A New Collaborative Filtering Approach Utilizing Item's Popularity

Weiwei Xia, Liang He, Lei Ren, Meihua Chen, Junzhong Gu
Department of Computer Science and Technology, East China Normal University, Shanghai, China

Abstract - **Collaborative filtering (CF) is one of the most successful technologies in recommender systems, and widely used in many personalized recommender areas, such as e-commerce, digital library and so on. However, most collaborative filtering algorithms suffer from data sparsity which leads to inaccuracy of recommendation. In this paper, we focus on nearest-neighbor CF algorithms and propose a new collaborative filtering approach. First, we suggest a new missing data making up strategy before user's similarity computation, which smoothes the sparsity problem. Meanwhile, the notion of item's popularity weight is defined and introduced into the computation. After then, when facing with new users, we also find a kind way to alleviate the difficulty in recommendation. The experimental results show our proposed approach outperforms the other existing collaborative filtering algorithms. It can efficiently smooth the inaccuracy caused by ratings sparsity, and can work well in generating recommendation for new users.**

Keywords – **Collaborative Filtering, Recommender System, Item's Popularity Weight, Sparsity Problem**

I. INTRODUCTION

Since the beginning of the 1990s, with the widespread Internet, Web has become an important way to access information. But on the other side, with the increasing web information, we have to spend much time on finding the interesting content we just need. Personalized recommendation has emerged in response to the information overload problem. It can understand customer tastes by analyzing user activities on the platform, then recommend content tailored to user preferences. At present, personalized recommendation technology is popular in many fields, such as e-commerce, digital library, news sites, entertainment sites and IPTV.

Collaborative filtering is one of the most successful technologies in recommender systems [1,2]. There are many famous sites using CF to develop their personalized recommendation, such as WebWatcher [3], GroupLens [4], Firefly [5], SELECT [6], and SiteSeer [7]. CF is based on the truth of friends' recommendation or "word of mouth". The main idea is that the target user is likely to enjoy the items which other users with common interests. This methodology can recommend information for users only according to user-item ratings data, without taking content's detail into account. Especially, it has proved to be very successful in multi-value rating data domain [8].

This study was fully supported by the grants of "Shanghai Key Technology R&D Project" (06DZ15008) and "Shanghai Science and Technology Professional Project" (07QB14036).

But there are several weaknesses in CF process still unsettled, such as sparsity problem and new user problem, which result in the inaccurate recommendation. In addition, because finding user's neighbors is key step of CF, it's important to accurately measure user's similarity. But we also find that the existing methods seldom think about the items with different popularity may reflect different importance in computation. For example, a popular film "Gone with the wind" and an unpopular film "Autumn Spring" indicate different weight in evaluating user's similarity. So in computing user's similarity process in CF, it's necessary to take the factor of item's popularity into account.

In this paper, we focus on nearest-neighbor CF algorithms. First, we propose a new missing data making up strategy before computing user's similarity. This method can smooth the sparsity problem to a certain extent. Second, we define item's popularity weight and introduce it into the computation of similarity. The result will be accurate to generate user's true neighbors. Finally, in the prediction and recommendation step, we try a new method for new users by defining item's composite recommendation value (ComRV), by which whether an item is recommended or not is decided. All of these contribute to more accurate recommendation. We have carried out experimental evaluation, which takes into account the above new aspects. The result demonstrates the superiority of our method, which achieves about 20% improvements of precision over existing CF algorithms.

The rest of this paper is arranged as follows: In the following section, collaborative filtering and its related research are introduced. In section 3, we present our proposed approach in detail. Section 4 presents the experimental evaluation. Finally, in section 5 we summarize our discussion and provide directions for future research.

II. BACKGROUND

In this section, we review traditional collaborative filtering algorithms and several major improvements in collaborative filtering. We also analyze the problem in traditional work and clarify our contributions in CF process on the existing problems.

