International Journal of Grid and Distributed Computing
Vol. 9, No. 12 (2016), pp.203-212
http://dx.doi.org/10.14257/ijgdc.2016.9.12.18

A Collaborative Filtering Recommendation Algorithm Fusing Rating and Time Interval Similarity on Item Attributes

Xiao-hui Cheng[1], Yu Wu[2] and Yun Deng[3,*]

[1,2,3]*College of Information Science and Engineering, Guilin University of Technology, Guilin Guangxi, China*
[1]*cxiaohui@glut.edu.cn*, [2]*wushiyuan51@126.com*, [3]*woshidengyun@sina.com*

Abstract

Traditional methods UBCF have limitations of poor recommendation quality and problems of data sparsity. To alleviate these problems, a novel collaborative filtering algorithm is designed, which firstly get the users' ratings and time intervals for each attribute from the users' ratings for items, then produce two methods to calculate the similarity between users, introduce a weighting parameters to control the weight between the two similarity methods in order to get a fusion similarity between two users. The results show that this method is able to improve the accuracy of predicted values, resulting in improving recommendation quality of the collaborative filtering recommendation algorithm.

Keywords*: attribute ratings, time interval, similarity, fusion*

1. Introduction

Collaborative filtering recommendation algorithm is widely used for it is only directed to users' social, rather than analysis and processing large volumes of complex content, that is to analysis the rating information for the user, instead of analyzing and processing complex content such as audio and video. The unit user evaluated is an item like a movie or a music. Collaborative filtering recommendation algorithm predict ratings of unrated items through historical rate information that users gave items, and recommend items that has high predict ratings to users. The accuracy of recommendation is the most important factor to evaluate a recommendation algorithm.

2. Defects to User-based Collaborative Filtering

The basic idea of traditional user-based collaborative filtering recommendation algorithm (User-based Collaborative Filtering, UBCF) is that to recommend items to you which interested by other users who have same interests with you. The main method is to find the neighbor users though user ratings for items and recommend items to the target user which is not rated by target user and is highly rated by its neighbor users.

The key of UBCF algorithms is to find neighbor users to target user, it accords user ratings for items to calculate similarity between users by similarity calculation. The algorithm only regard items as unit that user interested in, if the user rate high on an item, then the user is interested in the item, but it ignores the fact that a user may not be interested in a single project, he may be interested in the items with some special attributes. There must be more than one property of an item, for example, a film may contains two properties what is drama and martial arts, if there are two users seeing this movie and giving a high score, while one of the two users is interested in drama and the

* Corresponding Author:
Email: woshidengyun@sina.com (Yun Deng)

ISSN: 2005-4262 IJGDC

other is interested in martial arts, UBCF algorithm will treat these two users as similar in this item, and recommend martial arts movies which is interested by the second user to the first user, and recommend drama movies which is interested by the first user to the second user. This will reduce the accuracy of recommendation.

Xu *et al*. [1] get user's rating to attributes according to the change width of ratings target user rated items and the score calculated by collaborative filtering recommendation. Yang *et al*. [2] statistics the number of rate times to optimize the user similarity on the perspective of item attributes. Sun *et al*. [3] predicted the unrated items by analyzing different users' interests to various attributes of items and integrating the attributes of rated items to reduce the sparsity of datasets, and then to improve the accuracy of items' similarity calculation Ren Lei [4] aims at the causes for the concept drift, a new collaborative filtering algorithm using marking time features was presented. Sun GF*et al*. [5] build the relationships structure between users or items through sequential information, then calculate similarity, integrate the similarity into probability matrix decomposition algorithm. XIONG Zhong-yang [6] improved the algorithm based on the technique of item classification. However, these studies did not take into account the comprehensive of utilization the ratings on item attributes and time interval of ratings on item attributes to calculate the similarity. YAO Ping-Ping [7] synthesizes user preferences and the project properties, but she didn't consider time influence.

