

An Improved Collaborative Filtering Recommendation Method Based on Timestamp

Yancui Shi

School of Computer Science and Information Engineering, Tianjin University of Science and Technology, Tianjin 300222, china

Email: syc@tust.edu.cn

Abstract—The collaborative filtering is the traditional method in the recommender system. In order to improve the accuracy of recommender system, an improved collaborative filtering recommendation method based on the timestamp is proposed in this paper. Since the preferences of the user towards to items are a time series, so they are changing. Therefore, we choose user ratings that are related with the current user ratings through the sliding window technology. To accurately calculate the similarity between users, the timestamp information is employed to pre-filter the user preference. When recommending items for user, the timestamp is used again to filter the items that may be used by the target user. In this paper, the effectiveness of the proposed method is verified by Filmtipset datasets.

Index Terms—collaborative filtering, similarity, sliding window, timestamp

I. INTRODUCTION

Recommender system has been applied in many fields, such as E-commerce, social network or personalize service. The most familiar applications are item or friend recommendation in Amazon, taobao, facebook, pengyou.com and so on. In these applications, the common recommendation methods are organized into three categories: content-based, collaborative filtering and the hybrid. In this paper, the focus is about how to improve the collaborative filtering algorithm according to the timestamp.

A. About the Application of the Timestamp

The idea of collaborative filtering is to predict the item that the target user most interest according to the preferences of the users whose hobbies are similar with the target user. The key thing in the method is how to find the most fitting similar users to enhance the accuracy of recommender system. In other words, it is how to find the set of nearest neighbors. In order to resolve the problem, the researchers do a large number of studies. Some researches handle the problem by alleviating the sparsity problem. For example, some researches reduced the dimensions of user-item matrix to alleviate the sparsity by filtering [1] or SVD [2, 3].

From a different perspective, some researches resolve the problem by introducing the time information. In this paper, we utilize the timestamp information to improve the performance of recommender system. The timestamp information can be divided into different purposes [4]: (1) some researches only regarded the time as context, such

as morning, evening or weekend [5,6,7]; (2) the preferences of the user will change over time and the influence of the rating also will reduce, so some researches employed the aging factor of the rating in the process of finding nearest neighbors [8,9,10]. The impact of the recent rating is greater than the old rating.

B. The Improved Collaborative Filtering Approach based the Timestamp

The new users and new items will join in the recommender system constantly over time and the preferences of the existing users will change. Taking into account the problem, Neal et al [9] set the window size and predicted the rating of the target user to the given item regularly. Similarly, Wen et al [10] employed the dynamic item clustering to improve the performance of the collaborative filtering. Nachiketa et al [11] improved the collaborative filtering by tracking the individual user. Ernesto et al [12] proposed RMFX, and the novel approach followed a selective sampling strategy to perform online model updates the task of identifying relevant items from a pool of mostly uninteresting ones.

Above researches mainly improved the performance of the recommender system by regularly updating or on-line learning. They scarcely considered the aging of ratings. From the fresh-based perspective, Kostas et al [4] proposed using different aging schemes to decrease the effect of historical ratings and increase the effect of the fresh and novel ratings. The different aging themes included the damped window mode and the sliding window model. In the damped window model, the exponential fading function was employed to weight the historical ratings, while the sliding window model is similar with the method that was proposed in [9]. Li et al [13] took the time factor into account and regarded the users' interest as the memory process, so proposed a time weight iteration model based the memory principle. Li also verified the exponential function is superior to the logistic function.

However, the existing researches ignored the sequence of the ratings. The conservative-minded user always selected the item after the other users used the item. For example, user A always rated item after the user B, but before the user C, so it only needed to compute the similarity between user A and user B when predicting the preference of user A. If in process of finding the nearest neighbors, this problem was not consider, it brought some

negative effect, such long running time and low accuracy. Therefore, it is necessary to select the appropriate users when computing the similarity. It can not only enhance the accurate of recommender system, but also can shorten the running time.

