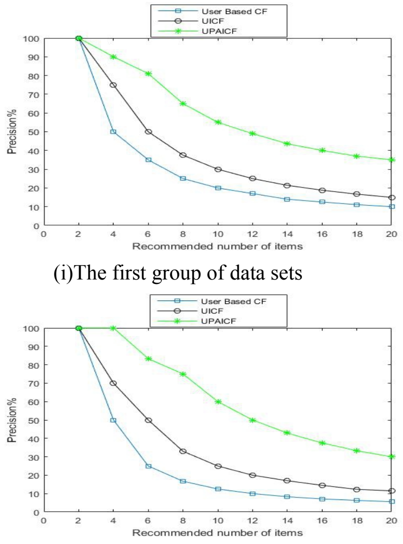
(2) With the increase of the nearest neighbor number ***k***, the MAE values of the three algorithms gradually decrease, the decreasing range of which get smaller and smaller. It shows that when the nearest neighbor number ***k*** is getting smaller and smaller, the performance of the algorithm is more and more important; when the nearest neighbor number ***k*** is getting larger and larger, he performance of the algorithm is getting smaller and smaller effect. Therefore, the selection of nearest neighbor number ***k*** plays a decisive role in the performance of the algorithm.

Comparing graph (a) and (b), when the proposed UPAICF algorithm is based on the nearest neighbor number k=100, MAE has a minimum value is MAE=0.7440; when the nearest neighbor number in the UICF algorithm k=100, MAE has a minimum value MAE=0.8451 and in UCF algorithm the nearest neighbor number k=70, MAE has a minimum value MAE=0.9237. The other information is basically the same as Figure (a).

The two groups of experimental results above have further proven that the improved UPAICF algorithm, compared with the UCF and UICF algorithms, is more effective to reduce the value of MAE.

Experiment 2. The UPAICF algorithm proposed in this paper is compared with the User Based CF and UICF algorithms on the precision rate standard.

The threshold ***ε*** and the weight play a decisive role in the clustering of the project. After several experiments, when the threshold value ***ε*** of the project similarity is set to ***ε = 0.9***, and the weight of each attribute feature is taken ***=1/n (k=1,2,3,…,n)***, better experimental results are obtained, where ***N*** represents the number of project attribute characteristics. The precision rate value of the three Kinds comparison algorithm in the two groups of data sets is shown in Fig. 2 below:



(i)The first group of data sets

(ii)The second group of data sets

Figure 2: Comparison of the Accuracy of the Four Algorithms on Two Groups of Data Sets

The following information can be obtained from the experimental results of the first set of data sets in Fig. 2:

(1) This paper presents an improved UPAICF algorithm in the number of recommended projects. At different values, the accuracy rate of the value is always higher than that of UCF algorithm and UICF algorithm. In addition, when the number of recommended items equals 2, three algorithms are the same value. When the number of recommended items is equal to 2, the improved UPAICF algorithm in this paper achieves the maximum accuracy and the maximum value is precision=100%. When the number of recommended items is 20, the minimum accuracy is achieved, and the minimum value is precision=35%.

(2) With the increase of the number of recommended items, the accuracy of the three algorithms are gradually decreasing, and the decreasing range is getting smaller and smaller. It shows that when the recommended project number is getting smaller and smaller, the accuracy of the algorithm is more and more important; when the number recommended project is getting larger, the accuracy of the algorithm is lower. If the number of recommended items is too small, it may not be able to meet the needs of users to the project, but when the number of recommended items is too large, it will cause the an reduction of the system accuracy. In this case, to select the number of reasonable recommended items is essential.

Compared with the experimental results of graph (a) and graph (b), the other information is basically the same as Figure (a), except that the values of the system accuracy are different when the number of recommended items is different. From the above experimental results, the improved UPAICF algorithm in comparison with UCF and UICF algorithm is more effective to enhance the accuracy of the system.

4.Conclusion and Outlook

This paper presents a collaborative filtering algorithm based on hybrid recommendation. This paper verifies that the method can not only effectively reduce the average absolute error (MAE) but also can greatly improve the accuracy of the recommended system on the MovieLens dataset. At the same time, there is a need for further research on how to establish a reliable and reasonable similarity calculation model.

References

[1] Schafer J B, Konstan J A, Riedl J. *E-commerce Recommendation Applications*[J]. Data Mining and Knowledge Discovery,2001,5(1/2): 115-153.

[2] Resnick P, Varian H R．*Recommender Systems*[J]．Communications of the ACM,1997, 40(3):56-58.

[3] Hyung JA. *A new similarity measure for collaborative filtering to alleviate the new user cold - starting problem*[J]. Information Sciences, 2008,178(1):37-51.

[4] SARWAR B M, KARYPIS G, KONSTAN J A, et al. *Application of dimensionality reduction in recommender system-A case study*[J]. Expert Systems with Applications,2004,26(2):233-246.

[5] Wu Yi Fan, Wang Hao Ran. *Collaborative Filtering Algorithm Using User Background Information*[J]. Journal of Computer Applications,2008, (11):2972-2974.

[6] Yang Yan. *Research on Collaborative Filtering Recommendation Algorithm Based on Project clustering*[D]. Chang chun: Northeast Normal University, 2005.

[7] LENG Ya Jun, LU Qing, LIANG Chang Yong. *Survey of Recommendation Based on Collaborative Filtering*[J]. Pattern Recognition and Artificial Intelligence,2014,27(8):720-734.

[8] MENG Xiang Wu, LIU Shu Dong, ZHANG Yu Jie, et al. *Research on Social Recommender Systems*[J]. Journal of Software,2015,26(6):1356-1372.

[9] Guo Lan Jie，Liang Ji Ye, Zhao Xing Wang. *Collaborative Filtering Recommendation Algorithm Incorporating Social Network Information*[J]. Pattern Recognition and Artificial Intelligence，2016, 29 (03):281-288.

[10] Y. Ding and X. Li. *Time weight collaborative filtering*. Proc[J]. 14th ACM international conference on Information and knowledge management (CIKM'04),pp.485–492,2004.

[11] Sun Guang Fu, Wu Le, Liu Qi, Zhu Chen, et al.. *Recommendations Based on Collaborative Filtering by Exploiting Sequential Behaviors*[J]. Journal of Software,2013,24(11):2721-2733.

[12]SARWARB, KARYPISG, KONSTANJ, etal. *Item-based Collaborative Filtering Recommendation Algorithms* [C] // Proceedings of the 10th International World Wide Web Conference[J]. New York: ACM Press, 2001:285-295.

[13]Su JH, Yeh HH, Yu PS, Tseng VS. *Music Recommendation Using Content and Context Information Mining*[J]. IEEE Intelligent Systems, 2010, 25(1):16-26.

[14]Li Hua, Zhang Yu, Sun Jun Hua. *Research on Collaborative Filtering Recommendation Based on User Fuzzy Clustering*[J]. Computer Science, 2012,39(12):83-86.

[15]Xing Chun Xiao, Gao Feng Rong, Zhan Si Nan, et al. *A Collaborative Filtering Recommendation Algorithm Incorporated with User Interest Change*[J]. Journal of Computer Research and Development, 2007,(02):296-301.