UICF:A New Collaborative Filtering Algorithm for Recommendation

Jiawei Wu [a], Lingling Shen [b] and Gang Qian [c]

School of Nanjing Normal University, Nanjing 210097, China

[a]18912435780@163.com, [b]13770608879@163.com, [c]qgmail@vip.sina.com

Abstract

With the rapid development of Internet, the era of big data emerged and people are always confused when they face a large amount of data. The emergence of recommendation system matches people's need of data and collaborative filtering(CF) represents a widely adopted strategy today to build recommendation engines. But with the increasing number of users and project resources, traditional collaborative filtering algorithm exposed many problems such as data sparse and cold-start. In addition, it can not reflect the phenomenon that users' interests are changing over time. All these problems may reduce the accuracy of recommendation. In this paper we propose an user-item based collaborative filtering algorithm (UICF), a new approach for recommendation. Meanwhile, we introduce time weighting factor into our approach to catch the change of people's interests during a period of time. Experimental results show that our approach can improve the precision when we calculate user similarity and item similarity. And compared with traditional collaborative filtering algorithm, our approach have higher precision.

Keywords

Recommendation System, Collaborative Filtering, Time Weighting Factor, User Similarity, Item Similarity.

1. Introduction

Under the background of the era of big data, data breaks out at a staggering rate, such as WeChat, whose push volume has reached hundreds of millions a second. However, users are confused by the large amount of data every moment because it is hard for them to choose useful information from the vast data, and the problem is particularly prominent in e-commerce[1]. For example, on JD and Taobao and other e-commerce websites, wide variety of information of beautiful things and customer evaluations make users difficult to buy satisfying goods. So recommendation is particularly important.

A widely adopted approach to build recommendation engines is represented by collaborative filtering (CF) algorithms. Research efforts spent in the last several years on this topic yield some solutions Gemulla et al.(2011)[2]; Zhuang et al.(2013)[3]; Lin et al.(2014)[4]; Zhang et al.(2016)[5] that, as of today, provide accurate rating predictions, but vast computational costs may be incurred when large data sets emerge. This lack of efficiency is going to quickly limit the applicability of these solutions at the current rates of data production growth, and this motivates the need for further research in this field.

In this paper we propose an user - item collaborative filtering recommendation algorithm (UICF), and we introduce time weighting factor into our approach to catch the change of people's interests during a period of time.

The rest of this paper is organized as follows: Section 2 presents related work; Section 3 presents our approach; Section 4 is our experiments; finally Section 5 concludes the paper.

2. Related Work

2.1 Collaborative Filtering Recommendation Algorithm

Traditional collaborative filtering algorithm is divided into two categories: user-based collaborative filtering recommendation algorithm (UBCF) [6] and item-based collaborative filtering recommendation algorithm (IBCF)[7,8]. Collaborative filtering recommendation algorithm is based on the idea that we predict whether the target user is interested in the item according to his or her similar users' scores on this item. However, because there is a limited amount of information associated with users, their scores are not entirely related to items, which leads to the high sparse user - item rating matrix and can not fully reflect the relative relationship. So it increases the difficulty of choosing similar users and affects the efficiency of the recommended systems. Item-based collaborative filtering recommendation algorithm, based on the scores of the similar items, predicts the target user scores on ungraded items. However, when users' scores are rare, it easily leads to ignore items own attributes, thus reducing the efficiency of recommendation. With the era of big data coming, the rapid growth of the number of users and items resulted in the problems of cold start [7], sparse data [9] and so on. Besides, the traditional collaborative filtering algorithm cannot reflect the variation of users' interests during a period of time.

Researchers improved traditional collaborative filtering recommendation algorithm aimed at the problems it exposed under the background of big data. Due to the problems of cold start and sparse data, the literature [10] proposed a recursive prediction algorithm, which can filter the relatively close proximity of users when they don't evaluate the items. When users evaluate items faintly, the algorithm makes a recursion of predicting targets and integrates them into the corresponding prediction process, thus reducing the adverse effects of recommendation which are brought by sparse matrix. Literature [11] proposed a recommendation based on item category and item personalized context resource, which classifies target items into different item categories, and then introduces the context of personalized into them, so we can narrow the range of target items and similar items and enhance the accuracy of the prediction of target items. Literature [12] proposed a algorithm that combines user-based collaborative filtering algorithm with item-based collaborative filtering algorithm, where in the process of predicting user ratings, we introduce parameter λ to make linear combinations about the user-based collaborative filtering algorithm and item-based collaborative filtering algorithm, which can enhance the quality of the recommendation algorithm. For the problems that users' interests are changing over time affecting the quality of recommendation, the literature [13] proposed the concept of aging quantification, where gradually over time based on gradient descent, it noted for aging quantitative exponential function, so that the problem that the users' interest changing due to the time effects the quality of recommendation have been solved.

