Time-Aware Collaborative Filtering for Recommender Systems

Suyun Wei, Ning Ye, and Qianqian Zhang

College of Information Science and Technology, Nanjing Forestry University,
210037 Nanjing, China
{weisuyun,yening,zhangqianqian}@njfu.edu.cn

Abstract. Traditional collaborative filtering algorithms only take into account the users' historical ratings, which ignore the user-interest drifting and item-popularity changing over a long period of time. Aiming to the above problems, a time-aware collaborative filtering algorithm is proposed, which tracks user interests and item popularity over time. We extend the widely used neighborhood based algorithms by incorporating two kinds of temporal information and develop an improved algorithm for making timely recommendations. Experimental results show that the proposed approach can efficiently improve the accuracy of the prediction.

Keywords: recommendation systems, collaborative filtering, time weight.

1 Introduction

Collaborative Filtering (CF) is a technology that has emerged in e-Commerce applications to produce personalized recommendations for users [1-3]. It is based on the assumption that people who like the same things are likely to feel similarly towards other things, which traditionally deals with applications having only two types of entities, users and items, and infers users' preferences based on their historical ratings. However, in real-world scenarios, user interests are not static and may drift over time since they are continuously affected by moods, contexts, and pop culture trends. These observations show that, although many aspects of user interests can be found based on users' historical ratings, at a certain time slice, one user's interest may only focus on one or a couple of aspects. Thus, the static CF methods built on the entire historical ratings are inadequate to capture user-interest drift. In order to track user interests and create comprehensive user profiles such that different recommendation strategies can be used for consistent taste users and changing-taste users, a CF method that can model user interests over time is required.

Recently, researchers have observed the underlying temporal dynamics in CF and some temporal and evolutionary CF methods have been reported [4-8]. The solution [4] adopted is to model the temporal dynamics along the whole time period, allowing to intelligently separate transient factors from lasting ones. Liu et al. propose the temporal relevance measure [5] for ratings at different time steps and developed online evolutionary collaborative filtering algorithms by incorporating this measure into

C.-L. Liu, C. Zhang, and L. Wang (Eds.): CCPR 2012, CCIS 321, pp. 663–670, 2012.

nearest neighbor algorithms and incrementally computing neighborhood similarities. By introducing a set of additional time features to traditional factor-based collaborative filtering algorithms, and imposing a smoothness constraint on those factors, Xiong et al. present the Bayesian Probabilistic Tensor Factorization algorithm for modeling evolving relational data [6].Xing et al adopted two data weight techniques: the data weight based on time window and the data weight based on item-item similarity [7], then used a linear time function to describe gradually decay the history of past behavior as time goes by. Zhang et al present a collaborative filtering algorithm based on time period partition [8], in which users' rating history has been divided into several periods and analyzed users' interest distribution in these periods and quantize every user's interest.

In this paper, we propose a time-aware collaborative filtering framework for making recommendations based on user feedback data collected over time. We consider several effective time weighting techniques to incorporate temporal information into the widely used neighborhood based prediction algorithm so as adapt to changes in both user and item characteristics over time. We have validated our approach on real world data sets and the results demonstrated significant improvements of our methods in terms of both effectiveness and efficiency.

2 Traditional Collaborative Filtering

Many collaborative recommender systems try to predict the rating of an item for a particular customer based on how other customers previously rated the same item. More formally, the rating r_{ui} of item i for user u is estimated based on the ratings r_{vi} assigned to the same item i by those users v that are "similar" to user u.

