International Journal of Database Theory and Application
Vol.8, No.6 (2015), pp.243-252
http://dx.doi.org/10.14257/ijdta.2015.8.6.22

An Hybrid Similarity Function for Neighbor Selection in Collaborative Filtering

Shixiong Xia, Shaoda Chen and Zhixiao Wang*

College of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China
zhxwang@cumt.edu.cn

Abstract

Collaborative Filtering, which plays an important role in the recommendation, is based on the way that users rate the items throughout history. Although the CF recommendation system is used widely, the performance of recommendation still needs improving. In order to improve the accuracy of CF, the weight of item and the factor of time are considered in this paper. A new similarity method is proposed by improving the traditional similarity function with the weight of items. When putting the similarity of items and the factor of time as the weight of the target item, the neighbors of a user is not identical for all his items. Experimental results from MAE, precision, recall, f1 represent that the algorithm proposed can improve the performance of recommendation system.

***Keyword**: collaborative filtering; similarity calculation; weight of item; time factor*

1. Introduction

In recent years, due to the rapid development of the internet and information technology, there is more and more overloaded information. People try to solve the problem from various fields. Like personalized recommendation service, it finds users' potential interests by analyzing their browse history. Personalized recommendation has been widely used in e-commerce and e-library and has thus brought about huge economic history. A large number of algorithms have been proposed such as Content-based filtering (CBF), Demographic filtering, Collaborative filtering (CF) [2-4].

Collaborative filtering (CF) is a very popular technology in the information filtering and information systems. In order to predict the target user's interest, CF system store a wealth of information of users' item rating and make recommendation to the target users by comparing the information of similar users [1]. So how to find the similar users is very important to the CF system. For now, there are a lot of similarity calculation methods. The most commonly used of similarity algorithms are the Pearson correlation coefficient, cosine similarity, or distance-based similarity [5]. These similarity functions are used to calculate the similarity between the target user and the other user. In this process, the same weight is attached into items, leading that the neighbors of a target user is identical for all target items. But when the target item is comedy, users with a liking of comedy movies call for more attention than others. Meanwhile, for a user, the interest of items is not static, it changes over time. An item, which is rated 5 by a user one year ago, can be rated 1 by the same user now. Therefore, we should give the different value as the weight of item when calculating the similarity. In this paper, we consider the difference between the items and the factor of time at the same time. The similarity of items is taken into account to measure the difference between the items and a linear function of time is used as the representation of the time factor. When we put these two factors into the calculation of similarity between users, we can get more precise similarity of users, which results in more accurate final results.

In this paper, we proposed a collaborative filtering algorithm by putting the similarity of items and the factor of time as the weight of items to improve the recommendation System (RS) preference.

2. Related Work

Although CF-based recommendation system have been developed, but there are still many problems. A number of researchers have attempted to improve the quality of recommendation by addressing these problems. Jesús Bobadilla *et al.* [6] improve traditional similarities by taking contextual information, drawn from the entire body of users, and use it to calculate the singularity to improve the RS. Kyung-Yong Chung *et al.* [7] proposes a categorization for grouping associative items discovered by mining, for the purpose of improving the accuracy and performance of item-based collaborative filtering. Yi-Chung Hu [8] make further use of the preference relation to design a similarity measure by measuring the overall strength of one user’s preference over that of another. Keunho Choi and Yongmoo Suh [9] proposed a new similarity function in order to find different neighbors for each different target item. In their paper, they give different weight to each of the co-rated items rated by both users. SongJie Gong and GuangHua Cheng [10] believe that the current research on recommendation should pay attention on the use of time-related data in personal recommendation systems and proposed a methodology of probing user’s time span of their interest shift to improve the performance of systems.

3. Traditional Collaborative Filtering Based on Pearson

The main idea of CF is to predict the user’s interest for the corresponding items via the previous data of users. We define $I = \{i_1, i_2, i_3 \ldots i_m\}$ as the set of items and $U = \{u_1, u_2, u_3 \ldots u_n\}$ as the set of users. Traditional collaborative filtering consists of four parts:

(1)Construct a matrix about users and items from rating information;

(2)Calculate a target user’s similarity with other users by similarity function and select target’s neighbors;

(3)Predict the ratings of an item, on the basis of target users’ neighbors;

(4)Recommend top-rated n items.