2.2 Limitations

Traditional collaborative filtering recommendation algorithm only attaches importance to the similarity between users or items. User-item collaborative filtering algorithm, due to the limited user contact information, can easily produce the problem that the user-item evaluation matrix is serious sparse and the problem of cold start. Item-based collaborative filtering algorithm exists the problem that the scores of users on items are rare and we may easily ignore the items own attributes. These two kinds of algorithms both ignore the phenomenon that users' interests are changing over time. So when a user's interest changes, recommended system can't update in time, which may reduce the efficiency of recommendation.

3. The UICF Algorithm

3.1 The User Interest Similarity Based on Time Weight

Users can mark the label to represent the preference for the item, and we can make the label set L(u,i) to represent the user u's preference to item i, so we can construct user-item preference matrix based on it, and calculate user similarity combined with time weight based on the matrix.

3.1.1 User's Interests in Items

In general, a tag, used by the user frequently, represents high important degree, so we define user u prefer to label l as follows:

$$Pre(u,l_k)=\frac{Freq(u,l_k)}{\sum_{l_k\in L(u)}Freq(u,l_n)} \tag{1}$$

where L(u) represents the tag set which is used by users, Freq(u,lk) represents the frequency of use u using label l.

At this point, the user u's preference for item i can be represented as:

$$Pre(u,i)=\sum_{t_k\in L(u,i)}\mathrm{Pr}\,e(u,l_k) \tag{2}$$

where L(u,i) represents the tag set of user u making labels on item i.

3.1.2 Users' Interest Similarity Calculation Based on Time Weight

Traditional collaborative filtering recommendation algorithm does not take into account that the users' interests are changing over time, which reduces the recommendation quality to some extent. This section uses the user-item evaluation matrix mentioned above in section 2.1, considering the time when users made labels on items, to calculate user similarity. The basic idea is: the user's interests in recent items should be highlighted and the user's interests in past items should have smaller weight. Therefore, we define the user similarity between u and v by formula(3):

$$sim(u,v)_{time}=\frac{\sum_{i\in I}(r_{u,i}-\overline{r_u})(r_{v,i}-\overline{r_v})\,f_{time}(t_{u,i},t_{v,i})}{\sqrt{\sum_{i\in I}(r_{u,i}-\overline{r_u})^2}\sqrt{\sum_{i\in I}(r_{v,i}-\overline{r_v})^2}} \tag{3}$$

where $I=I_u\cap I_v$ represents the subset of services items, which both user u and user v have invoked previously, $r_{u,i}$ is a vector of scores of item i evaluated by user u, $\overline{r_u}$ and $\overline{r_v}$ represent average scores of different items evaluated by user u and v respectively. $f_{time}(t_{u,i},t_{v,i})$ represents the decay of user u's time weight on item i and user v's time weight on item i.

$$f_{time}(t_{u,i},t_{v,i})=\exp\{-(1+\alpha\times(t_{u,i}-t_{v,i}))\} \tag{4}$$

where α represents attenuation factor.

3.2 Item Similarity Based on Time Weight

3.2.1 User Interests Weighted Function Based on Time

We define I_u as the item set user u have invoked. By defining a time window T to get the I_{uT} user u have invoked in recent T period. So I_{uT} ,to some extent, reflects the user's recent interests. Whenever user u invoked item i, if there are many items have high similarity with item i, that is to say item i is associated with user u's recent interests. So during a period of time in the future, the items, which are interested in by user u, may be similar to item i. Then we define user interests weighted function based on time as follows:

$$f(u,i) = \frac{\sum_{j \in I_{uT}} sim(i,j)}{|I_{uT}|} \quad (5)$$

where $|I_{uT}|$ represents the number of IuT.By changing the length of the time window T, we can get I_{uT} of different recent periods of time , which can affect the quality of recommendation.