This paper presents a collaborative filtering recommendation algorithm based on similarity fusion of evaluation of project properties and time intervals (Based Attribute Rating and Time Interval Collaborative Filtering, BTCF). The core of the algorithm is find out the item attributes users really interested in from ratings and time interval users gave items, then calculate the similarity between users through the two angles, weight the two similarities to get the real similarity to users, and get the neighbors of user to calculate predict ratings. The method dispersed item ratings to the attributes, and reduce the sparsity of data.

3. Fusion of Rating Similarity and Time Interval on the Item Attributes

3.1. Similarity Computation

Similarity calculation between users is directly related to recommendation quality. Accurate calculation of similarity is the key to improve the quality of recommendation system. Similarity computation is the key in collaborative filtering recommend algorithm. The most widely used similarities are cosine similarity, adjusted cosine similarity and Pearson Correlation coefficient similarity. All these three methods are based on vector similarity calculation. Cosine similarity user cosine angle of users' rating vector to measure similarity, adjusted cosine similarity is based on cosine similarity and counteract the effects of rating size users gave items. Pearson Correlation coefficient similarity calculate similarity according to common rating items. In the paper, we take all the attributes of items into account, so we choose the adjusted cosine similarity as the basic similarity method, it's defined as follows.

$$\mathrm{sim}(u,v)=\frac{\sum_{i\in I_{u,v}}\left(R_{u,i}-\overline{R_u}\right)\left(R_{v,i}-\overline{R_v}\right)}{\sqrt{\sum_{i\in I_u}\left(R_{u,i}-\overline{R_u}\right)^2}\sqrt{\sum_{i\in I_v}\left(R_{v,i}-\overline{R_v}\right)^2}} \quad (1)$$

where $I_{u,v}$ represent the items that are rated by both user u and user v. I_u represents the items user u had rated. I_v represent the items that user v had rated. $\overline{R_u}$ represent the average of ratings user u gave items. $\overline{R_v}$ represent the average of ratings user v gave items. $R_{u,i}$ represent the rating user u gave item i, $R_{v,i}$ represent the rating user v gave item i.

3.2. The Matrix of User-item, Item-attribute and Time-interval

Assume that the total number of users participate in rating is m, the number of items been rated is n, then can produce a $m \times n$ matrix of user-item named R. Shown in the following table 1.

Table 1. User-item Matrix R

	$Item_1$	⋯	$Item_j$	⋯	$Item_n$
U_1	$r_{1,1}$	⋯	$r_{1,j}$	⋯	$r_{1,n}$
⋮	⋮		⋮		⋮
U_i	$r_{i,1}$	⋯	$r_{i,j}$	⋯	$r_{i,n}$
⋮	⋮		⋮		⋮
U_m	$r_{m,1}$	⋯	$r_{m,j}$	⋯	$r_{m,n}$

According to the relationship between items and its attributes, item attributes matrix A can be produced. If the item j have the attribute k, then have $a_{j,k}=1$ in matrix A, or $a_{j,k}=0$ if the item i don't have the attribute k. Shown in the following table 2.

Table 2. Item-attribute Matrix A

	$attr_1$	⋯	$attr_k$	⋯	$attr_p$
$Item_1$	$a_{1,1}$	⋯	$a_{1,k}$	⋯	$a_{1,p}$
⋮	⋮		⋮		⋮
$Item_j$	$a_{j,1}$	⋯	$a_{j,k}$	⋯	$a_{j,p}$
⋮	⋮		⋮		⋮
$Item_n$	$a_{n,1}$	⋯	$a_{n,k}$	⋯	$a_{n,p}$

According to the time difference between rating time and release time, compute the time interval. Shown in the following table 3.

In the table of time interval, $t_{i,j}=time_{u,j}-date_j$, $time_{u,j}$ is the time user u rate the item j, and $date_j$ is the release time of item j.