3.1 .Build User-Rating Matrix

At first, set up the user-rating matrix. In the matrix, the column represents an item’s set of scores rated by all users and the row represents a user’s set of scores to items which he used to rate. The matrix is following:

$$\begin{bmatrix} R_{u_1,i_1} & R_{u_1,i_2} & . & R_{u_1,i_n} \\ R_{u_2,i_1} & R_{u_2,i_2} & . & R_{u_2,i_n} \\ . & . & . & . \\ R_{u_m,i_1} & R_{u_m,i_2} & . & R_{u_m,i_n} \end{bmatrix} \qquad (3\text{-}1)$$

$R_{u,i}$ is the rating of user u on item i. $R_{u,i} = \{0,1,2,3,4,5\}$, if the user u doesn’t rate on the item i, then the value of $R_{u,i}$ is 0.

3.2. Calculate the Similarity Based on Pearson

We choose the Pearson to calculate the similarity between two users.

$$Pearson(a,b) = \frac{\sum_{i=1}^{m}(R_{a,i} - \overline{R_a})(R_{b,i} - \overline{R_b})}{\sqrt{\sum_{i=1}^{m}(R_{a,i} - \overline{R_a})^2}\sqrt{\sum_{i=1}^{m}(R_{b,i} - \overline{R_b})^2}} \tag{3-2}$$

$Pearson(a,b)$ shows the similarity between user a and user b. $R_{a,i}$, $R_{b,i}$ represent that the rate of user a or user b to item i . $\overline{R_a}$, $\overline{R_b}$ are the average score of all items rated by user a or user b. The m in formula (1) is the number of items. $Pearson(a,b)$ is closer to 1,then the similarity between user a and user b is higher. On the contrary, the similarity will be lower.

3.3. Predict Rates about Target User

Now predict a user's rates to items by selecting a set of users with a high similarity calculated by Pearson with the target user. In CF recommendation system, using the following formula to predict the rates:

$$P_{a,i}^{predicted} = \overline{R_a} + \frac{\sum_{b=1}^{k} Usim(a,b) \times (R_{b,i} - \overline{R_b})}{\sum_{b=1}^{k} |Usim(a,b)|} \tag{3-3}$$

$P_{a,i}^{predicted}$ is the predicted rate made by the target user. k is the number of neighbors. $Usim(a,b)$ represents the similarity between user a and user b. $R_{b,i}$ is the item i 's scores rated by user b. $\overline{R_b}$ is the average score of all items rated by user b. The final rates of items about target user can be got by equation (3-2) and equation (3-3),and then recommend the top-n to the user.

4. Improved Similarity Calculation

In traditional collaborative filtering system, people put the equal weight with item when they calculate the similarity between two users by equation (3-1). The method is not accurate, because the weight about item is not simply given 1. However the weight of item should be considered from many aspects. Firstly, for selecting neighbors by one user, the set of users who share similarity with the target user should vary with different target items [9]. We should take this into consideration and justify the weight. In this paper, when calculating the similarity, we give more weight to items that are similar to a target item instead of giving equal weight to all items. Secondly, user's interest changes over time. There is a greater possibility of interest change with long intervals involved. That is, the time to select items will affect the similarity between users. So the closer the time is to now, the more likely it is to reflect the actual situation.

