Improved Slope One Algorithm Based On Time Weight

Tongqiang JIANG
School of Computer Science and Information Engineering,
Beijing Technology and Business University, BTBU
Beijing, China
jtongqiang@yahoo.com.cn

Wei LU
School of Computer Science and Information Engineering,
Beijing Technology and Business University, BTBU
Beijing, China
skylu.fei@163.com

***Abstract*—Collaborative filtering is regarded as the most prevailing techniques for recommendation system. Slope one is a family of algorithms used for collaborative filtering. It is the simplest form of non-trivial item-based collaborative filtering based on ratings. But all the family of use CF algorithms ignores one important problem: ratings produced at different times are weighted equally. It means that they can't catch user's different attitudes at different time. So in this paper, we present a new algorithm, which could assign different weights for items at different time. Finally, we experimentally evaluate our approach and compare it to the original Slope One. The experiment shows that the new slope one algorithms can improve the precision**

Keywords-Recommendation System; Collaborative Filtering; Slope One; Time Weighting

1. INTRODUCTION

With the rapid development of the Internet and information technology, the growth of information is much faster than the speed of information processing. Collaborative filtering technology is widely prevalent and most successful algorithm how to deal with the massive amounts of information. It predicts which items are one user's favorite according to analysis the track when user access to website. And now it has been widely applied to various areas of the Internet: film and video, music, social networking, reading, news, personalized e-mail and advertising and so on.

Slope one collaborative filtering algorithm is a classic 2005 users - project scoring matrix algorithm. Compared to the traditional user and project-based collaborative filtering algorithms, the algorithm is easy to implement better maintenance, high efficiency, and be able to produce in case of lack of information user reviews of recommendation [1]. Even though slope one algorithm shows good performance, but there are still some places that sound not make sense. In this approaches, ratings produced at different times are weighted equally. That is to say, changes in user interests are not taken into consideration. This is obviously exists great discrepancy with real world [2]. So we propose some improvement on slope one algorithm, and through a series of experiments results, we could prove that our improvement on slope one is effective.

II.SLOPE ONE ALGORITHMS

A. *Background of slope one*

Slope one algorithm is put forward in 2005 by Professor Daniel Lemire an Item-Based recommendation algorithm. The basic idea of the algorithm is very simple, which is to use the average instead of the scoring difference between a two unknown individuals. Their simplicity makes it especially easy to implement them efficiently. Slope one algorithm is one method of item-based, and it has more advantage than traditional item-based:

- it easy to implement and maintain
- updateable on the fly
- efficient at query time
- expect little from first visitors
- accurate within reason

The slope one schemes take into account both information from other users who rated the same item and from the other items rated by the same user. And its predictors are of the form f (x) = x + b, where b is a constant and x is a variable presenting rating values. The free parameter is then simply the average difference between the two items' ratings. It was shown to be much more accurate than linear regression in some instances. For example, the ratings for five movies which user A, B, C, D recorded are as follow:

TABLE I. FOUR USER'S RATINGS FOR FIVE MOVIES

User/movies	**1**	**2**	**3**	**4**	**5**
A	3	4	2	-	3
B	2	2	3	2	2
C	4	-	3	5	4
D	-	3	4	-	-

If we want to know how the user D rate to the movie of five, we must firstly compute the average difference value which user D had rated to the movie 2 and movie 3, the calculation process is:

$$((3-4) + (2-2))/2=-0.5,$$
$$((3-2) + (2-3)+(4-3))/3=0.3$$

Published by Atlantis Press, Paris, France.

Then we can get the predictive value user D for movie 5 through the user D rated movie 2 and movie 3 plus the corresponding arithmetic mean:

((3-0.5) +(4+0.3)/2=3.4

So, we can fill the empty value table in a similar manner.

B. *Problem of slope one*

As we all know, the time information is invisible attributes of user behavior in a very important, the information often be automatically saved in the system log file. So it is very easy to get the information of this dimension in production system. Slope one algorithm performance already has a good performance, but unfortunately is the slope one not the time factor into consideration. Therefore, this article attempts User Rating time information as an impact factor model to measure the degree of correlation between the project and user ratings thereby improving the quality of the recommended combination.

III.SLOPE ONE ALGORITHM BASED ON USER RATINGS TIME

Traditional collaborative filtering algorithm based on the project is mainly to study how to contact the user interest and items [6], items that will best meet the user's interest and recommended to the user, but these algorithms are overlooked the point where is the user context. The context information includes many users info such as user access to the system time, place, and mood and so on and these factors are very important for improving the performance of the recommendation system. To solve the above problem, we propose the rated time-based collaborative filtering, which take into account combining time information with the slope one recommendation modeling. So the new model can accurately predict user interest in a particular moment. Essentially speaking, the model is still TopN recommended to the user--how to generate a recommendation list of length N and the list will includes the user most favorite items in one moment.

Through the introduction of Section II, we know that the slope one algorithm is very simple and efficient, which consists of two core parts [3]:

- calculate the similarity of items offline according to users historical rating
- Give users do online personalized recommendation based on the user's historical rating

For example, an item that was rated recently by a user should have bigger impact on the prediction of future user behavior than an item that was rated a long time ago. It means that user's recent behavior can reflect the user's interest better in reality than the behavior long before. For the prediction of the user's interest, it should increase the weight of the user's recent behavior and give priority recommended for those items which is similar to user's recently favorite items [4].

We use the following formula when calculate the similarity through slope one algorithm:

$$dev_{j,i}=\sum_{u\in S_{j,i}(\theta)}\frac{u_j-u_i}{card(S_{j,i}(\theta))} \qquad (3\text{-}1)$$

Where u_j and u_i identify the user's ratings on item j and i.

