

## Neighbor-based Data Weight Collaborative Filtering

<sup>1</sup> Xuesong Yin, <sup>2</sup> Jie Yu, <sup>1</sup> Rongrong Jiang

<sup>\*1</sup>School of Information & Engineering, Zhejiang Radio & Television University, Hangzhou, China

<sup>2</sup> Zhejiang Agriculture & Forestry University, Hangzhou, China

<sup>1</sup>School of Information & Engineering, Zhejiang Radio & Television University, Hangzhou, China

**Abstract:** Existing Collaborative Filtering (CF) based recommendation approaches suffer from the following issues: (1) the number of resources accessed and evaluated by each user is only such a very small part that leads to sparse rating matrix; (2) dynamic change of user interest makes recommended resources largely deviate from the need of the user. To address these problems, we develop a novel algorithm titled as neighbor-based data weight CF recommendation of learning resources (NARR). Firstly, the neighbor of the user or the neighbor of the resource is selected in terms of the rating matrix; secondly, we compute data weight for representing dynamic change of user interest; finally, we use neighbor relationship and data weight in the objective of CF-based algorithm to choose learning resources. Experiments results show the feasibility and effectiveness of the proposed method.

**Keywords:** Collaborative Filtering, Data Weight, Similarity, Data Sparseness

### I. Introduction

Development of information technology makes knowledge transfer from the traditional paper-based forms of communication transition to digital transmission. Internet's rapid development and growing popularity makes online information become an important source of knowledge. Distance education regarded as a network technology based education form can maximize the use of network teaching resources. Under the education mode learners do not have to be time and space constraints and carry out learning at any time and any place according to their own needs. However, the main problem faced by distance education is learning resource overload. With explosive growth in the number of resources, on the one hand, learners cannot find the resources they need to learn; on the other hand, they are often unable to distinguish the pros and cons of resources, which results in low learning resource utilization. Recommender system (RS) is the very promising approach to solve the problem of current information overload. Different with information retrieval, RS provides personalized recommendations based on user profile and preference. Accurate recommendations enable users to quickly locate desirable items without being overwhelmed by irrelevant information so that users are gradually dependent on the system. Thus, RS can not only improve personalized service for users, but set up long-term stable relationship with users and improve customer loyalty.

RS can be considered as social networking tools that provide dynamic and collaborative communication, interaction and knowledge [1][2]. RS is inspired by human social behavior and covers a wide variety of applications. It is common to take into account the tastes, opinions and experiences of our acquaintances when making all kinds of decisions (choosing films to watch, selecting schools for our children, choosing products to buy, etc.) [3][4][5]. Decisions of users are modified according to their interpretation of the similarity that exists between user group of acquaintances. Specifically, a RS aims at improve a user with those items that might be of his or her interest. RS has three elements: recommended candidates, users and recommended methods. The general recommendation system model is shown in Figure 1. RS actively collects users' information or users voluntarily provide their preferences to RS. Then, RS can use different recommended strategies to show items. For example, RS calculates personalized information and user data collected to obtain recommended results or directly bases on the knowledge database modeled to recommend those items.

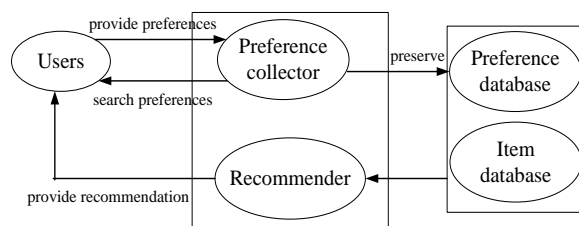


Figure 1. General model of recommender system

The rest of this paper is organized as follows. We formally present related work of RS in Section 2. We present a new collaborative filtering (CF) based recommended approach in Section 3. We then make a comparison of CF-based social recommendation approaches. Finally, we set our most relevant conclusions in Section 4.

## II. Related work

Given a user set  $C$  and a recommended object set  $S$ , RS aims to find items that are of interests of users and is defined as follows [6]:

$$\forall c \in C, s^* = \arg \max_{s \in S} u(c, s) \quad (1)$$

where the function  $u(c, s)$  is used to calculate the recommended degree of the user  $c$  towards the object  $s$  and is a non-negative real number.