3.2.2 Calculating Item Similarity Based on Time Weight

When we calculate item similarity, considering the user's interests changing over time, we can introduce time weight into it, so we compute similarity between users u and v by formula(6):

$$sim(i,j)_{time} = \frac{\sum_{u \in U}(f(u,i) \times (r_{u,i} - \overline{r_i}))(f(u,j) \times (r_{u,j} - \overline{r_j}))}{\sqrt{\sum_{u \in U}(f(u,i) \times (r_{u,i} - \overline{r_i}))^2}\sqrt{\sum_{u \in U}(f(u,j) \times (r_{u,j} - \overline{r_j}))^2}} \quad (6)$$

where $U = U_i \cap U_j$ is the subset of users who have invoked both item i and item j previously, $\overline{r_i}$ and $\overline{r_j}$ represents the average scores of the items i and j evaluated by different users.

3.3 Prediction of Ungraded Items

User-based collaborative filtering approach applies similar users to predict the scores by formula (7):

$$r_u(u,i) = \overline{r}(u) + \frac{\sum_{v \in S(u)} sim_{time}(u,v)(r_{v,i} - \overline{r}(v))}{\sum_{v \in S(u)} sim_{time}(u,v)} \quad (7)$$

In which $\overline{r}(u)$ denotes average scores of user u, and $\overline{r}(v)$ denotes average scores of user v. $S(u)$ denotes the set of similar neighbors of user u:

$$S(u) = \{v \mid sim(u,v) > 0, v \neq u\} \quad (8)$$

Similar to the user-based approaches, item-based collaborative filtering approach applies similar items to predict the scores by formula (9):

$$r_i(u,i) = \overline{r}(i) + \frac{\sum_{j \in S(i)} sim_{time}(i,j)(r(u,j) - \overline{r}(j))}{\sum_{j \in S(i)} sim_{time}(i,j)} \quad (9)$$

In which $\overline{r}(i)$ denotes average scores of item i, and $\overline{r}(j)$ denotes average scores of item j. $S(i)$ denotes the set of similar neighbors of item i:

$$S(i) = \{j \mid sim(i,j) > 0, j \neq i\} \quad (10)$$

Since these two predicted scores may have different prediction performance, we utilize a tunable parameter λ to balance these two predicted scores. We integrate the two approaches by formula (11):

$$r(u,i) = \lambda r_u(u,i) + (1-\lambda) r_i(u,i) \quad (11)$$

Finally, this approach makes recommendation according to the scores.

4. Experiments and Analysis

4.1 Data Set Description

We adopt MovieLens [14] data sets to evaluate our algorithm. Experiments use the training set for predicting scores on the ungraded films and we compare them with the test set at the same time.

Experimental training set contains 952 users to 1596 films of 90000 score records (score value is an integer from 1 to 5), experimental test set contains 478 users of 30000 score records (score value is an integer from 1 to 5).

This experiment of sparse degree of user ratings is as follows:

$$1-\frac{90000}{952\times 1596}\times 100\% = 94.08\%$$

Therefore, the sparse degree of user ratings is 94.08%, so the score data is very sparse.

4.2 Metrics

We utilize the mean absolute error (MAE) as a standard of measurement [15]. MAE measures the accuracy of prediction through calculating the deviation between the predicted scores and actual scores from users. And the smaller MAE value represents the higher quality. The test set contains N score data, represented as $\{q_1, q_2, ..., q_n\}$, and the training set contains the same number score data, represented as $\{p_1, p_2, ..., p_n\}$.The MAE can be calculated by formula (12):

$$\text{MAE}=\frac{\sum_{i=1}^{N}\left|p_i-q_i\right|}{N} \tag{12}$$

The smaller MAE value represents the higher accuracy of prediction.

4.3 Experimental Results

In this section, we verify the accuracy of our algorithm through five experiments respectively.

Experiment 1: Impact of Parameter α

As the attenuation factor, α has an important position. The improper values of α will cause the time weight value f_{time} too large or too small, which affects the quality of recommendation. In this experiment, we change the values of parameter α from 0 to 1 with a step of 0.1. We set the training user number to 200, λ to 0.3, T to 10 and S to 50.

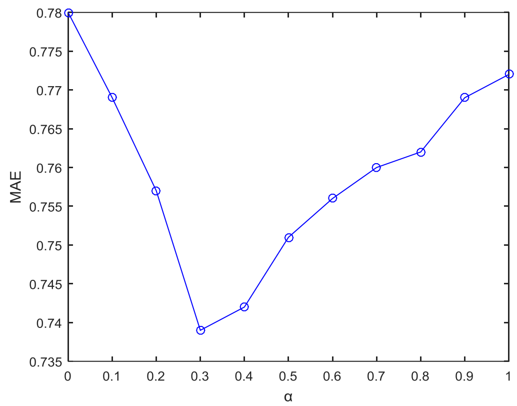

Fig1.Impact of parameter α

The smaller MAE value represents the higher accuracy of prediction. Figure 1 shows the impacts of parameter α on the results. We observe that optimal α value settings can achieve better prediction accuracy. As α increases, the MAE value reaches a minimum, and then increases.