We conducted some preliminary study and thought that the current preferences of the given user are similar with the preferences of another user in certain period of time [14]. Therefore, we selected the items and ratings that happened before the corresponding items of ratings of the target user.

In this paper, we do the further study. The rest of this paper is organized as follows: Section II describes the proposed method in detail; in the section III, we verify the effectiveness of the proposed method by the real dataset. Finally the conclusion and future work is given in the section IV.

II. THE IMPROVED COLLABORATIVE FILTERING METHOD

In the section, the improved collaborative filtering method will be elaborated. We employ the sliding window to select the fitting items and ratings firstly. Secondly, the pre-filtering technology is used to select the set of the users that need to calculate the similarity with the target user. And then, when calculating the similarity, the time decay factor is considered. Finally, the after-filtering technology is employed to select which items need to be predicted the ratings.

A. The Sliding Window

The recommender system is a dynamic system. In the system, the new items and the users will join in and some outdated items will be removed. Therefore, the data in the recommender system can be regarded as data stream and the data will increase continuously. Obviously, it's infeasible to do research in the data stream due to the gigantic computation. It is the most apposite method to employ the sliding window to choose the useful data [4].

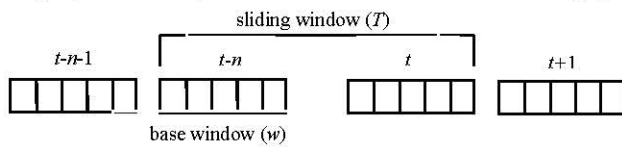


Figure 1. The sliding window

B. The Pre-filtering Technology

The preferences of user are not only related with his own nature, but also may be affected by his friends, family, colleague, classmate or celebrity. For example, user A always watches movies that user B (user A's friend) has watched and the ratings are relatively high. If we grasp this principle, we can recommend the movies that user B has watched to user A. The existing methods mostly find the nearest neighbors by the similarity between users. However, the similarity has the symmetry. In the above example, $sim(A,B)=sim(B,A)$, according to the exiting methods, it also recommends the movies that user A to user B. Evidently, it is illogical and even brings negative effect to the performance of recommender

system. Therefore, it is necessary to consider the sequence of ratings.

Definition 1: Stylish User. It is that these users who like following the trend and are easier to accept the fresh item. When the new items are launched, the stylish user will use them firstly.

$$T_{u_i} = \frac{\sum_{k=0}^{N_{new}-1} t_{u_i, s_k} - t_{s_k}}{N_n} \quad (1)$$

Where, $N_{new}=|S_{new}|$ represents the number of new items that are launched, and S_{new} represents the set of new items. t_{u_i, s_k} is the timestamp of the ratings and t_{s_k} is the time that the new item was launched. If $T_{u_i} < T_s$, the user i was regarded as a stylish user. T_s is a threshold, in different application, the value is different. For example, in movie, T_s is not more than two weeks.

Definition 2: The Follower. In recommender system, some users will use the new items firstly due some reasons, such as stylish user. These users are regarded as followed users. While the other users will choose the item according to the comment or the opinions of the users that have used the item. These users are regarded as followers. Note that a user may be both a follower and followed user. For example, friends can influence each other. On the one hand, the user will be influenced by his friend. On the other hand, he also will influence his friend.

Definition 3: The Following Time. $T_{i,j,k}$ represents the interval that the user i and user j rate the same item k . $T_{i,j,k}=t_{i,k}-t_{j,k}$, where, $t_{i,k}$ represents the timestamp that the user i rates the item k . We can judge whether there is a following relationship between two users according to the interval $T_{i,j,k}$.

The pre-filtering method is as follows. We take the user i for example, that is u_i .

Step 1. Selecting the common ratings of u_i and u_j , and $S_{i,j}$ represents the set of the common ratings. If $|S_{i,j}| > N_c$, go to Step 2. N_c represents the threshold of the common ratings.

