

# Collaborative Filtering Algorithm Based on Item Attribute and Time Weight

CHEN Qian

School of Mathematics and Computer Science  
Anhui Normal University  
Wuhu, China

LI Wanggen

School of Mathematics and Computer Science  
Anhui Normal University  
Wuhu, China

LIU Jiao

School of Mathematics and Computer Science  
Anhui Normal University  
Wuhu, China

**Abstract**—Collaborative filtering is a recommendation algorithm which is used in personalized system. To solve the problem of low accuracy caused by sparse data in the user-item matrix of traditional collaborative filtering algorithm, this paper presents a hybrid algorithm. Firstly, it uses the data based on the similarity of item's attributes to fill the matrix. And then a weight decrease by the time is given to increase the effectiveness of the measurement, thereby to improve the accuracy of the collaborative filtering algorithm. Experimental results show that the algorithm proposed in this paper can improve the accuracy of recognition and enhance the quality of the recommendation system.

**Keywords**—Collaborative filtering; Sparse data; Item attribute; Time weight

## I. INTRODUCTION

With the explosive growth of network information, the overload has become a major problem to users. Recommendation system can effectively overcome this and help users locate from massive amounts of data and find beneficial information quickly. Among them, Collaborative Filtering is one of the most effective recommendation technologies [1-3].

With the growth of users and items, the date in User-Item ratings matrix increases rapidly in traditional collaborative filtering algorithm. This leads to the problem of extremely sparse data in rating matrix, which affect the accuracy of nearest neighbor search. To deal with the problem of sparse data, SVD, Bayesian network and BP ANN were used to ease the problem of sparse [4-6]. These methods can effectively improve the accuracy of the recommendation. But they did not consider the influence of the item attributes. Peng et al. proposed an approach to calculate the similarity of items for rating matrix, but they didn't consider the time property of the data [7]. On the other hand, some work considered only the time property [8-9].

For this purpose, this paper brings out a new collaborative filtering recommendation algorithm based on item attribute and time weight. To prediction item rating and fill the user-item rating matrix and alleviate the data sparseness problem, the similarity of item attributes have been considered properly. At the same time, introducing the function of time, considering the

timeliness of rating data, this improves the quality of recommendation algorithms.

## II. THE PROBLEM DEFINITION AND BASIC METHODS

Collaborative filtering algorithm is the most widely used and most mature personalized recommendation algorithm. The core idea is: System based on existing rating data and computing users (or items) similarity, according to the similarity search for the nearest neighbors of target users (or items), thus according to nearest neighbor prediction score to generate recommendation, which most commonly used were user-based collaborative filtering and item-based collaborative filtering. This paper mainly introduces the user-based collaborative filtering algorithm.

In user rating data, user set is represented as  $U = \{U_1, U_2, \dots, U_m\}$ , and item set is represented as  $I = \{I_1, I_2, \dots, I_n\}$ . The matrices of rating data are as shown in Table 1.

	$I_1$	$I_2$	...	$I_n$
$U_1$	$R_{11}$	$R_{12}$	...	$R_{1n}$
$U_2$	$R_{21}$	$R_{22}$	...	$R_{2n}$
...	...	...	...	...
$U_m$	$R_{m1}$	$R_{m2}$	...	$R_{mn}$

### A. The traditional similarity measure method

There are three methods for calculating the similarity of users in generally: Pearson similarity, Cosine similarity and Adjusted Cosine similarity. Here we set  $R_{u,i}$  and  $R_{v,i}$  as the rating for user u and user v to item i,  $I_{U,V} = \{i \in I | R_{u,i} \neq \emptyset \text{ and } R_{v,i} \neq \emptyset\}$  as the common item rating for user u and user v.

- Pearson similarity

The Pearson similarity between user u and user v is:

$$sim(u, v) = \frac{\sum_{i \in I_{U,V}} (R_{u,i} - \bar{R}_u)(R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_{U,V}} (R_{u,i} - \bar{R}_u)^2 \sum_{i \in I_{U,V}} (R_{v,i} - \bar{R}_v)^2}} \quad (1)$$

In this formula,  $\bar{R}_u$  and  $\bar{R}_v$  as the average rating for user u and user v.