Experiment 2: Impact of Parameter λ

In our approach, parameter λ controls how much our approach relies on UBCF and IBCF. If λ=0, we only utilize IBCF for make prediction. If λ=1, we only utilize UBCF for make prediction. In other cases, we integrate UBCF and IBCF for value prediction. In this experiment, we change the values of parameter λ from 0 to 1 with a step of 0.1. We set the training user number to 200, α to 0.3, T to 10 and S to 50.

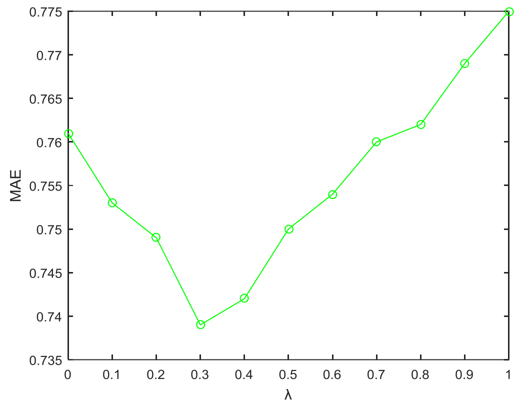

Fig2.Impact of parameter λ

Figure 2 shows the impacts of parameter λ on the results. We observe that optimal λ value settings can achieve better prediction accuracy. As λ increases, the MAE value reaches a minimum, and then increases.

Experiment 3: Impact of Parameter S

Parameter S determines the number of similar users to the user u or similar items to the item i in our approach. To study the impact of parameter S, we change the values of parameter S from 5 to 50 with a step of 5. We set the training user number to 200, α to 0.3, T to 10 and λ to 0.3.

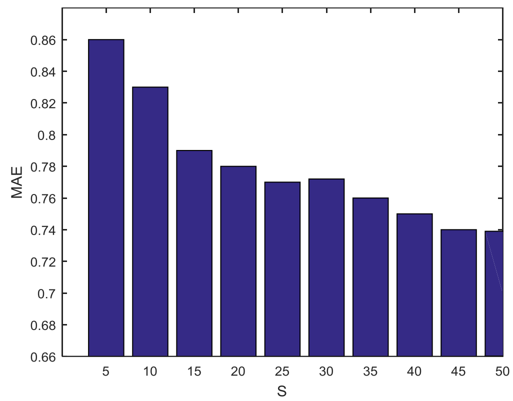

Fig3.Impact of parameter S

Figure 3 shows the impacts of parameter S on the results. With the increase of number of neighbor S value, MAE value shows the tendency of decrease. When the number of neighbor S value increases, MAE value levels off, and the highest accuracy of prediction of this algorithm appears as the S value is 50.

Experiment 4: Impact of Parameter T

Users' interests may change over time. To analyze how the value of time window T affects MAE value, we change the values of parameter T from 5 to 30 with a step of 5. We set the training user number to 200, α to 0.3, S to 10 and λ to 0.3.

Figure 4 shows the impacts of parameter *T* on the results. As *T* increases, the MAE value reaches a minimum, and then increases. This suggests that the length of time window T has a certain influence on accuracy of prediction, and longtime cannot reflect users' current interests.

Experiment 5: Impact of Collaborative Prediction

There are different approaches to collaborative prediction for fulling in missing values in matrix (e.g., UBCF [6], IBCF [7, 8], etc.).

To compare the influence of different approaches, we implement three versions of our algorithm, using the UBCF, IBCF and our approach, respectively. We change the values of parameter S from 10 to 80 with a step of 10.