Step 2. Calculating the interval, $T_{i,j,k}$. if $0 < T_{i,j,k} < T_{max}$, add k into the set P_i , else if $0 < |T_{i,j,k}| < T_{max}$, add k into the set P_j . T_{max} is the threshold of the interval, and it is set by the many experiments.

Step 3. Determining the set of the followed users U_f . If $|P_i| > 1/2 * N_c$, add u_j into the set U_f . The set of P_j can be used when recommending item to u_j .

C. Calculating the Similarity

When calculating the similarity, we employ the exponential function to reflect the time decay. The Pearson correlation coefficient is chose to measure the similarity between u_i and u_j . The formula is as follows:

$$Sim(u_i, u_j)_t = \frac{\sum_{s_k \in P_i} (r_{u_i, s_k, t} - \bar{r}_{u_i, t})(r_{u_j, s_k, t} - \bar{r}_{u_j, t})}{\sqrt{\sum_{s_k \in P_i} (r_{u_i, s_k, t} - \bar{r}_{u_i, t})^2} \sqrt{\sum_{s_k \in P_j} (r_{u_j, s_k, t} - \bar{r}_{u_j, t})^2}} \quad (2)$$

Where, $u_j \in U_f$, s_k represents the item k and $r_{u_i, s_k, t}$ represents the rating that introduces the decay factor and it is as follows:

$$r_{u_i, s_k, t} = r_{u_i, s_k, t_1} * f(t_1, k) \quad (3)$$

Where, t_1 is the timestamp of ratings of u_i to s_k . the time decay is as follows:

$$f(t_1, k) = e^{-k(t-t_1)} \quad (4)$$

Where, k is employed to adjust the decay rate.

According to the similarity, we can find the K nearest neighbors of the target user.

D. The After-filtering

After getting the set of nearest neighbors, the existing methods will predict the ratings of the items that the target user will use possibly in future. However, these methods seldom study the set of items that need to predict the ratings. Existing methods think that all items that the nearest neighbors have used but the target user don't use yet need to be predicted. But in recommender system, the user may rate many items, so it needs to predict amount of ratings according to the thought of the exiting methods. It not only increases the running time, but also may bring some negative impact.

TABLE I. THE ITEMS AND THE RATINGS

item user	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
u_1	4		5	3	1				
u_2		3	5	3	1		1		4
u_3	4		5		1	2		3	

In the table I, according to the traditional method, the $sim(u_1, u_2) = sim(u_1, u_3)$, so u_2 and u_3 are both the nearest neighbors of the u_1 . The items that need to predict ratings include s_6, s_7, s_8, s_9 . But, in fact, the u_1 only uses the s_7 . How to select the more accurate set of items is a challenge.

In this paper, we still employ the interval to select the set of items that need to be predicted. The after-filtering method is follows.

Step 1. Initially identifying the set of items that the target user didn't use yet but the nearest neighbors have used to time t . And S_i represents the set of items.

Step 2. Selecting the set of the items that the target user most likely uses. If $T_{i,j,k} < T_{max}$, add the item k into the set S_f where item $k \in S_i$, the user $j \in U_f$.

E. Predict the Ratings

There is a problem in the recommender system: the cold-start, which is caused by the new items and the new users. In this paper, we employ the timestamp to process the cold start problem caused by the new item.

Firstly, we judge whether the user is a stylish user or not. If the user is stylish user, we employ the item-based method to predict the ratings to the new item. The formula to predict the ratings is as follows:

$$r_{u_i, s_k, t} = \bar{r}_{u_i, t} + \frac{\sum_{s_l \in S_f} sim(s_l, s_k) * (r_{u_i, s_l} - \bar{r}_{u_i, t})}{N_f} \quad (5)$$

Where, $sim(s_l, s_k)$ is the similarity and we can get it according to the attributes of the item. For example, in movies, the attributes include the type, length, actor, director and so on. S_f represents the set of the items that are similar with the new item s_k , $N_f = |S_f|$.