Therefore, according to these two aspects of the problems, this paper proposed a new method to calculate the similarity based on Pearson. The similarity of items and the factor of time are employed in the similarity function in this paper. The equation can be set up as follows:

$$sim(a,b)^{i} = \frac{\sum_{j=1}^{m} Isim(i,j)^{2} \times (W(a,j) \times R_{a,j} - \overline{R_{a}})}{\sqrt{\sum_{j=1}^{m} \{Isim(i,j) \times (W(a,j) \times R_{a,i} - \overline{R_{a}})\}^{\wedge}2}} \times \frac{\sum_{j=1}^{m} (W(b,j) \times R_{b,j} - \overline{R_{b}})}{\sqrt{\sum_{j=1}^{m} \{Isim(i,j) \times (W(b,j) \times R_{b,j} - \overline{R_{b}})\}^{\wedge}2}} \tag{4-1}$$

$sim(a,b)^{i}$ is the similarity between user a and user b when recommending item i. $Isim(i,j)$ is the similarity between item i and item j. $W(a,j)$, $W(b,j)$ is the weight of time. $R_{a,i}$, $R_{b,i}$ represent that the rate of user a or user b to item i. $\overline{R_{a}}$, $\overline{R_{b}}$ are the average score of all items rated by user a or user b.

In this function, when predicting the rate of an item, we consider the similarity of item and the weight of time, so the neighbor belong to the only item can be found, which increasing the prediction accuracy.

In equation (4-1), $Isim(i,j)$ represent the similarity of items. In this paper, we still use the Pearson as the similarity function.

$$Isim(i,j) = \frac{\sum_{a=1}^{n} (R_{a,i} - \overline{R_{i}})(R_{a,j} - \overline{R_{j}})}{\sqrt{\sum_{a=1}^{n} (R_{a,i} - \overline{R_{i}})^{2}} \times \sqrt{\sum_{a=1}^{n} (R_{a,j} - \overline{R_{j}})^{2}}} \tag{4-2}$$

The n is number of items. $R_{a,i}$, $R_{a,j}$ is the rate of item i or item j rated by user a . $\overline{R_{i}}$, $\overline{R_{j}}$ is the average rate of item i or item j .

User's interests will change over time. If the time that a user selected the item is closer to now, the interest is nearer to the real that means the weight of time is larger. On the contrary, the time is longer, the weight is less. So we use $W(u,i)$ as the weight of time. T_{u} is the interval time from the time that the user started to use the system to the latest time the user used the system. T_{ui} is the time span from initial attention to final decision. Finally the formula can be following:

$$W(u,i) = (1 - \alpha) - \alpha \frac{T_{ui}}{T_{u}} \tag{4-3}$$

In the equation(4-3),the α is a factor to control the weight of time. $\alpha \in (0,1)$. If the α is greater , the weight of time will be larger in the similarity calculation.

5. Experiment

In order to verify the effectiveness of the improved algorithm, the paper selects MovieLens datasets for testing. The data consists of over 100,000 rating (1-5) from 943 users on 1682movies, and each user has rated at least 20 movies and each movie is rated by at least one user. The 1-5 rating shows the degree of users' liking for movies. The higher the rating, the deeper the degree.

In this paper, we compare the performance of four approaches: 1) traditional Pearson algorithm, referred CFP; 2) the similarity with time weight, referred CFPT; 3) the similarity with item weight, referred CFPI; 4)the similarity that the paper proposed with time and item weight ,referred CFPIT.

5.1. The Parameter α

In equation (4-3), α is a control factor to adjust the weight of time in similarity calculation. To evaluate the effect of α , experiment use different value to compare rates that predicted by using parameter α and the actual rates. MAE is used to describe the difference in predicted rate and the actual rates. Figure 1 plots the trend of MAE when change the value of α from 0.1 to 0.9. As can be seen from the Figure 1, the MAE is less when the parameter is 0.5. So before the experiment, the value of α is set to 0.5.