A. Collaborative Filtering

978-1-4244-2630-0/08/$25.00 ©2008 IEEE

Collaborative filtering [9] is a technique of using peer opinions to predict the interests of others. It has three basic steps as follows:

- Obtain User-Item Ratings Data

System collects the explicit or implicit interests information of users. After preprocess and normalize the original information, we can form a rating matrix, in which user's personalized preferences is implied.

- Compute Similarity and KNN Selection

Based on the former matrix, we should find the most similar users or items of the target user or target item. They are respectively called user-based and item-based CF. Computing similarity is the key step in CF. Weather the neighbors are selected rightly based on the results will directly affect the quality of recommendation.

Two commonly used algorithms are the Pearson Correlation Coefficient (PCC) algorithm [10] and the Vector Space Similarity (VSS) algorithm [11]. These two approaches differ in the computation of similarity. As described in [11], the PCC algorithm generally achieves higher performance than VSS method.

- Predict Missing Data and Generate Recommendation

Finally, we can use the neighbors' preferences to do a prediction for the target user on the non-rated items, and select the items with high scores as recommendations. And here CF comes to an end.

B. *Other Related Work*

To alleviate the data sparsity problem, [12] proposed to reduce the dimensionality of recommender system database using the method called Singular Value Decomposition (SVD) and then users have ratings on every item on reduced dimension. But [13] indicated that this method may lead to loss of information and its effects are very data dependent. In high dimension domain, dimensionality reduction can't achieve good performance. Reference [14] proposed a generative probabilistic framework to exploit more of the data available in the user-item matrix by fusing all ratings with a predictive value for a recommendation to be made. However, methods mentioned above can't get high-quality recommendation.

After Herlocker [10,15] proposed to add a correlation significance weight in computing similarity to balance the affects of rating's sparsity. Reference [16] used the enhanced Pearson Correlation Coefficient algorithm by adding one parameter which overcomes the potential decrease of accuracy when computing the similarity of users or items. This parameter did a little change on Herlocker's significance weight to bounds the similarity to the interval [0,1]. Reference [17] provided a thorough analysis of factors involved in CF and then proposed several extensions and new approaches, which greatly improved the entire CF approaches.

To improve the accuracy, another way is to set a threshold in the selection of nearest neighbors, rather than to select based on a certain number of nearest neighbors. Reference [16] also introduced a user threshold and an item threshold to overcome the flaws of Top-N neighbors' selection. It achieved more accurate predictions for the active user, and then made more precise recommendations for him.

C. *Existing Problems and Our Contributions*

- Sparsity problem – employing a new making up strategy

Almost all CF Algorithms are confronted with sparsity problem. Because the ratings data are not sufficient, the similarity computation isn't so accurate and leads to a serious degradation of recommendation quality. We proposed a novel ratings data making up strategy better utilizing the item's category information. After the making up, according to the modified ratings data, the following calculation will be well-founded and the result will be more accurate. It also solves new item problem to a certain extent.

- Neglect items' difference – utilizing item's popularity weight in similarity computation

The existing CF algorithms didn't think of the different significance weights of items with different popularity in correlation computation. So based on this truth, we define a notion of item's popularity, and introduce it as an importance weight in the successive computation, which leads to more accurate predictions and recommendations.

- New user problem – alleviating with ComRV

When a new user comes, there are no ratings data or small number ratings about him, so it's hard to form his neighbors and recommend the proper items for him. In the recommendation step, when facing with new users, we give a composite recommendation value which combines the popularity and mean rating value of item. The items with the Top-N highest ComRV will be recommended to new users and satisfy them.

III. PROPOSED METHODOLOGY WITH SMOOTHING

In the sequel, we describe our proposed approach in detail. First, we define the notations that are used throughout this paper. Let $U = \{u_1, u_2, \cdots, u_m\}$ be a set of users, $T = \{t_1, t_2, \cdots, t_n\}$ be a set of items in database; $A(m,n)$ is the rating matrix, and $R_{i,j}$ indicates that item j is rated as $R_{i,j}$ by the user i (here $1 \le i \le m$, $1 \le j \le n$); u_a is an active user—the user for whom we need to provide recommendations of the items that he hasn't seen before; $U(t)$ denotes the users set in which every user has rated the item t; $T(u)$ denotes the items set in which every item has been rated by user u; $\overline{R_u}$ denotes user u's average rating; The rating scale goes from 1 to

$r_{\max}$; P_t denotes the popularity value of item *t*; and w_t denotes the item's popularity significance weight of item *t*.