We can utilize the following formula to process the cold start problem caused by the new user:

$$r_{u_i, s_k, t} = \frac{\sum_{u_q \in U_Q} r_{u_q, s_k, t}}{|U_Q|} \quad (6)$$

Where, U_Q represents the set of users that have used the item s_k .

The common prediction is as follows:

$$r_{u_i, s_k, t} = \bar{r}_{u_i, t} + \frac{\sum_{u_j \in U_n} sim(u_i, u_j) * (r_{u_j, s_k} - \bar{r}_{u_j, t})}{\sum_{u_j \in U_n} sim(u_i, u_j)_t} \quad (7)$$

Where, $s_k \in S_f$, U_n represents the set of the nearest neighbors of u_i .

F. The Proposed Method

From the above analysis, the proposed method in this paper is as follows:

Step 1. We employ the sliding window to select the dataset that need learn. In the sliding window, the new data will join in and the old data will be removed over time. In order to avoid frequently updating, we use the base window technology. Until the new data fills in a new base window, the learning is triggered.

Step 2. Determining the set of U_f by using the pre-filtering technology.

Step 3. Introducing the time decay factor and calculating the similarity between users.

Step 4. Determining the set of S_f by using the after-filtering technology.

Step 5. Predicting the ratings for different situations.

Step 6. According to the predicted ratings, selecting the top- N items.

III. EXPERIMENTS AND RESULTS

In the section, we describe the experiments and analyse the results.

A. The Dataset

Filmtipset is the Switzerland's largest dataset for social movie recommendation, which has more than 90,000 registered users and more than twenty million film ratings. The dataset includes the timestamp of ratings and the attribute information, such as year, type and so on.

B. Experiment Setup

In order to verify the effectiveness of the proposed method, we employ the traditional collaborative filtering method as the bench method. Here, the value of K that is

the number of the nearest neighbors is set according to the conclusion in [14].

(1) Firstly, determine the sizes of the sliding window and base window. We set the size of the sliding window respectively 4,8,12 months, and the size of the base window is set 0.5, 1, 1.5 months.

(2) Secondly, set the threshold of interval, T_{max} . T_{max} is set 10,20,30,40,50 days. The threshold of the common items, N_c , is set according to the method in [6].

(3) And then, set the time decay parameter, k . k is set 0.001, 0.005, 0.01. Here, the unit of time is day.

(4) Finally, determine the threshold of interval that is employed to judge whether the user is a stylish user, T_s . T_s is set 5,10,15,21 days.

We employ the MAE, Precision to evaluate the performance of the proposed method. These evaluation methods are described in the [14].

C. Experiment Results and Analysis

(1) The size of the sliding window

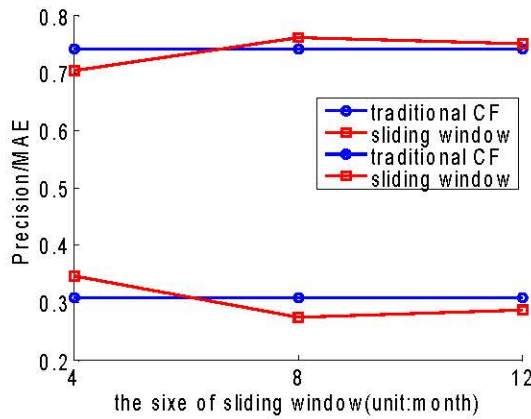


Figure 2. The results based on different sliding windows

From the Figure 2, we can know that when the size of the sliding window is set 8 months, the results are the best. When the size increase, the accuracy will reduce. When the size trends to the infinity, the accuracy trends to the accuracy that is get by the traditional CF. When the size is set 4 months, the results are worse than that got by the traditional CF.

The reason is that when the size is relatively small, the items and the ratings are few, so it can't compute the similarity accurately. When the size increases, it may bring the outdated data, while the size of traditional CF can be regarded as infinity, so the results still superior to the results that are got by the traditional CF and will trend to it gradually.

In the following experiments, the size of the sliding window is set 8 months and the size of the base window is set 1 month.