Various approaches have been used to compute similarity measure $sim(u, v)$ between users in collaborative recommender systems. In most of these approaches, $sim(u, v)$ is based on the ratings of items that both users u and v have rated. The two most popular approaches are the *correlation-based* approach

$$sim(u,v) = \frac{\sum_{i \in Suv} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in Suv} (r_{ui} - \bar{r}_u)^2 \sum_{i \in Suv} (r_{vi} - \bar{r}_v)^2}} \tag{1}$$

and the *cosine-based* approach

$$sim(u,v) = \cos(U,V) = \frac{U \cdot V}{\|U\|_2 \times \|V\|_2} = \frac{\sum_{i \in Suv} r_{ui} r_{vi}}{\sqrt{\sum_{i \in Suv} r_{ui}^2 \sum_{i \in Suv} r_{vi}^2}} \tag{2}$$

Where r_{ui} and r_{vi} are the ratings of item i assigned by users u and v respectively, S_{uv} is the set of all items co-rated by both users u and v, and $U \cdot V$ denotes the dot-product of the rating vectors U and V of the respective users.

Once comparisons between the user and the rest of the community of recommenders are complete, rating predictions of unrated content can be computed based on the entire collection of items previously rated by the users. That is, the value of the unknown rating r_{ui} for user u and item i is usually computed as an aggregate of the ratings of some other (e.g., the K most similar) users for the same item i. Some examples of the aggregation function are:

$$\begin{aligned}
&(a) \quad r_{ui} = \frac{1}{K}\sum\nolimits_{v\in Nu} r_{vi} \\
&(b) \quad r_{ui} = \sum\nolimits_{v\in Nu} sim(u,v)\times r_{vi} / \sum\nolimits_{v\in Nu} |sim(u,v)| \\
&(c) \quad r_{ui} = \bar{r}_u + \sum\nolimits_{v\in Nu} sim(u,v)\times(r_{vi}-\bar{r}_v) / \sum\nolimits_{v\in Nu} |sim(u,v)|
\end{aligned} \tag{3}$$

Where N_u denotes the set of N users that are the most similar to user u and who have rated item i (N can range anywhere from 1 to the number of all users) and $\bar{r}_u$ is the average rating of user u.

3 Temporal Relevance

Traditional collaborative filtering algorithms have achieved success both in research and practice. However, in these approaches data collection is regarded as static. Ratings produced at different times are weighted equally. That is to say, changes in user purchase interest are not taken into consideration. However, we argue that the target user's recent ratings recent his/her future preferences more than the old ratings. A good collaborative filtering algorithm should gradually decay the influence of old data and predict the user future preferences precisely. From the perspective of intuition, an item that was rated recently by a user should have a bigger impact on the user's prediction than an item that was rated a long time ago. In this paper, we find appropriate time weights for different items that are rated at different times.

3.1 Time Weight Based on User Interest

As we assumed before, the user purchase interest is sensitive to time. The recommendation process should assign a greater level of importance to recent data. More recent data should have higher value in the time weighting. So we propose that each rating of item i assigned by users u is assigned a weight defined by a function $fu\_s(t_{ui})$ to the recent time.

Suppose that t_{ui} denotes the distance between the time when the user u rated on item i and the time when the user u visited one item in the first time, which can calculate by day or month. Generally the recent data is more valuable than the old data, to illustrate the feature, $fu\_s(t_{ui})$, a monotonic increasing function is defined. The recent time weight function of user interest is defined specifically as follow:

$$fu\_s(t_{ui}) = \frac{\gamma}{1+\exp(-t_{ui})} \tag{4}$$

Where, γ is weight factor, which can be adapted to optimize the prediction.

Within the system intersecting multiple products and customers, many different characteristics are shifting simultaneously, while many of them influence each other and often those shifts are delicate and associated with a few data instances. A more sensitive time weight is required, which can make better distinctions between transient effects and long term patterns. Thus, the early time weight function of user interest $fu\_l(t_{ui})$ is defined under the influence of old data.