Table 3. Time Interval between Rating Time and Release Time Matrix T

	$Item_1$	⋯	$Item_j$	⋯	$Item_n$
U_1	$t_{1,1}$	⋯	$t_{1,j}$	⋯	$t_{1,n}$
⋮	⋮		⋮		⋮
U_i	$t_{i,1}$	⋯	$t_{i,j}$	⋯	$t_{i,n}$
⋮	⋮		⋮		⋮
U_m	$t_{m,1}$	⋯	$t_{m,j}$	⋯	$t_{m,n}$

3.3. Similarity of User-attribute

1) user-item-attribute similarity

If two users' ratings on the same attributes are close, it indicates that the two users have the same interests. Due to a project must have multiple attributes, it can't judge whether the user preference for the item attributes just by a single rating on an item. Through users' ratings on all items to generate attribute ratings, it can not only reduce the spare of user ratings matrix, and also can find the root cause of users' interests on attributes more accurately.

According to the matrix R and A, the rating weight matrix of item user rated to attributes can be built for each user which is shown in the following table 4.

Table 4. Item-attribute Rating Matrix B_1 for Each User

	$attr_1$	⋯	$attr_k$	⋯	$attr_p$
$Item_1$	$c_{1,1}$	⋯	$c_{1,k}$	⋯	$c_{1,p}$
⋮	⋮		⋮		⋮
$Item_j$	$c_{j,1}$	⋯	$c_{j,k}$	⋯	$c_{j,p}$
⋮	⋮		⋮		⋮
$Item_n$	$c_{n,1}$	⋯	$c_{n,k}$	⋯	$c_{n,p}$

Among them, $c_{j,k} = \frac{r_{i,j} * a_{j,k}}{\sum_{k=1}^{p} a_{j,k}}$.

Then, the matrix users to each attribute on ratings can be calculated by the formula shown below. The matrix users to each attribute matrix can be shown below as table 5.

Table 5. User-attribute Rating Matrix B_2

	$attr_1$	⋯	$attr_k$	⋯	$attr_p$
U_1	$b_{1,1}$	⋯	$b_{1,k}$	⋯	$b_{1,p}$
⋮	⋮		⋮		⋮
U_i	$b_{i,1}$	⋯	$b_{i,k}$	⋯	$b_{i,p}$
⋮	⋮		⋮		⋮
U_m	$b_{m,1}$	⋯	$b_{m,k}$	⋯	$b_{m,p}$

Among them, $b_{i,k} = \frac{\sum_{j=1}^{n} c_{j,k}}{Num_{I_{u,k}}}$, in the formula, Num_{I_u} represents the items number which is rated by the user u and have the attribute of k.

Then, we can get attribute rating similarity between users by using the below formula (2).

$$\mathrm{sim}_b(\mathrm{u},\mathrm{v}) = \frac{\sum_{k=1}^{p}(b_{u,k} - \overline{b_u})\ (b_{v,k} - \overline{b_v})}{\sqrt{\sum_{k=1}^{p}(b_{u,k} - \overline{b_u})^2}\sqrt{\sum_{k=1}^{p}(b_{v,k} - \overline{b_v})^2}} \quad (2)$$

2) user-time-attribute similarity

The information that user rating time characteristics on each attribute can also reflect users' preference for each attribute. If two users are close on the time interval between rating time and release time, it indicates they are similarity. In the same way, for an item have multiple attributes, the user's preference can be identified through the time interval. This can understand the user's interest more completely.

According to the matrix T and A, the time weight matrix of item user rate time interval can be built for each user which is shown in the following table 6.

Among them, $tp_{i,k} = \sum_{j=1}^{n} \left(\frac{t_{i,j} * a_{j,k}}{\sum_{k=1}^{p} c_{j,k}} \right)$, in the formula, Num_{I_i} represents the items total number which is rated by the user i. The smaller the factor $\left(\frac{t_{i,j} * a_{j,k}}{\sum_{k=1}^{p} c_{j,k}} \right)$ value, the bigger the factor $tp_{i,k}$, and represents the more active the user rate the item which has this attribute. $tp_{i,k}$ reflects the preference of users to the special attribute.