(2) The threshold of interval: T_{max}

The Figure 3 shows that it is feasible to introduce the pre-filtering technology into collaborative filtering. And when the T_{max} is set 30 days, the accuracy is the best.

This is because that when the value of T_{max} is too small, the limit to the interval is too strict. Therefore, the users who meet the condition are few, and the results are not

ideal. When the value of T_{max} is too big, it loses the meaning, so the result will trend to that obtained by the method that does not use the pre-filtering.

In the following experiments, the T_{max} is set 30 days.

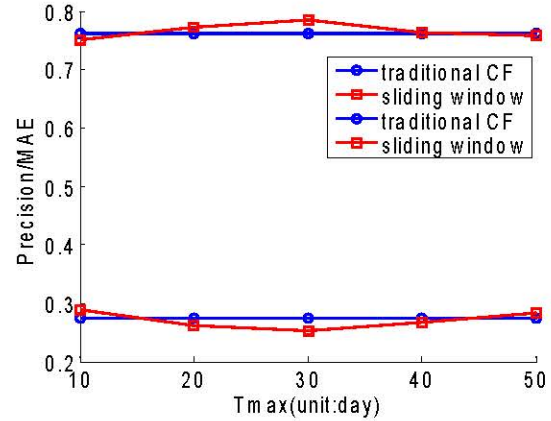


Figure 3. The results based on different values of T_{max}

(3) The parameter of time decay: k

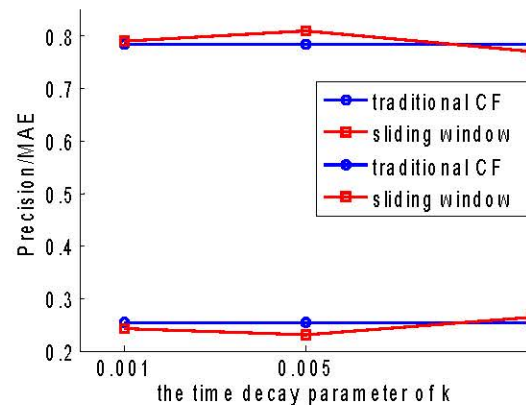


Figure 4. The results based on different values of k

From the Figure 4, we can know that when k is set 0.005, the results are the best. When the value of k is small, the decay rate is slow. For example, when $k=0.001$, the rating of 8 months ago is 78% of the original value. Therefore, though the performance is improved, but the level of the improvement is not obvious. When the value of k is too big, the rating reduces quickly. For example, when $k=0.01$, the rating of 8 months ago is 9% of the original value. So, the bigger the value of k is, the worse the results are. Even it is worse than the method that does not consider the time decay.

(4) The threshold of interval: T_s

The Figure 5 shows that, when T_s is set 10 days, the results are the best. The reason is the same with the Figure 3.

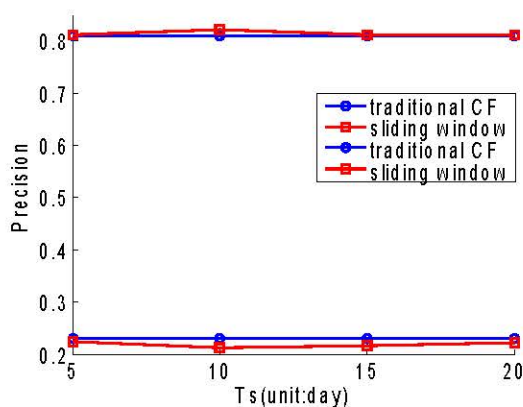


Figure 5. The results based on different value of T_s

IV. CONCLUSION AND FUTURE WORK

From the experiment results, we can know that the proposed method in this paper is feasible. In order to overcome the drawback of the collaborative filtering and improve the performance of the recommender system, researchers consider many factors and propose many useful methods. We also do some study, such as introducing the context, social relationship and so on. In this paper, we mainly focus on the timestamp and don't consider the other factors. In the future work, we combine the other factor to do more in-depth study.