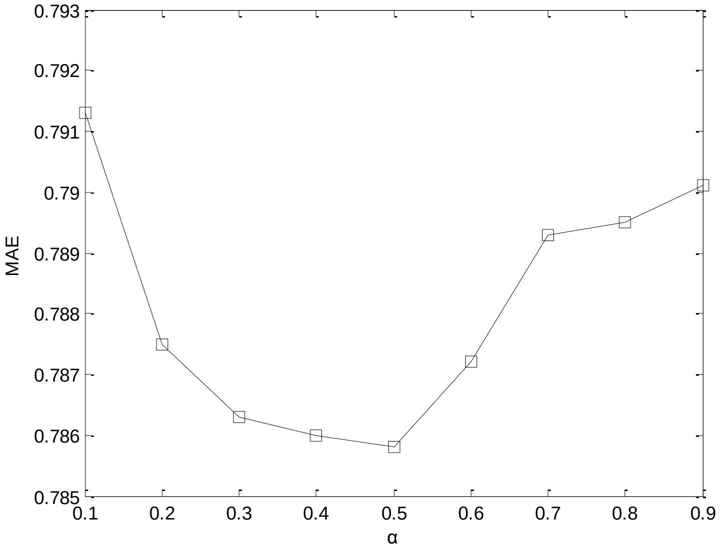

Figure 1.The Value of α

5.2. Evaluation

In order to measure the accuracy of algorithm the paper proposed, it is usual to use some evaluations such as the Mean Absolute Error (MAE), precision, recall and F1. MAE describes the difference between prediction and the real rate. Precision indicates the proportion of relevant recommended items from the total number of recommended items. Recall indicates the proportion of relevant recommended items form the number of relevant items. And F1 is a combination of precision and recall. We use the four evaluations to examine the improvement of our algorithm. In the following experiments, we change the number of neighbors from 5 to 25 s at the same evaluation to compare the effect of the four approaches.

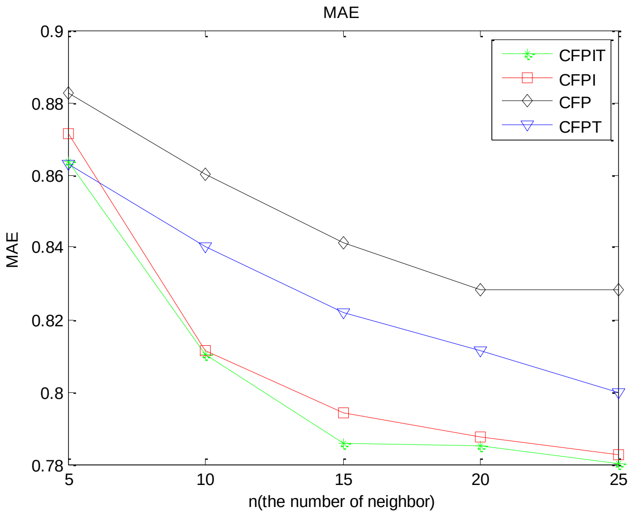

Figure 2. Comparison of Different Methods in MAE

In MAE, smaller value is better. Figure 2 shows the experimental results, the abscissa describe the number of neighbor and the ordinate represent the value of MAE. It's obviously that the value of MAE become small, along with the increasing number of neighbor. The algorithms of CFPIT and CFPI is always smaller than the rest no matter which value n is set. In addition, the effect of algorithm CFPIT is better compared with the CFPI. It can be seen, CFPIT get the lowest value when the number of neighbor is 15.

In precision, recall and f1, the value is bigger, and the result is better. In Figure 3, CFPIT and CFPI still have advantage in the precision comparing with the algorithm CFPT and CFP. With the increasing number of the neighbors, the precision obtained by CFPIT and CFPI are still higher and the precision obtained by CFPT and CFP drops when the n is more than 20. Certainly, we can see the value of CFPI is closer to CFPIT, even when the n is 15, the value is higher than the CFPIT. But from the Figure 3, the trend of the algorithm we proposed is gradually increasing.