Generally, there are four variants of recommendation methods: content based, collaborative filtering (CF) based, knowledge based and hybrid approaches. The content based approach recommends the object of other similar attributes to the user according to the object selected by the user. The CF based method is the most popular one. Its basic idea is to find other users that are similar to the recommended user, and then to recommend the suitable object to the user by computing the recommended degree. The knowledge based approach is of an inference technology and recommend the contents in terms of rules in specific areas. The hybrid one is to recommend the object to the user through combining the above methods. RS generally base their operation on a CF process, which provides personalized recommendations to active users of websites where different elements (products, films, holidays, etc.) can be rated [2]. Since most existing recommender systems are CF-based, in this paper, we aim at investigating CF-based algorithm. CF-based methods can be further grouped into model-based CF and memory-based CF [7]. The former uses user-item ratings to learn a predictive model. The general idea behind the approach is to model the user-item interactions with factors representing latent features of users and items in the system, such as the preference class of users and the category class of items. The later applies user-item ratings stored in the system to directly predict ratings for new items.

Input data are described to a rating matrix  $R$  of  $m \times n$ , where  $m$  is the number of users,  $n$  is the number of items and the element  $R_{ij}$  is a rating value which is the  $i$ -th user to rate the  $j$ -th item. If two users  $x$  and  $y$  become neighbor, they have the big similarity based on their ratings of items. General CF-based algorithms have the following step [8]:

Step 1: Computing the similarity of items. Traditional similarity calculation method has the following three kinds:

(1) Cosine similarity. The two items  $i$  and  $j$  are looked as the two matrixes  $a$  and  $b$ . If the user does not rate the item,  $R_{ij}$  is zero. The similarity between items is computed by cosine similarity and described as follows:

$$S(i, j) = \cos(a, b) = \frac{a \cdot b}{\|a\|_2 \|b\|_2} \quad (2)$$

(2) Fixed cosine similarity. Different users have different rating evaluations, thus cosine similarity rule does not legitimately reflect the relation of users. Through subtracting the rating mean value, the similarity of the two items  $i$  and  $j$  is computed as follows:

$$S(i, j) = \frac{\sum_{u \in U_{ij}} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U_{ij}} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U_{ij}} (R_{u,j} - \bar{R}_u)^2}} \quad (3)$$

where  $U_{ij}$  represents users rating the items  $i$  and  $j$ .

(3) Correlation similarity. Finding users for jointly rating the items  $i$  and  $j$ , correlation similarity is defined as:

$$S(i, j) = \frac{\sum_{u \in U_{ij}} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U_{ij}} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U_{ij}} (R_{u,j} - \bar{R}_j)^2}} \quad (4)$$

Step 2: Choosing  $k$  items being the most similar to the current item according to  $S(i, j)$ .

Step 3: Evaluating the score of the current item the user  $u$  rates:

$$E_{u,i} = \bar{R}_i + \frac{\sum_{j \in Neig(i)} S(i, j)(R_{u,j} - \bar{R}_j)}{\sum_{j \in Neig(i)} (|S(i, j)|)} \quad (5)$$

Where  $Neig(i)$  is neighbor items of the item  $i$ .

Symeonidis et al. [9] proposed an algorithm to realize both accurate and justifiable recommendations through constructing a feature profile for the users to reveal their favorite features. Buhwan et al. [10] presented a novel memory-based approach which incorporates the level of a user credit instead of using similarity between users. Symeonidis et al. [11] proposed new approaches that substantially improve the performance of CF algorithms in terms of accuracy and efficiency. Bobadilla et al. [2] proposed a CF-based framework that gives special importance to the quality of the predictions and the recommendations, as well as to the novelty and trust results. Yang et al. [6] presented how social network information can be adopted by recommender systems as additional input for improved accuracy. Shinde et al. [12] presented a hybrid approach which extracts user's current browsing patterns using web usage mining, and forms a cluster of items with similar psychology to obtain implicit users rating for the recommended item.

### **III. Personalized recommendation algorithm**

Existing CF-based methods suffer from the following issues: data sparseness and dynamic changes of user interest. On an E-commerce website or online learning platform, there are a large number of sources. For example, Taobao website has more than three hundred million merchandises with nearly ten thousand daily changing. In fact, the number of resources accessed and evaluated by each user is only such a very small part that leads to sparse rating matrix. On the other hand, CF-based methods that focus on the similarity between the user and the item ignore dynamic changes of user interest. In applications, user demand for resources changes with the passage of time. Traditional CF-based algorithms used only the rating matrix calculated without considering the specific time of user access to resources [13]. Thus, the interest change of the user does not reflect. When a user interest is transformed, the existing system is unable to timely detect the change, which causes recommended resources largely deviated from the user's needs.