A. Item's Popularity Computation

As we know, how many times the item has been rated can reflect weather it is popular or not. The rated number is larger, it is more popular. So it's simple to get every item's popularity value just via scanning every column of $A(m,n)$. It represents like this: $P_t = |U(t)|$.

B. User's Similarity Computation and KNN Selection

a) Ratings data making up

Because of the extreme sparsity of ratings data, the traditional similarity computing methods can't rightly select the nearest neighbors of the active user, and it's hard to generate the high-quality recommendation for him. This paper gives a new strategy to make up the sparse ratings, which makes the common rated items more between every two users. Steps are as follows:

Step 1: Calculate the union set $T(a,u)$ of the items voted by user a or user u , that is $T(a,u) = T(a) \cup T(u)$.

Step 2: For user a , find the item set he hasn't rated in $T(a,u)$, we denote it as $N(a)$, $N(a) = T(a,u) - T(a)$, then make up the missing data on every item j in $N(a)$. The strategy is like this: ~~First, get the category information of item *j*. Then calculate *a*'s mean rating value on the items he has rated in this category. Here we should note that if he has never rated any item in this category, we use the mean rating value on all items he has rated instead.~~ Finally, use the result as the making up data. Likewise, for user *u*, we make up the missing data on every item in $N(u)$. After this done, the number of common rated items by *a* and *u* becomes more.

b) Similarity computation utilizing item's popularity

In the existing similarity measures, neither PCC nor VSS thinks of the pop items and the unpopular items reflect different weight in similarity computation. All items make no difference in these measures. But the truth is that the universally rated items are not as useful in capturing similarity as less common items. So after the former step *A*, we define the popularity significance weight of item *t* as follows:

$$w_t = \log(m/P_t) \tag{1}$$

Here, *m* is the total number of users in the database. Via the above definition, this weight accords with the fact that more popular items reflect lower significance and less popular items contrarily reflect higher significance.

After then, we introduce this weight in PCC measure. The similarity calculation is modified as follows:

$$sim(a,u) = \frac{\sum_{t \in T(a,u)} w_t^2 (R_{a,t} - \overline{R_a})(R_{u,t} - \overline{R_u})}{\sqrt{\sum_{t \in T(a,u)} w_t^2 (R_{a,t} - \overline{R_a})^2} \sqrt{\sum_{t \in T(a,u)} w_t^2 (R_{u,t} - \overline{R_u})^2}} \tag{2}$$

c) K-nearest neighbors' selection

After computing all similarity values between the active user *a* and every other user in the database, we sort the results in descending order, and form a neighbor set $KNN(a) = \{knn_1, knn_2, \cdots, knn_k\}$, here $a \notin KNN(a)$, and $sim(a, knn_1) \geq sim(a, knn_2) \geq \cdots \geq sim(a, knn_k)$.

C. Prediction and Recommendation for Active Users

The predictions on the non-rated items for the active user are computed by the neighbors' ratings data on those items.

$$P(a,i) = \overline{R_a} + \frac{\sum_{u \in KNN(a)} sim(a,u) \cdot (R_{u,i} - \overline{R_u})}{\sum_{u \in KNN(a)} sim(a,u)} \tag{3}$$

Then generate the recommendation items list for user *a* based on the Top-N highest prediction value.

In this step, we find that if there is a new user first using the system, it's hard to recommend the proper items for him, because of no rating data from him. How to deal with this situation? So we try a novel way. We make statistics to get Top-N most popular items and which are also rated well. We define a composite recommendation value for every item *t* denoted as $ComRV(t)$. It is calculated by the equation as follows:

$$ComRV(t) = P_t \cdot \overline{R_t} \tag{4}$$

Here, P_t is the popularity, and $\overline{R_t}$ is the mean rating value of item *t.* Then select the items with the Top-N highest ComRV as the positive items for the new user and recommend them. The goal of this way is to improve the quality of recommendation for new users, trying best to recommend the items he possibly likes.

