International Journal of u- and e- Service, Science and Technology
Vol.8, No.4 (2015), pp.311-320
http://dx.doi.org/10.14257/ijunesst.2015.8.4.28

Collaborative Filtering Recommendation Algorithm Based on User Interests

Zuping Liu

Sichuan Vocational and Technical College,
Suining 629000, china
641433894@qq.com

Abstract

To overcome the problem of too much sparse *by scoring matrix, the paper proposes from the part of user interests the collaborative filtering recommendation algorithm based on such interest. In the e-commerce websites, recommendation items have various appearance, function and category attributes. Items which hold similar characteristics generally gain approximate scoring values. The aforesaid method extracts useful information for improvement from user interests. Users' scorings about different items imply their modes of interest. Interest association exists amongst items that were evaluated by the same user. Since individual interest shifts, the intensity of such correlation will gradually change along with days. By building interest intensity model with time decay, and discovering interest correlation among different items through that model, the proposed algorithm can predict scoring matrix and fill it, which is helpful to alleviate problems with* sparse *caused by user-item scoring matrix.*

Keywords: *collaborative filtering, recommendation algorithm, sparse matrix, similarity measure, user interest*

1. Introduction

User interests mining is a complicated issue in the collaborative filtering recommendation systems. User interest means users' option and preference for recommended items. But such interest is not constant [1-2]. Psychologically, interest is classified into long-term and short-term interest. The interest will change along with users' ages and environment to which users are adapted. Numerous researchers used association rule to mine user interests, e.g. recommendation algorithm based on association rule, which was developed by Choonho Kim [3] et al. Here, the paper proposed algorithm of collaborative filtering based on user interests (CFBUI). The strategy establishes the interest model with time reduction by relying on association rule mining. It portrays user interest variation. It predicts user-item scoring matrix and replenish it. Results reveal that the new method improved effectively the accuracy of recommendation [4-5].

In the paper, starting from user interests, it presents algorithm of collaborative filtering based on user Interests (CFBUI), which weakens the sparseness-related problems by user-item scoring matrix.

2. User Interests

It's generally believed that user scoring values against recommended items indicate the degree of a user's interest in one item [6-7]. If the value is 5 by a user against one item, it means high degree of recognition of that item; if the value is 1, meaning no much interest in the item. In video websites, it is commonly seen that some viewers detest some videos and state to give negative 5 or even negative 100 scores [8-9]. In fact, compared with

ISSN: 2005-4246 IJUNESST

videos which were not valued by watchers, those disgusting videos have features interesting to users. It's only because of interest in contents in those videos that users view them. Perhaps those contents are not their taste [10]. So they give negative comments. A majority of users won't evaluate videos which are not interesting at all. Users' application of a specific item represents their interest. If a user evaluates both item I_i and I_j, it means it shows interest in both items. Then, there must be interest association between item I_i and I_j. The paper proposed algorithm can dig out plenty of interest association through scanning user-item scoring matrix, as to fill in the spare matrix for higher accuracy rate of recommendation [11-12].

3. Collaborative Filtering Recommendation Algorithm Based on User Interests

3.1. Interest Association Analysis in CFBUI

In user-item evaluation matrix, when a user scores two items, it is thought that interest association exists between them. We use confidence level in the association rule to define association degree of both items. Set $count(I_i)$ the number of times when item I_i is rated by all users; $count(I_i, I_j)$ refers to the number of items when both item I_i and I_j are commented by same users; I_i - I_j 's interest association degree is defined like:

$$r(I_i, I_j) = \frac{count(I_i, I_j)}{count(I_i)} \quad (1)$$

Interest correlation is not symmetric between I_i and I_j, similarly I_i - I_j correlation:

$$r(I_j, I_i) = \frac{count(I_i, I_j)}{count(I_j)} \quad (2)$$

We set min_support and min-confidence. If $count(I_i, I_j)$ occurs no more than min_support, then $r(I_i, I_j) = 0$; if association degree $r(I_i, I_j)$ does not satisfy min-confidence, then $r(I_i, I_j) = 0$.

Through the scan of user-item scoring matrix, we can have the interest association degree among all recommended items and create the related matrix, as shown in Table 1.