Suppose that I_u denotes the set of the items visited by the user u , and I_{uT} is the set of the items visited by the user in a recent time-window T, which can actually kind of reflect the recent preference of the user. By measuring similarity between the item i ($i \in I_u$) visited in the early time and the items in I_{uT}, we can capture longer-term trends that reflect the inherent nature of the data. So the early time weight function of user interest $fu\_l(t_{ui})$ is defined as shown in (5)

$$fu\_l(t_{ui}) = \frac{\sum_{j \in I_{uT}} sim(i,j)}{|I_{uT}|} \tag{5}$$

The recent time weight function of user interest emphasizes the importance of the items visited recently and can capture the user recent interest over time. While the early time weight function of user interest avoids the ignorance of the valuable data in early time, which suits the situation where the user interest repeated periodically. The design of time weight based on user interest needs to find the right balance between assigning a greater level of importance to recent data, while capturing longer-term trends in old data.

$$fu(t_{ui}) = \alpha \times fu\_s(t_{ui}) + (1-\alpha) \times fu\_l(t_{ui}) \tag{6}$$

Where, scale factor $\alpha \in [0,1]$ is used to combine the two time weight function.

3.2 Time Weight Based on Item Popularity

In the recommendation system, the item popularity would change over time. For example, when a new product come into the market, it will attract the attention because of people' curiosity. As time goes by, the attention will gradually diminish. In addition, some products such as electronics, skin care and so on, is sensitive to season. So we propose time weight function $fi(t_i)$ based on item popularity to emphasize the impact of time on product popularity.

Suppose that t_i denotes the time period from the moment when the item com into people' sight to now, which can be calculated by day or month. The function $fi(t_i)$ definition which is used to compute time weight based on item popularity is shown in the formula (7) below.

$$fi(t_i) = \beta \times \exp(-\lambda t_i) + (1-\beta) \times \frac{\sum_{j=1}^{4} k_{ij}}{4} \tag{7}$$

Where, the value of $exp(-\lambda t_i)$ is reducing with t_i increasing, which shows the basic characteristic of new product popularity reducing over time. Besides, if k_{ij}=1 then the item i is sensitive to the season j else k_{ij}=0. Scale factor $\beta \in [0,1]$ is used to further improve the accuracy of the prediction.

4 Time Weight CF

Under the setting of evolutionary CF, the similarity between users is a function of time, which measures the similarity between two users based on their ratings on the set of users they are both rated up until time *t*.

To measure item similarities by incorporating temporal information, we modify the well known cosine similarity as follows:

$$sim(x,y)=\frac{\sum_{i\in Sxy}(fu(t_{xi})\times r_{xi})(fu(t_{yi})\times r_{yi})}{\sqrt{\sum_{i\in Sxy}(fu(t_{xi})\times r_{xi})^2}\sqrt{\sum_{i\in Sxy}(fu(t_{yi})\times r_{yi})^2}} \tag{8}$$

Incorporating the time weight based on user interest into the similarity computation as in (8) places right balance between importance to recent data and longer-term trends in old data, which is inclined to identify nearest neighbors, whose linked users are not only correlated but also synchronized in the sense that they are assigned similar ratings by the same set of users around time *t*.

For the score prediction step, we modify formula (3b) by designating time weight function based on item popularity $fi(t)$ to determine each rating's weight when making score predictions. Using separate temporal relevance functions for similarity computation and score prediction allows us to study the effect of temporal dynamics on neighborhood similarity and rating prediction independently.

$$r_{ui}=\frac{\sum_{v\in Nu} sim(u,v)\times fi(t_i)\times r_{vi}}{\sum_{v\in Nu}\left|sim(u,v)\right|} \tag{9}$$

We now describe the proposed time weight collaborative filtering algorithm in detail.

Algorithm 1. Time Weight CF algorithm (TWCF)

Input: user-item rating matrix R_{mn}, time information *t*.
Output: the *top-N* recommended set.