#### **3.1. Selection of Neighbor and Data Weight**

To alleviate the above issues, in this paper, we present a new approach titled as neighbor-based data weight CF recommendation of learning resources (NARR). Owing to data sparseness in the rating matrix, only using user rating or item rating exists in the inaccuracy question of calculating results. Moreover, the similarity obtained by K nearest neighbors is not high. The idea behind this paper is that we choose the quantity of that party between the item number and user number of neighbors to predict the rating, when recommending a specific project rated by the user.

The neighbor number of item or user depends on the value of the similarity. The user or the item that the similarity is greater than a threshold value is considered as a neighbor. Specifically, user neighbor is decided by  $S(u,v) > \alpha$ , where the user  $v$  is neighbor of  $u$ . Item neighbor is decided by  $S(i,j) > \alpha$ , where the item  $j$  is the neighbor of  $i$ .

The resource set visited by the user in a time window is regarded as  $A$ . The set  $A$  can reflect the recent interest of the user to some extent. If the learning resource  $i$ , such as a text, is very similar to some resources in  $A$ , it and the recent interest of the user is very relevant. Further, the resource the user interests may be similar to the resource  $i$  in the next period of time, that is, the resource  $i$  plays a very important role in recommending other resources to the user. Thus, weight function is defined as:

$$W(u,i) = \frac{\sum_{j \in A} S(i,j)}{|A|} \tag{6}$$

where  $|A|$  is the number of resources in the set  $A$ .

#### **3.2. The algorithm**

We propose an improved CF-based algorithm of learning resources by adding neighbor relationship and data weight to CF-based method. The proposed algorithm is described as follows:

Algorithm 1. Neighbor-based data weight CF recommendation algorithm.

Input: the visited resource set  $A$  and rating matrix  $R$ .

Output: top-N recommending set of the user  $u$ .

Step 1: Computing the similarity of resources  $i$  and  $j$  in terms of Equation (3);

Step 2: Computing the neighbor of users  $N_u = \{u_1, u_2, \dots, u_k\}$  and the neighbor of resources  $N_r = \{r_1, r_2, \dots, r_l\}$  according to the similarity;

Step 3: For each resource  $i$ , computing  $W(u,i)$  in terms of Equation (6);

Step 4: (1) Computing the weight recommendation of the resource  $i$  to the user  $u$  according to user neighbor:

$$E_{u,i} = \bar{R}_u + \frac{\sum_{v \in Neig(u)} W(u,v)S(u,v)(R_{v,i} - \bar{R}_v)}{\sum_{v \in Neig(u)} W(u,v)(|S(u,v)|)} \tag{7}$$

(2) Computing the weight recommendation of the resource i to the user u according to resource neighbor:

$$E_{u,i} = \bar{R}_i + \frac{\sum_{j \in Neig(i)} W(u,j)S(i,j)(R_{u,j} - \bar{R}_j)}{\sum_{j \in Neig(i)} W(u,j)(|S(i,j)|)} \tag{8}$$

Step 5: Sorting resources in the ascending manner and selecting the top N resources as the recommending set of the user u.

### 3.3. Experiment

We select the open data set to evaluate the effectiveness of the proposed approach. We use ml data set provided by MovieLens, which consists of 100000 items rated by 943 users. There are 1682 movies, in which each user rated more than 20 movies. We use 85% of the data set as training set and the rest as testing set. Also, in order to the accuracy of the experiment, we do five runs on each experiment.

Mean absolute error (MAE) is usually applied to evaluate the performance of recommendation algorithms, since it is simple and easily represented. MAE is written as [10]:

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

where  $P = \{p_1, p_2, \dots, p_n\}$  is a recommending set of the objective user in the testing set and  $Q = \{q_1, q_2, \dots, q_n\}$  is the real rating set of the objective user. The smaller the MAE value, the higher the recommending accuracy.

We compare the proposed algorithm with three representative clustering algorithms as follows:

- (1) CAND [5], a CF recommendation algorithm based on neighbor decision-making;
- (2) UFAM [7], a user credit-based collaborative filtering algorithm;
- (3) CRAI [10], a CF recommendation algorithm incorporated with user interest change.

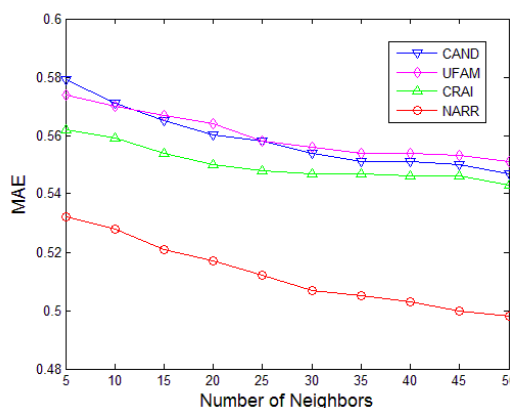


Figure 2. The results obtained by four algorithms.