Table 1. Item Correlation Matrix

Item	I_1	I_2	I_3	…
I_1	1	0.1	0.7	
I_2	0.5	1	0.76	
I_3	0.36	0.4	1	
…				

3.2. Interest Association Degree Analysis with Time Decay in CFBUI

User interest is not unchanging. For the change of user itself, its interest will vary along with time. Classic collaborative filtering algorithm can't cope well with such changes. Take for instance, user A used to love watching adventure films; while presently, it prefers nature documentary. User B loves adventure movies all the time. User C loves nature documentary. Since A and B have similar interest, their scores about

adventure items are close. When user A's interest is shifting, the similarity between A and B is higher than between A and C. when A's neighboring users are determined, traditional collaborative filtering recommendation methods will choose B instead of C. finally, what's recommended by the system will remain adventure films rather than nature documentary, despite currently, A and C share the same hobby.

Apparently, actions occurring earlier have unlike effects from the most recent ones. Actions which happened lately are more influential. Every score has its time attribute. The difference of time when two items are rated by the same user will affect the degree of association between them. If the difference is extremely big, we may think the association disappears. If it's small, there might be strong degree of association between them. Some scholars explored the change of user interests in terms of time damping. Zheng Xianrong *et al.* [13] used scoring time as affecting weight to design similarity calculation method based on time linear reduction, which captured user interest features in some extent.

We assume interest-time decaying process is just like the memory deterioration, which is a approximate exponential curve. Here we adopt the time damping function which bases on exponential function:

$$w(I_i, I_j) = e^{-k(abs(time(I_i)-time(I_j)))} \quad (3)$$

Where, $\text{time}(I_i)$ is what used for scoring, by second, calculation starting from January 1, 1970; k is the rate of forgetting. $k > 0,\ 0 < w(I_i, I_j) < 1$.

3.3. Comprehensive Similarity of CFBUI

We discussed about how to integrate scoring similarity and user interest association degree. User interest association degree is closely connected with co-scoring item numbers and non-commonly rated item numbers. User-item scoring matrix is extremely sparse in practical systems. Items for co-assessment are found very few. That will cause numbers of both commonly and non-commonly rated items for calculation to have big effects on estimating similarity. We'll introduce the definition of such two items.

3.3.1Co-rated Items: Take U_1 and U_2 in Table2 for example. Both U_1 and U_2 evaluate item I_2, which is thus called co-rated item of U_1 and U_2.

Table 2. User-item Scoring Matrix

Item	I_1	I_2	I_3
U_1	1	4	2
U_2	1	5	3
U_3	2	?	1

We set three comparative objects X, Y and Z. If X and Y have six commonly evaluated items, X and Z have four, their results might be close when estimated by traditional methods for the similarity, because traditional calculation doesn't take into account the number of items involved for that. But in effect, X and Y have a higher degree of similarity. Figure 1 explains that.

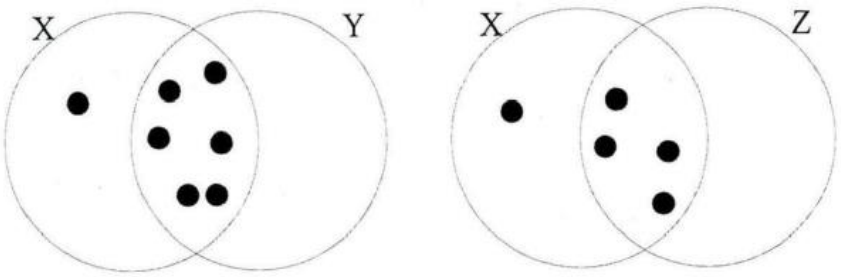

Figure 1. Different Number of Common Score Items

3.3.2 Non-commonly Appraised Items: Take U_1 and U_2 in Table2 for instance. Only U_1 scores item I_1, other than U_2; only U_2 scores items I_3, not U_1. Then, I_1 and I_3 are non-common items.

Suppose there are only four common items and two non-common items in X , Y; four common items and seven non-common ones in X and Z. If calculated by traditional methods, results of similarity might not be different. In the case of alike similarity between X ,Y and X , Z, the priority selection of X , Z will bring to similar user collection the interference of scoring values actually not involved in similarity calculation. Figure 2 shows that to us.