IV. EXPERIMENTAL ANALYSIS

We conduct following experiments to examine the effectiveness of our new approach for collaborative filtering with other methods, and address the following issues: (1) How does the neighbor size affect the accuracy of recommendation? (2) How does our method compare with traditional user-based CF, item-based CF and other existing approaches? (3) How do the recommendations produced by our proposed ComRV satisfy new users? In section A and B, we introduce the dataset and metrics for the experiment, then we present the experimental results and give an analysis in section C.

A. Dataset

One movie rating datasets called MovieLens (http://www.cs.umn.edu/Research/Grouplens/) are applied in our experiments. It provides two datasets. The first one contains 100,000 ratings (1-5 scales) rated by the 943 users on 1682 movies, and each use at least rated 20 movies. The other contains 1,000,000 ratings (1-5 scales) rated by 6040 users on 3900 movies. We use the first one in our experiments and extract 31468 rating records by

300 users on 1533 movies, sparsity is given as: $1-31468/(300\times1533)=93.16\%$.

We also get the movie-category matrix from attribute description file of 1682 movies. They array as follows: Unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western and other special information of movies.

B. *Evaluation Metrics*

We use the Mean Absolute Error (MAE) metrics to measure the prediction quality of our proposed approach with other collaborative filtering methods. If prediction ratings set of N users is $\{m_1, m_2, \cdots, m_N\}$, and corresponding true ratings set is $\{n_1, n_2, \cdots, n_N\}$, MAE is defined as:

$$MAE = \frac{\sum_{i=1}^{N} | m_i - n_i |}{N} \quad (5)$$

Lower the MAE, higher the precision of prediction and recommendation.

C. *Experimental Results*

Firstly, we need some regulations in experiments: if the prediction $P<1$, choose 1 as the result, else if $P>5$, choose 5 as the result. We do the experimental evaluation as follows.

a) Neighborhood size effect

The neighborhood size has a significant effect on the prediction quality. We performed an experiment where we varied the number of neighbors by step of 5 from 10 to 50. The results are shown in Fig.1, from which we can observe that in the beginning, as the number of neighbors is increasing, the MAE is lower, but after 30 neighbors are selected, there is no considerable improvement in quality, so we select 30 as our optimal choice of neighborhood size.

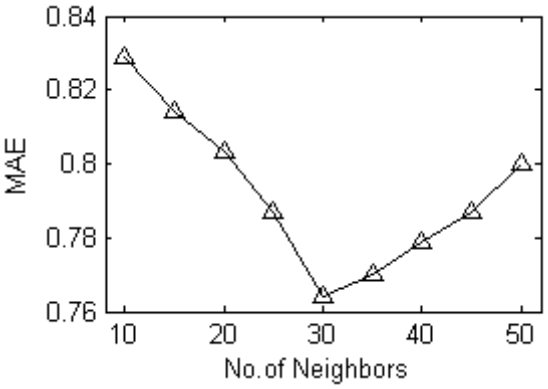

Fig. 1. Neighborhood size's impact on MAE.

b) Comparison

We select 300 users ratings from datasets, and from which we select 100, 200, 300 users respectively as training set. As to the active users, we vary the number of rated items provided by the active users from 5, 10, 15, to 20, and give the name of Given5, Given10, Given15 and Given20. Set neighborhood size as 30, the result can be seen in Table 1, Fig.2, Fig.3 and Fig.4, in which CFIP is our proposed approach, UPCC is user-based CF, IPCC is item-based CF and EMDP is the method proposed in [16] (parameters value in this method are set as $\lambda=0.7$, $\gamma=30$, $\delta=25$, $\eta=\theta=0.4$).