Table 6. User-ratetime-interval Matrix *TP*

	$attr_1$	⋯	$attr_k$	⋯	$attr_p$
U_1	$tp_{1,1}$	⋯	$tp_{1,k}$	⋯	$tp_{1,p}$
⋮	⋮		⋮		⋮
U_i	$tp_{i,1}$	⋯	$tp_{i,k}$	⋯	$tp_{i,p}$
⋮	⋮		⋮		⋮
U_m	$tp_{m,1}$	⋯	$tp_{m,k}$	⋯	$tp_{m,p}$

Then we can get attribute time similarity between users by using the formula (3).

$$\mathrm{sim}_t(u,v) = \frac{\sum_{k=1}^{p} (b_{u,k} - \overline{b_u})(b_{v,k} - \overline{b_v})}{\sqrt{\sum_{k=1}^{p} (b_{u,k} - \overline{b_u})^2} \sqrt{\sum_{k=1}^{p} (b_{v,k} - \overline{b_v})^2}} \quad (3)$$

3.4. Similarity Fusion

In order to get a more comprehensive view of the similarity of users' interests to attribute, the two similarities can be integrated into a new real similarity.

$$sim(u,v) = \alpha * sim_b(u,v) + (1-\alpha) * sim_t(*) \quad (4)$$

In the similarity formula, introduce a weight parameter to control the weight of two similarities, so that the weighted proportion between user-attribute rating similarity and user-attribute time similarity can be change through adjusting the size of α to gain optimal similarity results.

3.5. Predict Ratings

After the nearest neighbor set S has been generated for the target user u, the predicted ratings that user u give to any item can be calculate by the next formula.

$$P_{u,i} = \overline{R_u} + \frac{\sum_{v \in S} sim(u,v) \times \left(R_{v,i} - \overline{R_u}\right)}{\sum_{v \in S} \left(\left|sim(u,v)\right|\right)} \quad (5)$$

Among them, $\overline{R_u}$ is the average of ratings the user u gives, $sim(u,v)$ is the similarity weight between user u and user v, $R_{v,i}$ is the rating that user u has given to item i. Then choose the top - N elements from the item list in which the target user haven't rate and the predicted ratings are highest to recommend to the user.

4. Experiment and Result Analysis

4.1. Datasets

Experiences use the public data set provided by sites of Movie Lens, and its size is 100k. Movie Lens data sets were collected by the Group Lens Research Project at the University of Minnesota. This data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies and each user has rated at least 20 movies. Each film comprises at least one or more categories of 19 kinds of properties. Throughout experiment data is further divided into a training set and test set, the paper chooses 80% of the entire data set as a training set, 20% as the test set.

4.2. The Metric

MAE (mean absolute error) [8] is used as evaluation criteria and it is the common evaluation criteria for the common evaluation algorithm. Formally, if n is the number of actual ratings in an item set, then MAE is defined as the average absolute difference between the n pairs. Assume that p_1, p_2, p_3, ..., p_n is the prediction of users' ratings, and the corresponding real ratings data set of users is q_1, q_2, q_3, ..., q_4. See the MAE definition as follows:

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

The lower the MAE, the more accurate the predictions would be, allowing for better recommendations to be formulated. MAE has been computed for different prediction algorithms and for different levels of sparsity.

Recommendations from the long tail of the popularity distribution of items are valuable. On the other hand, recommendation accuracy tends to decrease towards the long tail items which often have fewer ratings and are more difficult to predict. In this paper, we use precision to measure recommendation accuracy, precision is defined as follows:

$$precision = \frac{|T \cap L|}{L} \quad (7)$$

where L is the recommendation list and T is the test set. The precision measures the ratio of common items in T and L to the recommendation list.