In the experiment, we select the number of neighbors from the interval [5, 50]. Experimental results are shown in Figure 2 and Table 1. We see from those results that MAE gradually descends as the number of neighbors ascends. It can be observed that the proposed algorithm has better accuracy compared to other three algorithms in the case of less number of neighbors. In fact, when the number of neighbors ascends, the accuracy of NARR still has a dominant position.

Table 1. Comparison of four algorithms on different neighbours

Neighbours	CAND	UFAM	CRAI	NARR
8	0.5787	0.5749	0.5621	0.5307
16	0.5663	0.5682	0.5573	0.5201
24	0.5627	0.5633	0.5560	0.5117
32	0.5614	0.5625	0.5568	0.5055

#### **IV. Conclusions**

In the paper, we represent the theoretical analysis of CF-based recommendation approaches. Also, we describe the existing CF-based recommendation approaches which suffer from data sparseness and dynamic changes of user interest. In order to address the issues, we present a new algorithm titled as neighbor-based data weight CF recommendation of learning resources (NARR). Specifically, we choose the neighbor of the user or the neighbor of the resource according to the rating matrix. Next, we compute data weight through the change of the user's interests. Finally, we apply neighbor relationship and data weight to CF-based recommendation algorithm to select recommended resources for the user. Experiment results show that the proposed algorithm can obtain better accuracy compared to other related algorithms. The next step is to apply the algorithm to a specific system to test actual operating results of the proposed algorithm.

#### **Acknowledgment**

This research is respectively supported by Modern Education Technology Center Key Foundation of Zhejiang Province under Grant No. JA017 and Humanity and Social Science Planning foundation of Ministry of Education of China under Grant No. 13YJA880095.

#### **References**

- [1] J. Bobadilla, F. Serradilla, A. Hernando, MovieLens, "Collaborative filtering adapted to recommender systems of e-learning", *Knowledge-Based Systems*, vol. 22, pp. 261-265, March 2009.
- [2] J. Bobadilla, A. Hernando, F. Ortega, J. Bernal, "A framework for collaborative filtering recommender systems", *Expert Systems with Applications*, vol. 38, pp. 14609-14623, 2011.
- [3] M. Wu, C. Chang, R. Liu, "Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices", *Expert Systems with Applications*, vol. 41, no. 6, pp. 2754-2761, 2014.
- [4] A. Bilge, H. Polat, "A comparison of clustering-based privacy-preserving collaborative filtering schemes", *Applied Soft Computing*, vol. 13, no. 5, pp. 2478-2489, 2013.
- [5] C. Birtolo, D. Ronca, "Advances in Clustering Collaborative Filtering by means of Fuzzy C-means and trust", *Expert Systems with Applications*, vol. 40, no. 17, pp. 6997-7009, 2013.
- [6] X. Yang, Y. Guo, Y. Liu, H. Steck. "A survey of collaborative filtering based social recommender systems", *Expert Systems with Applications*, 2013, in press.
- [7] X. Su, T.M. Khoshgoftaar. "A survey of collaborative filtering techniques", *Advances in Artificial Intelligence*, vol. 3, 2009.
- [8] C. Li, Z. Zhu, X. Gao, Y. Chen. "Collaborative Filtering Recommendation Algorithm Based on Neighbor Decision-making", *Computer Engineering*, vol. 13, pp. 34-37, 2010.
- [9] P. Symeonidis, A. Nanopoulos, Y. Manolopoulos. "Providing justifications in recommender systems." *IEEE Transactions on Systems, Man and Cybernetics*, vol.6, pp. 1262-1272, 2008.
- [10] J. Buhwan, L. Jaewook, C. Hyunbo. "User credit-based collaborative filtering", *Expert Systems with Applications*, vol. 36, pp. 7309-7312, 2009.
- [11] P. Symeonidis, A. Nanopoulos, A. Papadopoulos, Y. Manolopoulos. "Collaborative recommender systems: Combining effectiveness and efficiency", *Expert Systems with Applications*, vol. 34, pp. 2995-3013, 2008.
- [12] S. Shinde, U. Kulkarni. "Hybrid personalized recommender system using centering-bunching based clustering algorithm", *Expert Systems with Applications*, vol. 39, pp. 1381-1387, 2010.
- [13] C. Xing, F. Gao, S. Zhan, L. Zhou. "Collaborative Filtering Recommendation Algorithm Incorporated with User Interest Change", *Journal of Computer Research and Development*, vol. 2, pp. 296-301, 2007.