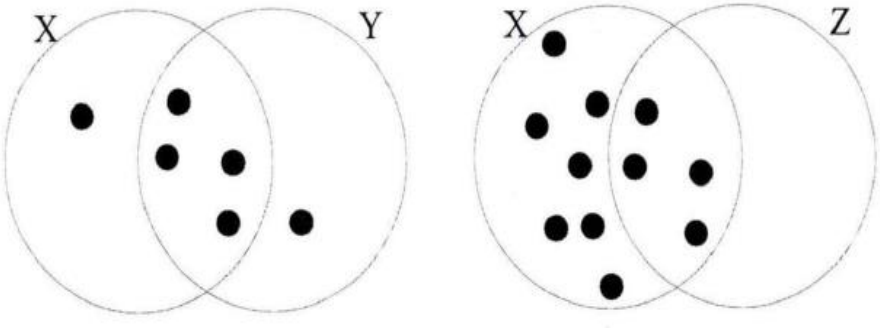

Figure 2. Different Number of Non-common Score Items

User interest correlation, the number of co-score items and the number of non-common score items are closely relationship. User interest correlation degree is bigger, it was said that co-score items proportion is higher, the proportion of non-common score items is low. Through the above analysis, the user interest correlation degree is bigger, in similarity computation, error is small. It can use adjustable parameter β to put score similarity and interest relation together, score similarity expression more perfect, to be better express the similarity of score. $\beta \in [0,1]$, the formula is as follows:

$$sim_3(I_i, I_j) = \beta, sim_{manh}(I_i, I_j) + (1-\beta) \cdot r_2(I_i, I_j) \qquad (4)$$

3.4. Workflow of the Proposed Method

By relying on user-based collaborative filtering algorithm, CFBUI has additionally a preprocessing procedure of scoring matrix. In the scoring and filling process, the predictive formula applied for scoring items is weight-similarity predictive formula, like:

$$predict(I_i, U_x) = \frac{\sum_{I \in Nei(I_i)} (r_{x,j} \times sim_3(I_i, I_j))}{\sum_{I \in Nei(I_i)} sim_3(I_i, I_j)} \qquad (5)$$

Where, $Nei(I_i)$ indicates $Nei(I_i)$ similar items set. CFBUI pseudo code is as follows:

Input: Target user U_x, user-item score matrix

Output: Recommended items set of target user U_x,

For each I_i in items set

For each I_j in items set

With the formula (4) to compute the similarity sim between user I_i and I_j.

If sim > Similarity threshold of items set

Put I_j to join the similar items set

For each unknown score in score set of I_i

Based on the formula (5) to compute predicted value

Filling the unknown score to pseudo scoring matrix

Output pseudo scoring matrix

For each unknown score in Unknown score set of user U_x

SumSim=0

SumScore=0

For each U_x in Similar neighbor set

SumScore+= $r_{x,i}^{*} sim_{ad\cos}$

SumSim+= $sim_{ad\cos}$

PreScore=SumScore/SumSim

4. Experiment Design and Discussion

4.1. Evaluation Standard

Researchers proposed many evaluation standards to validate the effectiveness of recommendation systems. They largely include two board categories: (1) validation of accuracy of recommended results; (2) confirmation of algorithm's temporal and spatial complexity. The paper concerns the performance improvement of proposed method from the point of recommendation precision. Mean average error (MAE) is chosen as the standard. That evaluation method firstly hides the real points of items by target users; then, it forecasts scores of those items with the help of recommendation algorithm; next, differentials between the predictive and real values are cumulated to be finally shared for the mean average error. Set predictive scoring points $\{p_1, p_2, ..., p_n\}$, corresponding to real points $\{q_1, q_2, ..., q_n\}$. The MAE can be obtained through the expression:

$$MAE = \frac{\sum_{i=1}^{n} | p_i - q_i |}{n} \quad (6)$$

4.2. Dataset

We chose data from grouplens (http://www.grouplens.org/) to use as test dataset. Grouplens collected a huge number of scoring information relating to movielen websites and made up to data packets of validation. Lots of collaborative filtering algorithms made use of those packages to verify the performance and precision of methods. The data packets we selected here include 100000 pieces of scoring information about 1682 movies assessed by 943 users. The dataset for experiment has sparsity of $1-100000/(943*1682)=0.9370$, which means the scoring matrix is very sparse.

