A Personalized Recommender Integrating Item-based and User-based Collaborative Filtering

XiaoYan Shi
Zhejiang Business Technology Institute,
Ningbo 315012, P. R. China
e-mail: shixiaoyanzjbti@163.com

HongWu Ye
Zhejiang Textile & Fashion College,
Ningbo 315211, P. R. China
e-mail: yehongwuzjbti@163.com

SongJie Gong
Zhejiang Business Technology Institute,
Ningbo 315012, P. R. China
e-mail: shixiaoyanzjbti@163.com

***Abstract*—Recommender systems employ prediction algorithms to provide users with items that match their interests. The collaborative filtering (CF) is the most popular system and the two of the most famous techniques in CF are the user-based CF (UBCF) and item-based CF (IBCF). Nevertheless each of them takes only one-directional information from the user-item ratings matrix to generate recommendations. In other words, the UBCF utilizes user similarities and the IBCF tries to make a prediction by utilizing item similarities. It means that methods may use only half of the total information from the given data set. For completing the missing part of usable information, this paper proposes a CF algorithm integrating the UBCF and IBCF, which takes both vertical and horizontal information in the user-item matrix. It produces perdition using IBCF to form a dense user-item matrix and then recommends using UBCF based on the dense matrix. The experimental results on MovieLens dataset show that the proposed algorithm outperformed in terms of prediction accuracy and robustness to data sparseness.**

Keywords-personalized recommender; sparsity; item-based collaborative filtering; user-based collaborative filtering

I. INTRODUCTION

The large amount of information of web sites comes forth to people as the rapid growth and wide application of the Internet. The problem of obtaining needful information from such environment becomes more and more serious [1,2]. To solve the problem, various types of recommender systems have also been developed. And the collaborative filtering is becoming the most popular recommendation algorithm in the real world.

Many researchers have proposed various kinds of CF technologies to make a quality recommendation. All of them make a recommendation based on the same data structure as user-item matrix having users and items consisting of their rating scores. There are two methods in CF as UBCF and IBCF [1]. UBCF assumes that a good way to find a certain user's interesting item is to find other users who have a similar interest. So, at first, it tries to find the user's neighbors based on user similarities and then combine the neighbor users' rating scores, which have previously been expressed, by similarity weighted averaging. And IBCF fundamentally has the same scheme with UBCF. It looks into a set of items; the target user has already rated and computes how similar they are to the target item under recommendation. After that, it also combines his previous preferences based on these item similarities. The challenge of these two CF as following [3,4]:

Sparsity: Even as users are very active, there are a few rating of the total number of items available in a database. As the main of the CF algorithms are based on similarity measures computed over the co-rated set of items, large levels of sparsity can lead to less accuracy.

Scalability: CF algorithms seem to be efficient in filtering in items that are interesting to users. However, they require computations that are very expensive and grow non-linearly with the number of users and items in a database.

Cold-start: An item cannot be recommended unless it has been rated by a number of users. This problem applies to new items and is particularly detrimental to users with eclectic interest. Likewise, a new user has to rate a sufficient number of items before the CF algorithm be able to provide accurate recommendations.

In allusion to use the user-item rating matrix in one direction, in this paper, we proposed to use the matrix twice, horizontally and vertically, that is, make two-way predictions. Sometimes one prediction looks more reliable than the other, and vise versa, according to some criteria. In this case we can make our final prediction based on these two, which have basically orthogonal relationship each other. It produces perdition using IBCF to form a dense user-item matrix and then recommends using UBCF based on the dense matrix. The experimental results show that the proposed algorithm outperformed in terms of prediction accuracy.

II. PREDICTION USING ITEM-BASED COLLABORATIVE FILTERING

A. The sparse user-item matrix

The task of the recommendation system concerns the prediction of the target user's rating for the target item, based on the users ratings on observed items. Each user is represented by item-rating pairs, and can be summarized in a user-item table, which contains the ratings Rij that have been provided by the ith user for the jth item, the table as following.

978-0-7695-3560-9/08 $25.00 © 2008 IEEE
DOI 10.1109/ISBIM.2008.191

Table1 user-item ratings table

Item / User	Item1	Item2	… …	Itemn
User1	R1,1	R1,2	… …	R1,n
User2	R2,1	R2,2	… …	R2,n
… …	… …	… …	… …	… …
Userm	Rm,1	Rm,2	… …	Rm,n

Where Rij denotes the score of item j rated by an active user i. If user i has not rated item j, then Rij =0. The symbol m denotes the total number of users, and n denotes the total number of items.

But in the real world, the user-item rating matrix is very sparse. So, we utilize the item-based CF to form the dense user-item matrix.