- Cosine similarity

The item rating of user can be viewed as the vector in n-dimensional space, the similarity between users by calculating the cosine between vectors. The angle between users is smaller, the similarity is higher. Setting the rating vector for user u and user v in the  $I_{U,V}$  of the common items as  $\vec{u}$  and  $\vec{v}$ , then the cosine similarity between user u and user v is:

$$sim(u, v) = \cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \times \|\vec{v}\|} = \frac{\sum_{i \in I_{U,V}} R_{u,i} R_{v,i}}{\sqrt{\sum_{i \in I_{U,V}} R_{u,i}^2 \sum_{i \in I_{U,V}} R_{v,i}^2}} \quad (2)$$

- Adjusted Cosine similarity

It doesn't consider the difference rating scale between users in the cosine similarity. In order to avoid the impact of extreme evaluation, an adjusted cosine similarity is proposed. The adjusted cosine similarity between user u and user v is:

$$sim(u, v) = \frac{\sum_{i \in I_{U,V}} (R_{u,i} - \bar{R}_u)(R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_U} (R_{u,i} - \bar{R}_u)^2 \sum_{i \in I_V} (R_{v,i} - \bar{R}_v)^2}} \quad (3)$$

### B. The result of collaborative filtering recommendation

Through the user similarity, we can find the nearest neighbor set  $N_u = \{n_1, n_2, \dots, n_m\}$  for user u, and  $sim(u, n_1) > sim(u, n_2) > \dots > sim(u, n_m)$ . To determine the nearest neighbor in the process, there are two kinds of methods: a fixed number of nearest neighbor method and similarity threshold. The fixed number of nearest neighbor method is the most commonly used.

By getting the set of target user's nearest neighbor  $N_u$ , then we can predict the target user's rating for item i through the rating by the nearest neighbor:

$$P_{u,i} = \bar{R}_u + \frac{\sum_{v \in N_u} sim(u,v) \times (R_{v,i} - \bar{R}_v)}{\sum_{v \in N_u} |sim(u,v)|} \quad (4)$$

In this formula,  $R_{v,i}$  is the rating given by user v for item i,  $\bar{R}_u$  and  $\bar{R}_v$  is the average rating for user u and user v.

Through the above method, we can predict the user's unrated item and get the recommended list to the target user u.

## III. COLLABORATIVE FILTERING ALGORITHM BASED ON ITEM ATTRIBUTE AND TIME-WEIGHTED

### A. The similarity based on item attributes

Setting  $A = \{a_1, a_2, \dots, a_k\}$  as item attributes set, item attribute is determined by the item itself. For example, in the movie recommendation system, the film features can be divided into the following attributes: action, horror, animation, comedy, romance, crime, and set those attributes as  $a_1 \sim a_6$ .

Then the feature attributes matrix for films  $M_1 \sim M_4$  are list in Table 2:

TABLE II.		MOVIE ATTRIBUTES TABLE					
	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	
$M_1$	0	1	0	0	1	0	
$M_2$	0	0	0	1	0	1	
$M_3$	1	1	0	0	1	0	
$M_4$	0	0	0	0	1	0	

In the Table 2, zero means the film doesn't has this attribute and 1 means has this attribute. Setting feature vector as  $m = \{m_1, m_2, \dots, m_k\}$  for film m and  $n = \{n_1, n_2, \dots, n_k\}$  for film n, the value of  $m_i$  or  $n_i$  is 0 or 1. Using Jaccard's similarity to calculation the item attribute similarity in this paper, the similarity between film m and film n as  $sim\_attr(m, n)$  is:

$$sim\_attr(m, n) = \frac{\sum_{i=1}^k S_i}{A_m + A_n - \sum_{i=1}^k S_i} \quad (5)$$

$$S_i = \begin{cases} 1, & m_i = n_i = 1 \\ 0, & \text{others} \end{cases}$$

In this formula,  $A_m$  means the number of attributes of film m and  $A_n$  means the number of attributes of film n, k means the number of item attribute. Through the above method, we can calculation the similarity between items.