Those data packages are consisted of u.data, u.info, u.item, u.genre, u.user and u.occupation.

u.data refers to score data in those packets. Data is recorded in rows and separated by vertical bars in the middle, just like user no. | movie no. | scoring points | time. The time should start since January 1, 1970, in seconds.

u.info has only one row, indicating the number of users, items and scoring data.

u.item shows the information about every single movie. Data is recorded in rows and separated by vertical bars in the middle, like movie no. | movie title | release date | show date | websites introducing movies in the IIMDB pool | unknown style | action | adventure | cartoons | preteen film | comedy | crime | documentary | drama | fiction | black and white | dracula | music | romance | science | thriller | war | cowboy. The last 19 items refer to movie's style and genre. Their attribute values are 0 or 1. 0 means none; 1 means yes.

u.genre has the list of genre attribute.

u.user is attribute of describing users. Data is recorded in rows and separated by vertical bars in the middle, like user no. | age | sex | occupation | postal code.

4.3. Experimental Strategy

The paper realized user-based classic collaborative filtering (CCF), collaborative filtering based on pre-filling (CFBPF) based on item scoring similarity, collaborative filtering based on item features (CFBIF) and collaborative filtering based on user interests (CFBUI). To avoid the size of similar neighboring collections affecting results, we still performed tests on the above mentioned five types of collections, of which the size is respectively 4, 8, 12, 16 and 20. Test data sets are partitioned into two mutually disjoint subsets, one for training set and the other for test set, at the ratio of 4:1. Data sets are randomly divided several times and experiment repeats, which will help reduce errors caused by such division.

4.4. Test Environment

The experiment program runs on Windows 7 operation system, which was compiled by C# language in Visual Studio Ultimate version 2010 development tools. .Net Framework2.0 or upper version should be installed before the program runs. The experimental machine is configured: Intel Core 2 Duo P8800 2.66GHz for CPU, 4G memory capacity.

4.5. Analysis of Parameters

β refers to the proportion of item scoring similarity affecting the comprehensive similarity. In different real contexts, data will show different features. To achieve the optimal outcome, we can train parameter β to optimize it as for different scenarios. For the experiment, we set forgetting parameter k=0 to test β for five sizes of similar user collections.

We can see from Figure 3, when β tends to grow up, MAE declines on the whole. When β is close to 0.95. MAE approaches the smallest for all five sizes and the algorithm reaches the best accuracy. When β is about 1, MAE begins to rise. Obviously, item scoring similarity influences greatly the comprehensive similarity. Attribute similarity affects the least.

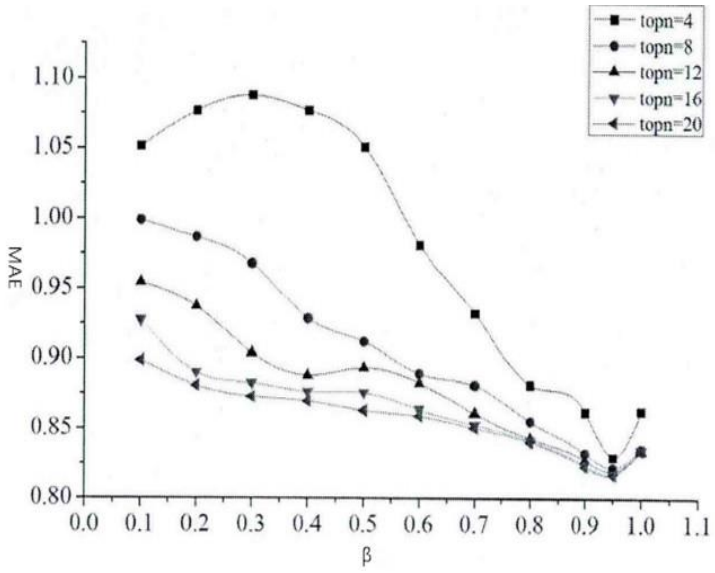

Figure 3. β Parameters Affection on CFBUI Algorithm

k stands for the rate of association degree changing with time decaying. If it decays too fast, only the latest scoring actions can reflect the degree of association. Since scoring matrix is considerably sparse and the most recent scorings fewer, systems' recommendation accuracy is remarkably reduced. If it decays too slowly, it will turn out to be the same with no time damping. To know interest losing on the exact time scale, it's required to measure k in real cases. $\beta = 0.95$ is better choice. So in that case we test k.