REFERENCES

- [1] F. Xu, X. Meng, L. Wang. "A collaborative filtering recommendation algorithm based on context similarity for mobile users", *Journal of Electronics & Information Technology*, vol. 33, no. 11, pp. 2785-2789, Nov 2011.
- [2] X. Hu, X. Meng, L. Wang, "SVD-based group recommendation approaches: An experimental study of Moviepilot" *Proc. of the 2nd Challenge on Context-Aware Movie Recommendation*, Chicago, 2011, pp. 23-28.
- [3] L. Wang, X. Meng, Y. Zhang and y shi. "New approaches to mood-based hybrid collaborative filtering", *Proc. of the 1st Challenge on Context-Aware Movie Recommendation*, Barcelona, Spain, 2010, pp. 28-33.
- [4] S. Kostas, N. Eirini, P. Mihalits, N. Kjetil, K. Hans-Peter. "A framework for modeling, computing and presenting time-aware

- recommendations". *IEEE Trans on Large-Scale Data- and Knowledge-Centered Systems*, vol. 8220, pp. 146-172, 2013.
- [5] Y. Shi, X. Meng, Y. Zhang, L. Wang, "Adaptive learning approach of contextual mobile user preferences", *Journal of Software*, vol. 23, no. 10, pp. 2533-2549, Oct 2012.
- [6] Y. Shi, X. Meng, Y. Zhang, M. Xiao, "A trust calculating algorithm based on mobile phone data," *Proc. of the IEEE Global Telecommunications Conference, (GLOBECOM)*, California, USA, 2012, pp. 2036-2041.
- [7] T. Dao, S. Jeong, H. Ahn. "A novel recommendation model of location-based advertising: context-aware collaborative filtering using GA approach", *Expert Systems with Applications*, vol. 39, no. 3, pp. 3731-3739, Feb 2012.
- [8] Y. Zheng, R. Burke, B. Mobasher, "Differential context modeling in collaborative filtering", *Proc. of School of Computing Research Symposium*, Chicago, IL USA, May 2013.
- [9] L. Neal, H. Stephen, C. Licia, A. Xavier, "Temporal diversity in recommender systems", *Proc. the 33rd international ACM SIGIR conference on Research and development in information retrieval*, Geneva, Switzerland, 2010, pp. 210-217.
- [10] J. Wen, W. Zhou, "An improved item-based collaborative filtering algorithm based on clustering method", *Journal of Computational Information Systems*, vol. 8, no. 2, pp. 571-578, Jan 2012.
- [11] S. Nachiketa, S. Vir, M. Tridas. "A hidden markov model for collaborative filtering", *MIS Quarterly*, vol. 36, no. 4, pp.1329-1356, Dec 2012.
- [12] D. Ernesto, N. Wolfgang, D. Lucas, S. Lars, "Towards real-time collaborative filtering for big fast data", *Proc. of the 22nd international conference on World Wide Web companion*, Rio de Janeiro, Brazil, 2013, pp.779-780.
- [13] D. Li, P. Cao, Y. Guo, M. Lei, "Time weight update model based on the memory principle in collaborative filtering" *Journal of computers*, vol. 8, no. 11, pp.2763-2767, Nov 2013.
- [14] Y. Shi, X. Meng, L. Wang, "A heuristic approach to identifying the specific household member for a given rating," *Proc. of the 2nd Challenge on Context-Aware Movie Recommendation*, Chicago, 2011, pp. 47-52.
- [15] D. Fernando, G. Pedro. "Movie recommendations based in explicit and implicit features extracted from the Filmtipset dataset", *Proc. the Workshop on Context-Aware Movie Recommendation at the 4th ACM Conference on Recommender Systems (ACM Recsys'10)*, Barcelona, Spain, 2010: 45-52.

Yancui shi, female, 1982, baoding city, hebei province, china. Ph.D graduated beijing university of posts and telecommunications. Now, teaching in the tianjin university of science & technology. The major field of study is about the personalize service, recommendation system, social compute.