And when recommend to the user, the rating on item j calculated by the following formula:

$$P(u)_j=\frac{1}{card(R_j)}\sum_{i\in R_j}dev_{j,i}+u_i \qquad (3\text{-}2)$$

As we assumed before, the user recent interest is sensitive to time, so we propose an improvement method by assigned a weight defined by a function f (t) to the time t:

$$dev_{j,i}=\sum_{u\in S_{j,i}(\theta)}\frac{(u_j-u_i)*f(|T_{uj}-T_{ui}|)}{card(S_{j,i}(\theta))} \qquad (3\text{-}3)$$

From the formula we can see that we introduce a time-related decay term into slope one algorithm model, which is the time of the user u of item j rating [5]. The meaning of the function is that the time interval user rating of item j and items i is larger, time impact factor $f(|T_{uj}-T_{ui}|)$ is smaller:

$$f(|T_{uj}-T_{ui}|)=\frac{1}{1+\alpha(T_{uj}-T_{ui})} \qquad (3\text{-}4)$$

α is time decay parameters, its value is different in the different systems. If user's interest changed rapidly, then will take a larger value of α, otherwise go to a relatively small value.

In addition to considering the influence of time information on the historical data, you should also consider the impact on the prediction formula. In general, the recent behavior of the user has more effects when predict user's behavior in future. So we can get the following new version formula:

$$P(u)_j=\frac{1}{card(R_j)}\sum_{i\in R_j}dev_{j,i}+u_i*(\frac{1}{1+\beta(T_0-T_{uj})})$$

(3-5)

T_0 represents the current time. When T_{uj} is close to T_0, items which are similar to item j will get higher rank in recommendation list of user u.

IV.THE EXPERIMENTAL RESULTS SHOW

In this trial, we want to prove that the slope one algorithm combines time attribute will have more great performances than the original algorithm. In order to confirm the validity of the new algorithm, we use the accuracy to test it. Forecast accuracy can be divided into two areas, one is rating prediction, and the other is TopN recommended. Although most of the prevalent algorithms are about to the rating predicted in current study, however, some scientists pointed out that this method may exists some problem. For example, the purpose of the movie recommendation is to find which movies are the most likely interested instead of predicting user will give the movie what kind of rating. A high rating movie may not the user's interesting movie. So the TopN recommended is more satisfied with the actual needs of the application, we will use TopN recommended to test the performance of the algorithm.

A R (u) is a recommendation list which based on user ratings in the training set, T (u) is a list of user behavior in the test set, and the recommendation result accuracy is defined as:

$$\text{Precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|}$$

We use part of Netflix data set as our experimental data sets, including 10,000 ratings and ratings time 1230 independent users of the 1682 movie. Each record is composed by users, items, rating and time. We use user ratings records for the last 20 days as a test set, the rest of the record as a training set.

Here, we do not contrast display the performance of the algorithm in the different attenuation parameter β, the final experiment was performed at the optimal parameter β = 0.8 results obtained under. Select a different N (10, 20 ... 50) 5 experiments to verify the performance of the algorithm, by comparing the T-Slope one algorithm and the original algorithm.

From Figure 4-1 we can see that, compared with the original slope one algorithm, the new version algorithm has higher accuracy when consider time dimension. So the result can adequately prove that time weight slope one algorithm could enhance the performance of the old, about improve 3.1% than the old.

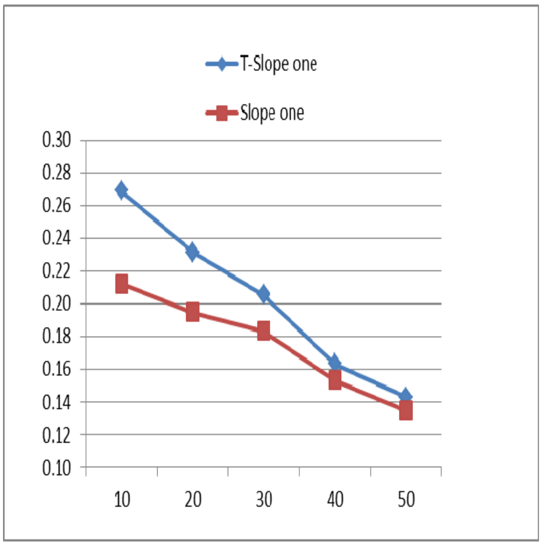

Figure 4-1

V.COPYRIGHT FORMS AND REPRINT ORDERS

We improve existing slope one algorithm based on this reason: More recent item rating should have higher value in the time weighting. Unlike the non-time weighted algorithms, we utilize a predefined time function for different items to assign different weight. The main idea is to predict precisely user future interest by deploying time weight. The items which were rated recently by a user should have more impact on the user's prediction than an item which was rated long time ago. The experiments have shown that time weights slope one algorithm can definitely improve the precision when compare with the old.

REFERENCE

[1] Lemire D, MacLauchlan A, Slope One Predictors for online rating-based collaborative filtering. In: SIAM Data Mining (SDM'05), Newport Beach, California, April 21-23, 2005

[2] R. Jin, J. Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In Annual ACM Conference on Research and Development in Information Retrieval, pages 337 -344, 2004.

[3] Q. Li and M. Zhou. Research and design of an efficient collaborative filtering predication algorithm. In Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003, pages 171-174, 2003.

[4] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), Volume 22(Issue 1):5-53, 2004.

[5] Y. Zhao, C. Zhang, and S. Zhang. A recent-biased dimension reduction technique for time series data. In Advances in Knowledge Discovery and Data Mining: 9th Pacific-Asia Conference, PAKDD 2005, volume 3518 / 2005. Lecture Notes in Computer Science, 2005

[6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In International World Wide Web Conference, pages 285-295, 2001.