### B. Predict the value of target item

According to the item attribute similarity from the previous section, we can get the neighbors. On the basis of the neighbors set  $M$ , calculation the target value  $P_{u,k}$  from user u to item k:

$$P_{u,k} = \bar{R}_k + \frac{\sum_{m \in M} S_{k,m} \times (R_{u,m} - \bar{R}_m)}{\sum_{m \in M} S_{k,m}} \quad (6)$$

In this formula,  $S_{k,m}$  means the similarity between item k and item m,  $R_{u,m}$  means the rating from user u to item m,  $\bar{R}_k$  means the average rating for film k and  $\bar{R}_m$  means the average rating for film m.

Make the calculated results filling to the user-item rating matrix; increase the matrix data, thereby reducing the sparseness of data.

### C. Time weight

Due to the traditional user-based collaborative filtering algorithm didn't consider the user's preferences change with the time when we find the user's nearest neighbor, we add time effect to modify the item evaluation to improve the accuracy.

In this paper, we use function of logistic for the time weight[10], the function is:

$$f(t_{v,i}) = \frac{1}{(1 + e^{-t_{v,i}})} \quad (7)$$

This function is a monotone increasing function, first made the value of time data standardization and mapping on [-1,1], with the  $t_{v,i}$  become larger, the rating time is closer, the value

of the interest between user and item is bigger. The modified formula according to (4) is:

$$P_{u,i} = \bar{R}_u + \frac{\sum_{v \in N_u} \text{sim}(u,v) \times (R_{v,i} - \bar{R}_v) \times f(t_{v,i})}{\sum_{v \in N_u} |\text{sim}(u,v)| \times f(t_{v,i})} \quad (8)$$

D. Algorithm description

This paper improves the traditional collaborative filtering algorithm, calculation the item attribute similarity, to predict the value to fill the user-item rating matrix, and introducing time-weight, modified the prediction value. Algorithm description is as follow:

Input: Data set

Output: The set of recommendations for target user

- (1) Using user rating data generated user-item rating matrix, item-attribute matrix and rating time matrix.
- (2) Using formula (5) to build the matrix of item attributes similarity based on item attribute matrix.
- (3) Using formula (6) to predict the user's unrated item rating and filled in the user-item matrix based on item attribute similarity matrix.
- (4) Using formula (1) to computation user similarity and find the neighbors set based on filled rating matrix.
- (5) According to the rating time matrix, using formula (7) to calculation the time-weight.
- (6) According to the neighbors set, using formula (8) to calculation the final rating and build the recommendation set.

IV. THE RESULTS OF THE EXPERIMENT

A. The experimental data and platform

By using the MovieLens data set, which contains 943 users, 1682 movies information, and 100 thousand film rating information in this paper. The 80% data set is as the training set and the remaining 20% is for the test set. The experimental platform using the processor: Intel(R) Pentium(R) CPU G3240 @ 3.10GHz, 4GB memory, the operating system is Windows 7, written by MATLAB.

B. Experimental results and analysis

In order to test the performance of the recommendation algorithm, we consider the two aspects which include mean absolute error (MAE) and precision in this paper.

• MAE

Set  $E_f = \{p_1, p_2, \dots, p_n\}$  as the prediction of user rating set, the actual rating set is  $E_p = \{q_1, q_2, \dots, q_n\}$ , the formula is as follows:

$$MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N} \quad (9)$$

The smaller the MAE, the prediction accuracy and the quality of recommendation are higher.

• Precision

The precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the posi-

tive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class), the formula is as follows:

$$Precision = \frac{|test \cap TopN|}{N} \quad (10)$$

The precision is bigger, recommend better.

Compared with the traditional User-Based Collaborative Filtering algorithm (UBCF) and the improved algorithm (CFBWJI) from reference [7], the neighbor's number of target user from 5 increases to 60, the results of MAE are shown in Fig 1. And the results of precision are shown in Fig 2.