Method:

a) *for* any two users *u* and *v*, *do*
 Calculate similarity $sim(u,v)$ according to the formula (8);
 Add $sim(u,v)$ into the similarity matrix R_{sim} ,i.e., $R_{sim}=R_{sim}\cup sim(u,v)$;

b) *for* each user *u, do*
 Obtain the *K*-nearest neighbors $N_u=\{v_{i1},\ldots,v_{ik}\}$, $u\notin N_u$ and $sim(u,v_{i1})\geq\ldots\geq sim(u,v_{ik})$ according to matrix R_{sim};

c) *for* each user *u, do*
 Score the prediction of non-rated items according to the formula (9);
 Sort items in non-increasing order with respect to that prediction, and the first *N* items are selected as the *top-N* recommended set.

---

5 Experiments

5.1 Dataset Preparing

We use dataset provided by The Body shop (the subsidiary of L'Oreal), the dataset include 600 kinds of cosmetics and 400 diverse ranges of products, including face, body and hair cleaning products, skin care products, fragrance, perfume oils, essential oils, and makeup. We select 475,154 users on 3,856 kinds of productions, totally 9,548,825 purchase records. The density of the real rated data 9548825/(475154*3856)= 0.52%, it is quite sparse. Meanwhile we divide the rate matrix into practicing set and testing set. To show the percent that practicing set in whole data set, we introduce variable x. For example=0.8 means that 80% of data set is practicing set, remaining 20% is testing set. In this experiment, we always let x=0.8.

5.2 Evaluation Protocol

Mean Absolute Error (MAE) between ratings and predictions is widely used to evaluate the quality of collaborative filtering methods. The MAE is a measure of the deviation of recommendations from their true user-specified values. For each ratings-prediction pair$<p_i, q_i>$ this metric treats the absolute error between them, i.e., $| p_i, q_i |$ equally. The MAE is computed by first summing these absolute errors of the n corresponding ratings-prediction pairs and then computing the average. Formally,

$$MAE = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n} \tag{10}$$

5.3 Performance Comparison

To experimentally determine the impact of the parameter α on the quality of the prediction, we selectively varied the value of α to be used for similarity computation from 0.4 to 0.6 in increments of 0.1. Fig. 1 shows the user interest has a significant impact on the recommendation quality, as different values of α lead to substantially different MAE. Meanwhile our algorithm considering the user interest achieved consistently higher performance than the classic User-Based CF algorithm (UBCF).

Products popularity changes over time, Figure 2 shows the season has influence on skin care product sales, based on the feature of product, winter have lower sales, thus it has a lower weight.

In this experiment, we compare our new Time Weight CF algorithm (TWCF) to the classic Item-Based CF algorithm (IBCF) and User-Based CF algorithm (UBCF). The results are presented in Figure 3. Obviously, our new algorithm is able to boost the prediction accuracy for all configurations, which could be due to the following reasons. Firstly, the similarity tends to favor old products over new products as old products have received more ratings on them and their similarity with other products tend to be higher in general, the temporal relevance weighting can reduce the weights of past ratings on old products, thus reducing this kind of bias in similarity computation. Secondly, the prediction based on temporal relevance would consider how similar a rated item is to the current item as well as how recent a rating is, which could better reflect user's current interest. This is consistent with the intuition that it is more likely for the tastes of old users to have drifted over time, so the incorporation of temporal relevance appeared to be highly effective in modeling items' relevance with respect to users' current interests given both his most recent and more historical ratings.