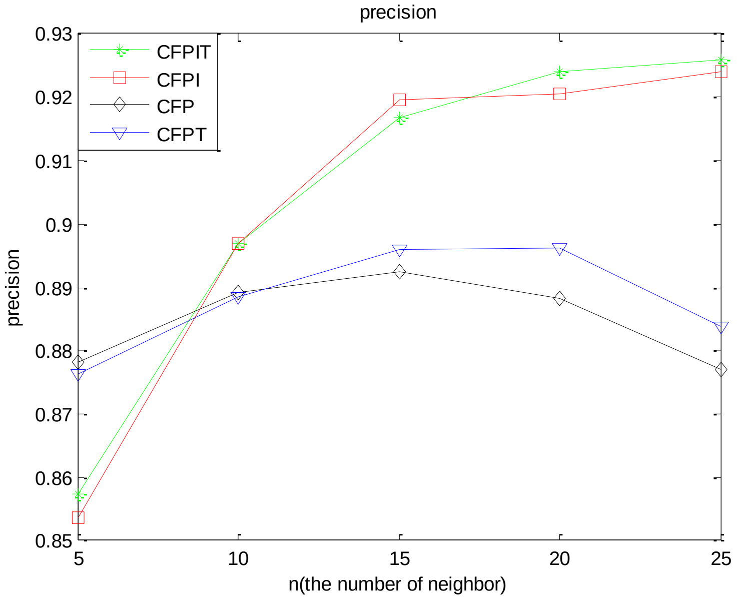

Figure 3. Comparison of different Methods in Precision

The Figure 4 is the result of the four approaches in recall. The Figure4 shows that the value of CFPIT, CFP, CFT is always higher and CFI (the traditional Pearson algorithm) begins to drop when the n is larger than 20. We can divide the Figure 4 into two parts according to the value of n. When the number n is less than 15, the value tends to grow quickly and CFPIT has the best effect in the four methods and the CFPI is followed by CFPI. When the number n is larger than 15, the value begin to grow slowly and CFPIT performs also the best in recall compared with the other algorithms. There is a great advantage of the algorithm proposed in this paper in Figure 4.

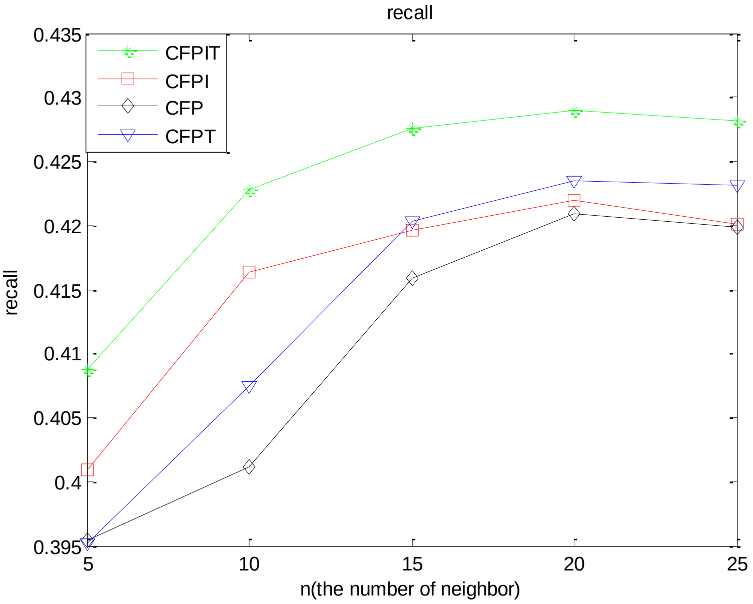

Figure 4. Comparison of different Methods in Recall

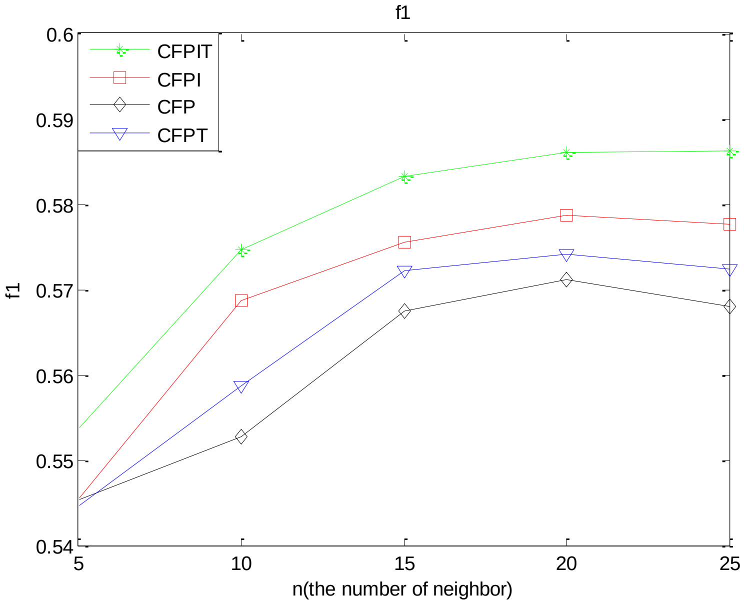

Figure 5. Comparison of different Methods in f1

Certainly, higher precision and higher recall represent a good performance of recommendation system. However, it is a contradiction that precision and recall increase at the same time. So f1 as combination of precision and recall should be used. The Figure