TABLE I
MAE COMPARISONS WITH OTHER METHODS

Training Set	Methods	Given5	Given10	Given15	Given20
ML_100	IPCC	0.892	0.850	0.838	0.826
	UPCC	0.876	0.846	0.831	0.812
	EMDP	0.834	0.826	0.813	0.790
	CFIP	**0.829**	**0.818**	**0.804**	**0.771**
ML_200	IPCC	0.855	0.834	0.821	0.812
	UPCC	0.844	0.822	0.816	0.808
	EMDP	0.833	0.820	0.811	0.781
	CFIP	**0.825**	**0.813**	**0.786**	**0.753**
ML_300	IPCC	0.851	0.829	0.813	0.807
	UPCC	0.840	0.816	0.809	0.800
	EMDP	0.828	0.816	0.802	0.774
	CFIP	**0.797**	**0.778**	**0.757**	**0.719**

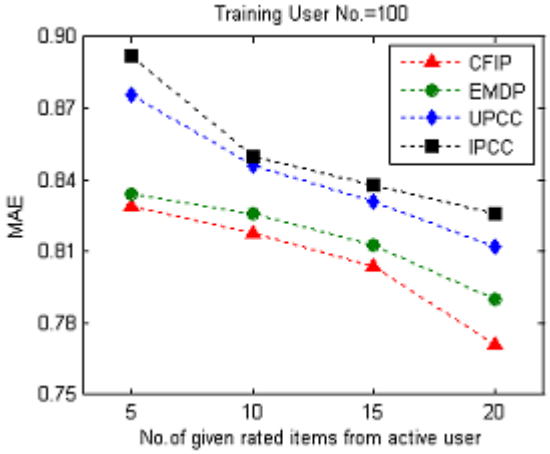

Fig. 2. MAE comparison with other methods with Training Users=100.

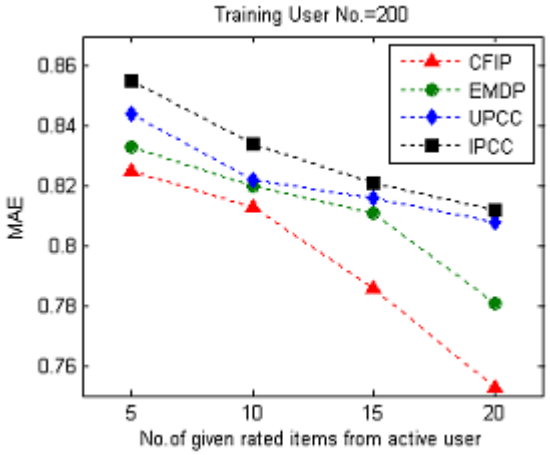

Fig. 3. MAE comparison with other methods with Training Users=200.

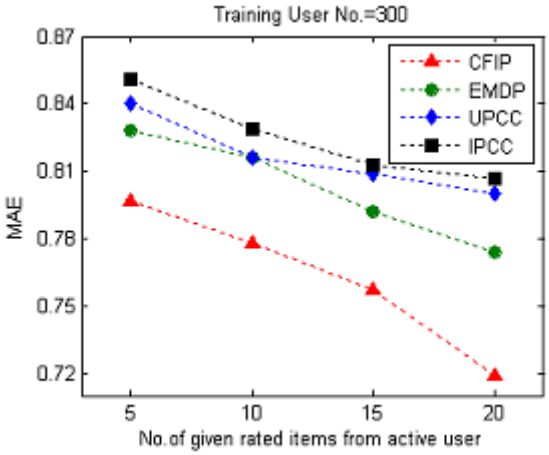

Fig. 4. MAE comparison with other methods with Training Users=300.