Where, when $k = 0$, suggesting no consideration of forgetting features. From Figure4 we can learn that only when $\text{topn} = 4$, the addition of forgetting parameter will make MAE bigger, meaning worse outcomes. In most cases, with account of forgetting factors, the precision of recommendation results can be enhanced.

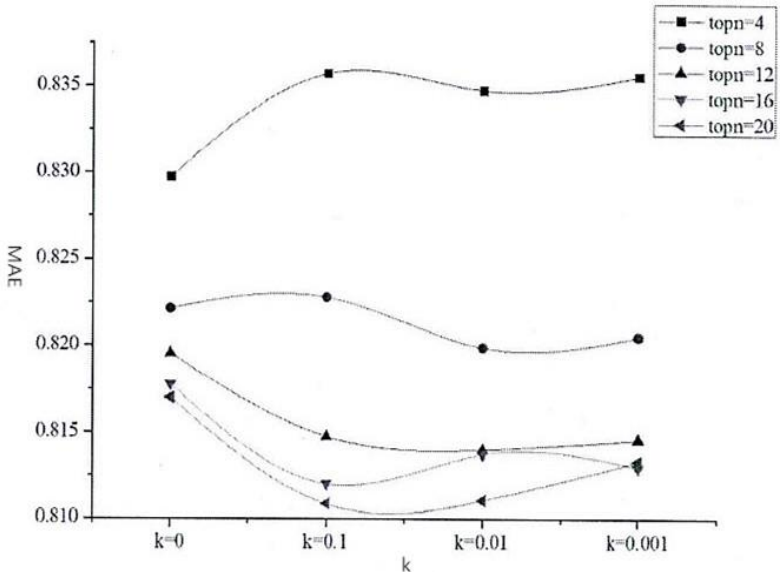

Figure 4. k Parameters Affection on CFBUI Algorithm

4.6. Analysis of Results

Results of each algorithm are compared in Figure 5-6. Figure 5 gives MAE values of each algorithm for comparison. Results by CCF method are far different from others. In

the same coordinate system, MAE values of other techniques stay together. For in-depth comparison among the other methods, Figure 6 portrays their curved lines of MAE values.

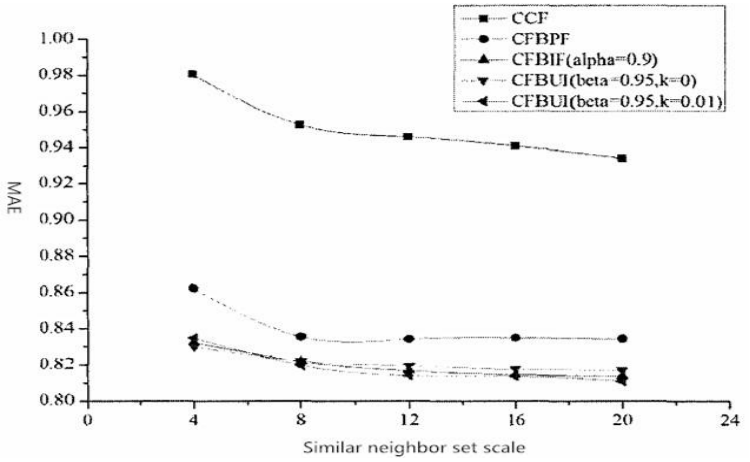

Figure 5. Comparison of MAE Value with CCF

As it's seen, MAE values would scale down with the enlargement of similar adjacent collections, with higher accuracy of prediction. MAE values descend unevenly, from fast to slowly. That means with magnification of alike neighboring collections, the similarity becomes smaller and smaller between following users and target users, lowering the precision of prediction. The size of similar neighboring collections has connection with the size of scoring data set and sparseness of user-item scoring matrix. They should be determined according to real situation.