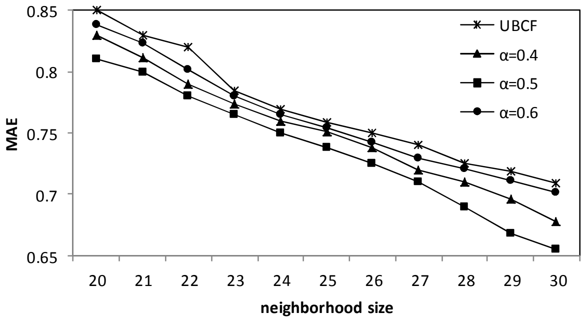

Fig. 1. Sensitivity of the parameter α on the user interest time weight

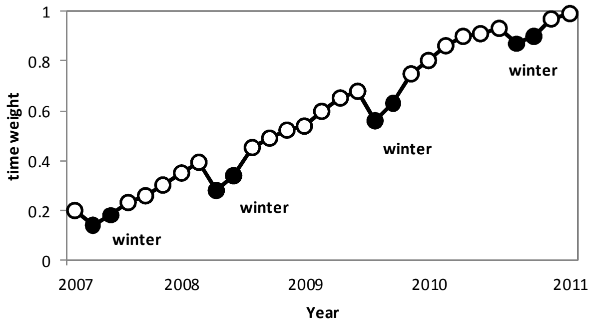

Fig. 2. The value of time weight based on item popularity

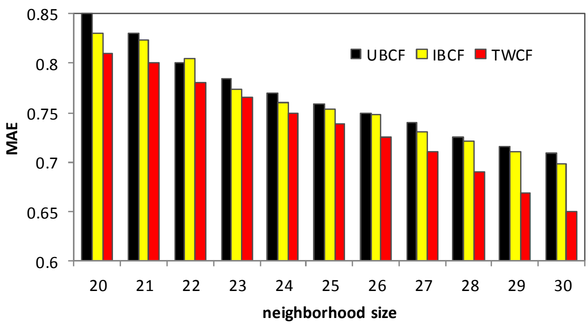

Fig. 3. Comparison of prediction quality of several CF algorithms

6 Conclusion

In this paper, we proposed the temporal relevance measure for ratings at different time steps and developed time-aware collaborative filtering algorithms by incorporating this measure into nearest neighbor algorithms. Experimental results on large real world data sets demonstrated that our algorithm can both significantly improve prediction accuracy. In the future, we would consider more complex models for the temporal dynamics of user feedbacks by incorporating more advanced time series analysis and modeling techniques.

Acknowledgments. The work is partially supported by National Basic Research Program of China (2012CB114505), China National Funds for Distinguished Young Scientists (31125008) ,Natural Science Foundation of Jiangsu Province(BK2009393), Jiangsu Qing Lan Project and Scientific and Technological Innovation Foundation of Jiangsu (CXLX11_0525), Technological Innovation Foundation of Nanjing Forestry Scientific (163070079), the Innovation Fund Project for College Student of Jiangsu (164070742).

References

1. Schafer, J.B., Konstan, J., Riedl, J.: Recommender Systems in E-Commerce. In: 1st ACM Conference on Electronic Commerce, pp. 158–166. ACM Press, New York (1999)
2. Sarwar, B., Karypis, G., Konstan, J.: Item-based Collaborative Filtering Recommendation Algorithms. In: 10th ACM Conf. on World Wide Web, pp. 285–295. ACM Press, New York (2001)
3. Linden, B., Smith, B., York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. Internet Computing 7, 76–80 (2003)
4. Koren, Y.: Collaborative Filtering with Temporal Dynamics. In: 15th ACM Conf. on Knowledge Discovery and Data Mining, pp. 447–456. ACM Press, New York (2009)
5. Liu, N., Zhao, M., Xiang, E., Yang, Q.: Online Evolutionary Collaborative Filtering. In: 4th ACM Conf. on Recommender Systems, pp. 95–102. ACM Press, New York (2010)
6. Xiong, L., Chen, X., Huang, T.: Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization. In: SIAM Conf. on Data Mining, pp. 211–222. SIAM, New York (2010)
7. Xing, C.X., Gao, F.D., Zhan, S., Zhou, L.: A Collaborative Filtering Recommendation Algorithm Incorporated with User Interest Change. Journal of Computer Research and Development 44(2), 296–301 (2007)
8. Zhang, Y.C., Liu, Y.Z.: A Collaborative Filtering Algorithm Based on Time Period Partition. In: 3rd IEEE International Symposium on Intelligent Information Technology and Security Informatics, pp. 777–780. IEEE Press, New York (2010)