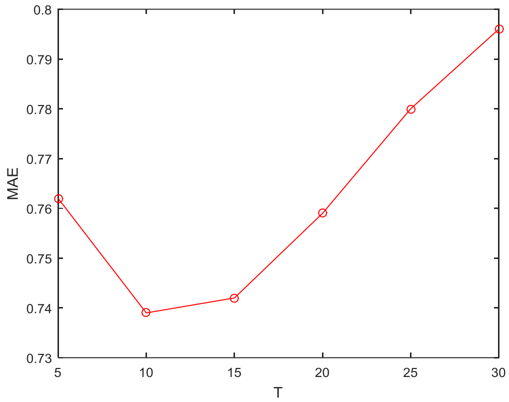

Fig4.Impact of parameter T

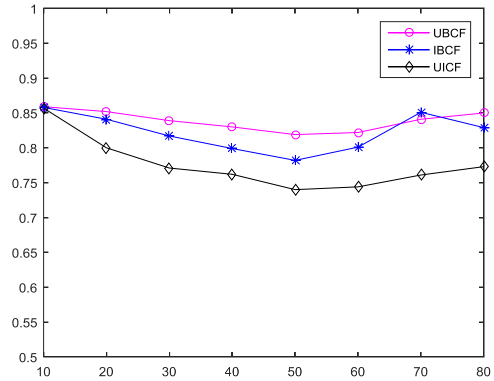

Fig5.Impact of Collaborative Prediction

Figure 5 shows that:

1.As *S* increases, the MAE value reaches a minimum, and then increases.

2.The accuracy of our approach outperforms the approaches (UBCF, IBCF). This observation indicates that it was better to consider more information for prediction.

5. Conclusion

We propose UICF to the prediction of scores for project sources on the past service usage experiences, which combines UBCF with IBCF and introduces time weighting factor to catch the change of users' interests during a period of time. The proposed approach can achieve higher prediction accuracy. The results of extensive experiments show the effectiveness of our approach.

The UICF approach in this study can be utilized to serve the business process of cloud application, where some of components invoke other cloud services to outsource part of business to other companies. And we plan to explore how to enhance extendibility of UICF algorithm in the future.

References

[1]Schafer J B, Konstan J, Riedl J. Recommender systems in e-commerce. Proceedings of the 1st ACM conference on Electronic commerce. ACM, 1999, p. 158-166.

[2]Gemulla R, Nijkamp E, Haas P J, et al. Large-scale matrix factorization with distributed stochastic gradient descent. NIPS Big Learn Workshop. Vol. 30(2011), p. 69-77.

[3]Zhuang Y, Chin W S, Juan Y C, et al. A fast parallel SGD for matrix factorization in shared memory systems. ACM Conference on Recommender Systems. ACM, 2013, p. 249-256.

[4]Lin H, Yang X, Wang W, et al. A Performance Weighted Collaborative Filtering algorithm for personalized radiology education. Journal of Biomedical Informatics. Vol. 51(2014), p. 107-113.

[5]Zhang J, Lin Y, Lin M, et al. An effective collaborative filtering algorithm based on user preference clustering. Applied Intelligence. (2016), p. 1-11.
[6]Zhao Z D, Shang M. User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop. International Conference on Knowledge Discovery and Data Mining, Wkdd 2010, Phuket, Thailand, 9-10 January. 2010, p. 478-481.
[7]Deng A L, Zhu Y Y, Shi B L. Collaborative filtering recommendation algorithm based on project score predicts. Journal of Software. Vol. 14 (2003) No. 9, p. 1621-1628.
[8]Luo Y, Yu Y, Zhao C L, et al. Recommended adaptive algorithm in the application of electronic supermarket personalized service system research. Journal of Communication. Vol. 27 (2006) No. 11, p. 183-186.
[9]Tian W, Peng Y Q. The collaborative filtering technology based on electronic commerce application improvement were reviewed. Engineering and Computer Science.Vol. 30 (2008) No. 10, p. 61-63.
[10]Zhang J, Pu P. A recursive prediction algorithm for collaborative filtering recommender systems. ACM Conference on Recommender Systems, Recsys 2007, Minneapolis, Mn, Usa, October. 2007, p. 57-64.
[11]Yang C, Li H. Based on the individual situation and categories of resources recommended research project. Computer Science. s1(2011) , p.175-177.
[12]Xiong W, Li B, Cui X. A Learning Approach to the Prediction of Reliability Ranking for Web Services. 2015 IEEE International Conference on Web Services. IEEE, 2015, p. 169-176.
[13]Zhang Y, Liu Y. A Collaborative Filtering Algorithm Based on Time Period Partition. International Symposium on Intelligent Information Technology & Security Informatics. IEEE Computer Society, 2010, p. 777-780.
[14]Kim B M, Li Q, Park C S, et al. A new approach for combining content-based and collaborative filters. Journal of Intelligent Information Systems. Vol. 27 (2006) No. 1, p. 79-91.
[15]Yang X, Guo Y, Liu Y, et al. A survey of collaborative filtering based social recommended systems. Computer Communications.Vol. 41 (2013) No. 5, p. 1-10.