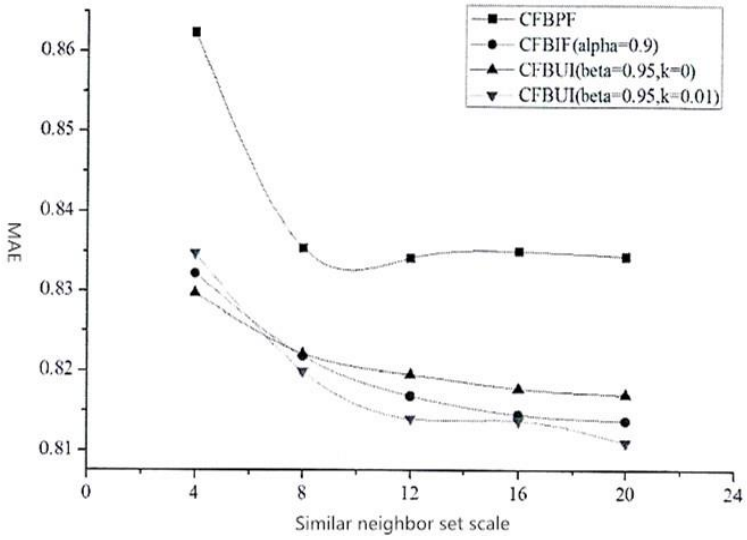

Figure 6. Comparison of MAE Value with Not CCF

5. Conclusion

User interest mining is a complex problem in Collaborative filtering recommendation system. User's interest is user preferences on the recommended items. The user's interest is not immutable and frozen. User interest will change with age and environment. This paper proposed CFBUI algorithm, through establishing user interest model with time decay, mining interest correlation between items and improving similarity measure

method, then the unknown user-item scoring matrix were predicted and filling. To overcome problem of the sparse user-item scoring matrix, to improve the accuracy of collaborative filtering algorithm.

References

[1] L. Sun and M. Huang, "Comprehensive user characteristics and properties of project collaborative filtering recommendation algorithm", The research and application of computer, vol. 2, (**2014**), pp. 384-387.
[2] F. Zhang, S. Liu, Z. Li and J. Sun, "A collaborative filtering recommendation algorithm fusion user reviews and environmental information", Micro computer system, vol. 2, (**2014)**, pp. 228-232.
[3] C. Kim and J. Kim, "A Recommendation Algorithm Using Multi-Level Association Rules", Proceedings of the IEEE/WIC International Conference on Web Intelligence, (**2003**), pp. 524-527
[4] Kerui. Shi and J. Liu, "The two step is to similar effect on collaborative filtering algorithm", Journal of University of Shanghai for Science and Technology, vol. 1, (**2014**), pp. 31-33.
[5] R. Huigui, a fire Xu, Hu Chunhua and Mo 进侠, "Collaborative filtering recommendation algorithm based on user similarity", Journal of China Institute of communications, vol. 2, (**2014**), pp. 16-24.
[6] Q. Cai, D. Han, H. Li, Y. Hu and Y. Chen, "Personalized resource labels and recommendation based on collaborative filtering", Computer science, vol. 1, (**2014**), pp. 69-71
[7] Y. Shen, G. Guo and J. Wu, "The recommendation and collaborative filtering algorithm based on mobile scenarios", Science technology and engineering, vol. 8, (**2014**), pp. 49-52
[8] H. Cao and K. Fu Fu, "Collaborative filtering recommendation system research method of clustering search", Computer engineering and applications, vol. 5, (**2014**), pp. 16-20.
[9] C. Lu, A. Hong, and J. Gong, "Algorithm Research on Collaborative Filtering Recommendation Based on PSO", Computer engineering and applications, vol. 5, (**2014**), pp. 101-107.
[10] Z. Feng, D. S. Yi, H. Qin and H. Deng, "Clustering genetic algorithm and collaborative filtering recommendation algorithm based on the combination of computer technology and development", vol. 1, (**2014**), pp. 35-38.
[11] X. Zhang, "Trust preferred personalized recommendation based on", The application of computer system, vol. 1, (**2014**), pp. 109-113.
[12] T. Chen, J. Shuai and M. Zhu, "A method is recommended based on collaborative filtering film", Computer Engineering, vol. 1, (**2014**), pp. 55-58
[13] X. Zheng and X. Cao, "Research on lineal gradual forgetting collaborative filtering algorithm", Computer Engineering, vol. 3, no. 3, 6, (**2007**), pp. 72-74.

Author

Zuping Liu, She received her master's degree of Sichuan University. Now she is an associate professor in Sichuan vocational and technical college. Her major fields of study are computer application.