4.3. Comparing with the Traditional UBCF

In order to verify the effectiveness of our approach, we compare our BTCF algorithm with traditional UBCF algorithms. We use (4) and (6) to compute MAE in our BTCF and traditional UBCF. The number of neighbor we chose is 5. We adjusted parameter α in our experiments at first, figure 1 shows that the result is optimal when α is 0.6.

The number of neighbor increases from 5 to 30 in our experiments, assume α is 0.6, the result is shown in figure 2 and figure 3. We can find that the MAE of BTCF is smaller than UBCF's, the prediction is more accurate after alleviating sparsity, and our BTCF

performs better in experiments. And the result shows our BTCF is more accurate than traditional UBCF.

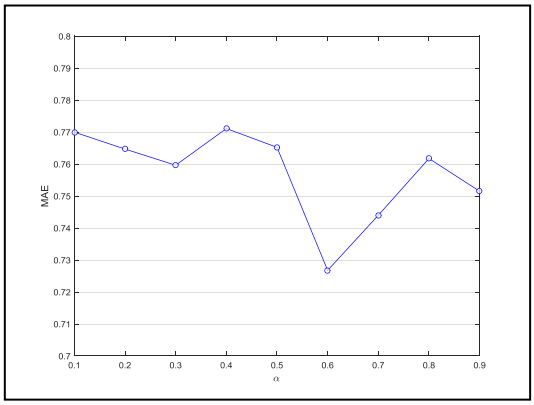

Figure1.The influence of α on MAE

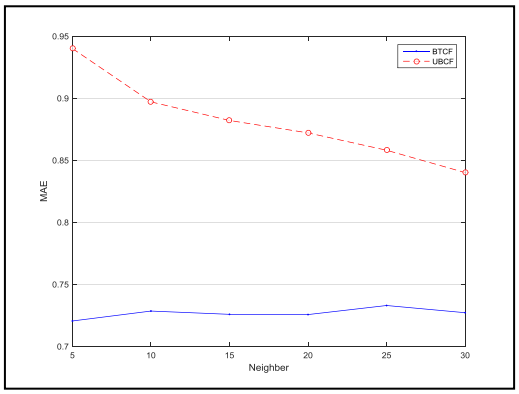

Figure 2.Neighbor on MAE

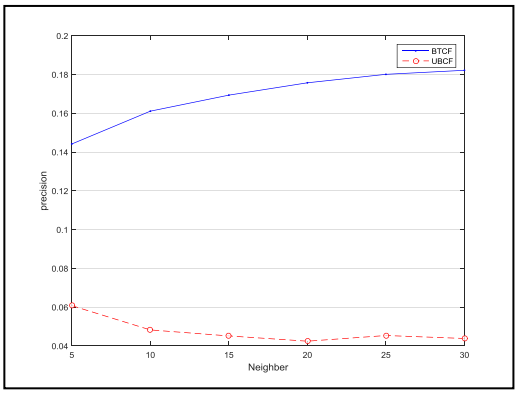

Figure 3. Neighbor on Precision

The number N in top-N module increases from 5 to 30 in our experiments, assume α is 0.6 and neighbor is 5, the result is shown in figure 4 and figure 5. We can also find that the MAE of BTCF is smaller than UBCF's, the prediction is more accurate.