5 is the result of F1 with the four methods. From the figure, we can see the value of CFPIT is the highest in the four methods. Meanwhile, the value of F1 obtained by CFPIT becomes high along with the increasing number of neighbors. Through the comparison of MAE, precision, recall and f1, the CFPIT indeed improve the performance of the prediction accuracy.

6. Conclusion

With the development of society, recommendation systems are increasingly being used widely. It's important to improve the performance of recommendation system. Traditional similarity calculation method has been unable to meet the current requirements of the recommended system. We need to find out the more valuable information in a flood of information. In order to improve the accuracy of the RS, the information unused in the past should utilize now.

In this paper, we proposed a new similarity function considering the weight of item. We start from two aspects of item. At first, the similarity of target item with the other items is attached into the traditional algorithms, which leads to more accurate neighbors for every item of target user. In addition, we also see the influence of time when the users choose the item. Therefore, we view similarity of target items and time intervals as the weight of item. Experimental results suggest that this approach completed the information of item and improved the performance of RS apparently.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.61402482), the Fundamental Research Funds for the Central Universities (No.2014QNB23).

References

[1] J. Bobadilla, F. Ortega, A. Hernando and A. Gutiérrez, "Recommender systems survey.", Knowledge-Based Systems, vol. 46, **(2013)**, pp. 109-132.

[2] Thilagavathi N. and Taarika R.., "Content based filtering in online social network using inference algorithm", 2014 International Conference on Circuit, Power and Computing Technologies, Nagercoil, India, March 20-21, **(2014)**.

[3] Z. Shuai, Z. Yuan, W. Yan, Z. W. Yu and L. Yong, "A demographic-based and expertise-enhanced collaborative filtering method for e-government service recommendation", Journal of Computational information Systems, **(2014)**, pp. 2463-2480.

[4] D. T. Lien and N. D. Phuong, "Collaborative filtering with a graph-based similarity measure", 2014 International Conference on Computing, Management and Telecommunications, Da Nang, Vietnam, April 27-29, **(2014)**.

[5] Z. Y. F. Gao, H. Lv and Y. S. Xiong G., "The application of recommendation systems", Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics, Qingdao, China, October 8-10, **(2014)**.

[6] J. Bobadilla, F. Ortega and A. Hernando, "A collaborative filtering similarity measure based on singularities", Inf. Process. Manag. (UK) 204-17, **(2012)**.

[7] K. Y. Chung, D. Lee and Kim K. J., "Categorization for grouping associative items using data mining in item-based collaborative filtering", Multimedia Tools and Applications, **(2014)**, pp. 889-904.

[8] Hu Y. C., "Recommendation using neighborhood methods with preference-relation-based similarity", Information Sciences, vol. 284, **(2014)**, pp. 18-30.

[9] K. Choi and Y. Suh, "A new similarity function for selecting neighbors for each target item in collaborative filtering", Knowledge-Based Systems, vol. 37, no. 146, **(2013)**, pp. 53.

[10] S. J. Gong and G. H. Cheng, "Mining user interest change for improving collaborative filtering", 2008 second international symposium on intelligent information technology application, Shanghai, China, December 21-22, **(2008)**.

Authors

Shixiong Xia, He is the professor and doctoral tutor of China University of Mining and Technology. His researches are data mining, Intelligent Control and Industrial communication networks.

Shaoda Chen, He is working towards the master's degree in China University of Mining and Technology. His research is personal recommendation.

Zhixiao Wang, He is the associate Professor of China University of Mining and Technology. His researches are Community found and Data field theory.