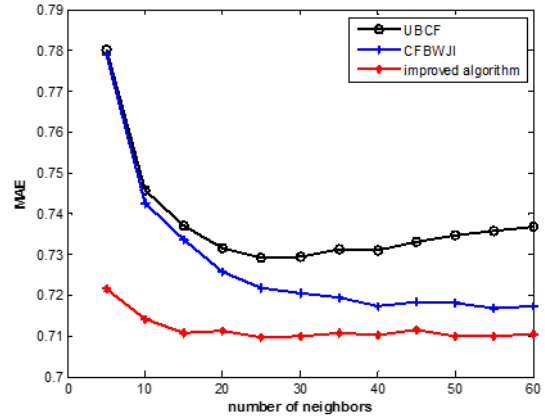


Fig. 1. MAE with different neighbors

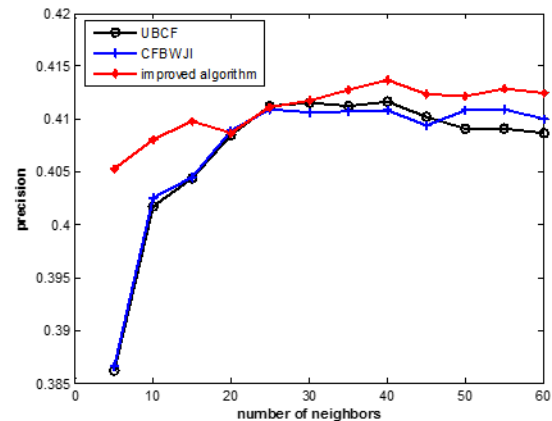


Fig. 2. Precision with different neighbors

The experimental results show that the improved algorithm in this paper is superior to the traditional User Based Collaborative Filtering algorithm (UBCF) and the improved algorithm (CFBWJI) from reference [7], can effectively improve the quality of the recommendation system.

V. CONCLUSION

The traditional User Based Collaborative Filtering algorithm has the problem of sparse data. To alleviate the data sparseness problem, in this paper using computation item at-

tribute similarity, predict item rating, introduce time-weight to modify the data timeliness. The results show that the improved algorithm can effectively improve the quality of the recommendation benefit. The next step, we will consider the item attribute and time data influence on the algorithm more fully, and then provide a more accurate recommendation.

#### REFERENCE

- [1] Yoon Ho Cho and Jae Kyeong Kim. "Application of Web usage mining and product taxonomy to collaborative recommendations in e-commerce." *Expert Systems with Applications* 26.2 (2004): 233-246.
- [2] Deepa Anand and Kamal K. Bharadwaj. "Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities." *Expert Systems with Applications* 38.5 (2011): 5101-5109.
- [3] Amir Albadvi and Mohammad Shahbazi. "A hybrid recommendation technique based on product category attributes." *Expert Systems with Applications* 36.9 (2009): 11480-11488.
- [4] Sarwar Badrul, Karypis George, Konstan Josph, Riedl John. "Application of Dimensionality Reduction in Recommender System - A Case Study." Minnesota Univ Minneapolis Dept of Computer Science, 2000.
- [5] Yung-Hsin Chen and Edward I. George. "A bayesian model for collaborative filtering." *AISTATS*. 1999.
- [6] Yong Xu and David Zhang. "Accelerating the kernel-method-based feature extraction procedure from the viewpoint of numerical approximation." *Neural Computing and Applications* 20.7 (2011): 1087-1096.
- [7] Shi Peng, Zhi-Bin Zhou, Guo-Jun Wang. "Collaborative Filtering Algorithm Based on Rating Matrix Pre-filling." *Computer Engineering* 39.1(2013): 175-178. In Chinese.
- [8] Tong Queue Lee, Young Park, Yong-Tae Park. "A time-based approach to effective recommender systems using implicit feedback." *Expert Systems with Applications* 34.4 (2008): 3055-3062.
- [9] Yi Ding and Xue Li. "Time weight collaborative filtering." *Proceedings of the 14th ACM international conference on Information and knowledge management*. 2005.
- [10] Ji Liu and Guishi Deng. "Link prediction in a user-object network based on time-weighted resource allocation." *Physica A: Statistical Mechanics and its Applications* 388.17 (2009): 3643-3650.