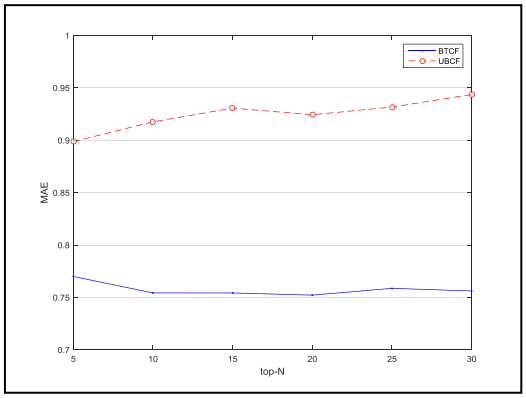

Figure 4. N on MAE

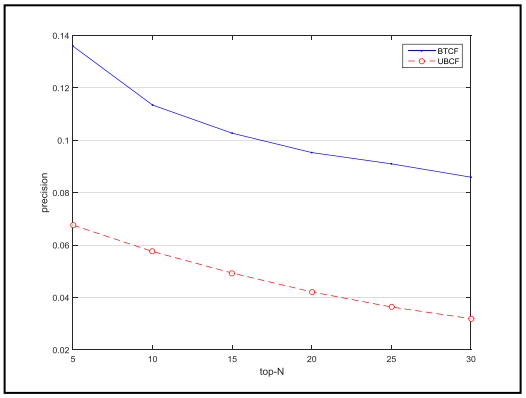

Figure 5. N on precision

5. Conclusion

In this paper, we present a new Collaborative Filtering algorithm using user-attribute rating and time characteristics. The results indicated that out method can find out users' interests on attributes more accurately by using statistical method. It produces recommendation results of a better quality.

Acknowledgments

As the research of the thesis is sponsored by National Natural Science Foundation of China (No: 61262075), major scientific research project of Guangxi higher education (No: 201201ZD012) and Project Foundation of Guangxi Experiment Center of Information Sciencep (No: 20130206), we would like to extend our sincere gratitude to them.

References

[1] H.-Y. Xu, W.-G. Du, Y. Feng, J.-M. Wang and M.-D. Liu, "A collaborative filtering algorithm based on multiple attribute score", Journal of liaoning university natural science edition, vol. 42, no. 2, **(2015)**.
[2] X.-Y. Yang, J. Yu, I. Turgun, Y.-R. Qian and H. Sun, "Collaborative filtering recommendation models considering item attributes", Journal of ComputerApplications, vol. 33, no. 11, **(2013)**, pp. 3062 -3066, 3106.
[3] L.-F. Sun and M.-X. Huang, "Collaborative filtering recommendation algorithm based on user characteristics and item attributes", Application Research of Computers, vol. 31, no. 2, **(2014)**, pp. 384-387.
[4] R. Lei, "A combined score of time characteristics of collaborative recommendation algorithm", Journal of computer applications and software, vol. 32, no. 5, **(2015)**.
[5] G. F. Sun, L. Wu, Q. Liu, C. Zhu and E. H. Chen, "Recommendations based on collaborative filtering by exploiting sequential behaviors", Ruan Jian Xue Bao/Journal of Software, vol. 24, no. 11, **(2013)**, pp. 2721-2733 (in Chinese).
[6] Z.-Y. Xiong, Q. Liu, Y.-F. Zhang and W.-T. Li, "Improved algorithm of collaborative filtering based on item classification", Application Research of Computers, vol. 29, no. 2, **(2012)**.
[7] P.-P. Yao, D.-S. Zou and B.-J. Niu, "Collaborative Filtering Recommendation Algorithm Based on User Preferences and Project Properties", Computer Systems & Applications, vol. 24, no. 7, **(2015)**, pp. 15-21.
[8] S.-H. Ding, D.-H. Ji and L.-I. Wang, "Collaborative filtering recommendation algorithm based on user attributes and score", Computer Engineer in Ganddesign, vol. 2, 36, no. 2, **(2015)**, pp. 487-491,497.

Authors

Xiaohui Cheng. Professor, College of Information Science and Engineering Guilin University of Technology. Current researches are on information processing and embedded system.

Yu Wu. Graduate Student, College of Information Science and Engineering, Guilin University of Technogy. Current researche is on information processing.

Yun Deng. College of Information Science and Engineering, Guilin University of Technology, Guilin, China. Current researches are on Internet of Things and embedded system.