From the table and figures, we observe that our new approach significantly improves the recommendation quality of collaborative filtering, and outperforms UPCC, IPCC and EMDP. We also find that the larger the numbers of training users and active users' rated items are, the more accurate the prediction is.

c) Impact on new user

We randomly select some users from training set, and wipe off their all rating records. Then they are same as new user. Via computation the item's ComRV to generate the recommendation for them. We set these recommended items a rating prediction as 4 and set N in Top-N as 20, then compare with their true ratings in the test set and calculate the MAE. It is about 0.952, which proves this novel measure is an effective way of recommending for new users.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we try a novel strategy of missing data making up, which alleviates the sparsity problem by making every two users have more common rated items. Furthermore, we propose an item's popularity weight in similarity computation. And in the recommendation step, dealing with new user problem, we give a new evaluation about items, which is Composite Recommendation Value. Based on it, it's simple and effective to find the proper items for new users. Through the experimental analysis, our approach proves to be an outstanding way in CF. It outperforms other existing CF methods in accuracy of recommendation. And it also does a good job when up to new users. Our proposed approach can be applied not only in movie or IPTV program recommendation, but also in E-commerce, such as recommender systems for online bookstore.

For future work, we will conduct more research on following parts: (1) How to describe and sort the items resource with better methods, for example, metadata technology. Then we can compute the correlation between items more accurately. (2) How to capture users' feedback in direct or indirect ways for better understanding users' interests and better recommendation quality.

REFERENCES

[1] G. Adomavicius, A. Tuzhilin, "Toward the next generation of recommender system: A survey of the state-of-art and possible extensions," in *IEEE Trans on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[2] G.Karypis, "Evaluation of item-based top-N recommendation algorithms," in *Proc. of CIKM 2001*, Atlanta, Georgia, USA, pp. 247–254.

[3] T. Joachims, D. Freitag, and T. Mitchell, "WebWatcher: a tour guide for the World Wide Web," in *Proceedings of the 1997 IJCAI*, Nagoya, Japan, pp. 770–777.

[4] J. A. Konstan, B. N. Miller, D. Maltz, J. L Herlocker, L. R. Konstan, J. Riedl, "GroupLens: applying collaborative filtering to usenet news," in *Communications of the ACM*, vol. 40, no. 3, pp.77–87, 1997.

[5] U. Shardanand, P. Maes, "Social information filtering: algo rithms for automating word of mouth," in *Proc. of the ACM CHI'95 Conference on Human Factors in Computing Systems*, Denver, Colorado, pp. 210–217.

[6] R. Alton-Scheidl, J. Ekhall, O. van Geloven., L. Kovacs, "SELECT: social and collaborative filtering of web documents and news," in *Proceedings of the 5th ERCIM Workshop on User Interfaces for All: User-Tailored Information Environments*, pp. 23–37, 1999.

[7] J. Rucker, M. J. Polanco, "Siteseer: personalized navigation for the web," in *Communications of the ACM*, vol. 40, no.3, pp. 73–75, 1997.

[8] J. L. Herlocker, J. Konstan, L. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," in *ACM Transactions on Information Systems*, vol. 22, no.1, pp. 5–53, 2004.

[9] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," in *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[10] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. of SIGIR*, Berkeley, California, pp. 230-237, 1999.

[11] G. Linden, B. Smith, J. York, "Amazon.com recommendations: Item-to-Item collaborative filtering," in *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[12] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-A case study," in *ACM WebKDD 2000 Workshop*.

[13] C. C. Aggarwal, "On the effects of dimensionality reduction on high dimensional similarity search," in *Proc. ACM PODS '01*, Santa Barbara, California, pp. 256–266.

[14] J. Wang, A. P. de Vries, and M. J. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proc. of SIGIR*, Seattle, Washington, pp. 501–508, 2006.

[15] J. L. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," in *Information Retrieval*, vol. 5, no. 4, pp. 287–310, Oct. 2002.

[16] H. Ma, I. King, and R. Lyu Michael, "Effective Missing data Prediction for Collaborative Filtering," in *Proc. of SIGIR* , Amsterdam, Netherlands, pp. 39–46, 2007.

[17] P. Symeonidis, A. Nanopoulos, A. Papadopoulos, and Y. Manolopoulos, "Collaborative Filtering Process in a Whole New Light," in *Proc. of IDEAS'06*, Delhi, India, pp. 29–36.