B. Measuring the item rating similarity

There are several similarity algorithms that have been used[5,6]: Pearson correlation, cosine vector similarity, adjusted cosine vector similarity, mean-squared difference and Spearman correlation.

Pearson's correlation, as following formula, measures the linear correlation between two vectors of ratings as the target item t and the remaining item r.

$$sim(t,r) = \frac{\sum_{i=1}^{m}(R_{it} - A_t)(R_{ir} - A_r)}{\sqrt{\sum_{i=1}^{m}(R_{it} - A_t)^2 \sum_{i=1}^{m}(R_{ir} - A_r)^2}}$$

Where R_{it} is the rating of the target item t by user i, R_{ir} is the rating of the remaining item r by user i, A_t is the average rating of the target item t for all the co-rated users, A_r is the average rating of the remaining item r for all the co-rated users, and m is the number of all rating users to the item t and item r.

The cosine measure, as following formula, looks at the angle between two vectors of ratings as the target item t and the remaining item r.

$$sim(t,r) = \frac{\sum_{i=1}^{m} R_{it} R_{ir}}{\sqrt{\sum_{i=1}^{m} R_{it}^2 \sum_{i=1}^{m} R_{ir}^2}}$$

Where R_{it} is the rating of the target item t by user i, R_{ir} is the rating of the remaining item r by user i, and m is the number of all rating users to the item t and item r.

The adjusted cosine, as following formula, is used for similarity among items where the difference in each user's use of the rating scale is taken into account.

$$sim(t,r) = \frac{\sum_{i=1}^{m}(R_{it} - A_i)(R_{ir} - A_i)}{\sqrt{\sum_{i=1}^{m}(R_{it} - A_i)^2 \sum_{i=1}^{m}(R_{ir} - A_i)^2}}$$

Where R_{it} is the rating of the target item t by user i, R_{ir} is the rating of the remaining item r by user i, A_i is the average rating of user i for all the co-rated items, and m is the number of all rating users to the item t and item r.

C. Selecting the target item neighbors

Select of the neighbors who will serve as recommenders. Two techniques have been employed in recommender systems:

(a) Threshold-based selection, according to which items whose similarity exceeds a certain threshold value are considered as neighbors of the target item.

(b) The top-n technique in which a predefined number of n-best neighbors is selected.

D. Prediction using item-based CF

Since we have got the membership of item, we can calculate the weighted average of neighbors' ratings, weighted by their similarity to the target item.

The rating of the target user u to the target item t is as following:

$$P_{ut} = \frac{\sum_{i=1}^{c} R_{ui} \times sim(t,i)}{\sum_{i=1}^{c} sim(t,i)}$$

Where Rui is the rating of the target user u to the neighbour item i, sim(t, i) is the similarity of the target item t and the neighbour item i, and c is the number of the neighbours.

III. RECOMMENDER USING USER-BASED COLLABORATIVE FILTERING

Through the calculating the vacant user's rating by item-based CF algorithm, we gained the complete users' ratings. Then, to generate prediction of a user's rating, we use the user-based collaborative filtering algorithms.

A. The dense user-item matrix

After we use the item-based CF, we gained the complete ratings of the users to the items. So, the original sparse user-item rating matrix is now becoming the dense user-item matrix.

B. Measuring the user rating similarity

We also use the Pearson correlation measurement to compute the users' similarity, as following formula.

Pearson's correlation, as following formula, measures the linear correlation between two vectors of ratings.

$$sim(i,j)=\frac{\sum_{c\in I_{ij}}(R_{i,c}-A_i)(R_{j,c}-A_j)}{\sqrt{\sum_{c\in I_{ij}}(R_{i,c}-A_i)^2\sum_{c\in I_{ij}}(R_{j,c}-A_j)^2}}$$

Where $R_{i,c}$ is the rating of the item c by user i, A_i is the average rating of user i for all the co-rated items, and I_{ij} is the items set both rating by user i and user j.

C. Selecting the target user neighbors

Select of the neighbors who will serve as recommenders. Two techniques have been employed in recommender systems:

(a) Threshold-based selection, according to which users whose similarity exceeds a certain threshold value are considered as neighbors of the target user.

(b) The top-n technique in which a predefined number of n-best neighbors is selected.

D. Recommender using user-based CF

Since we have got the membership of user, we can calculate the weighted average of neighbors' ratings, weighted by their similarity to the target user.

The rating of the target user u to the target item t is as following:

$$P_{ut}=A_u+\frac{\sum_{i=1}^{c}(R_{it}-A_i)*sim(u,i)}{\sum_{i=1}^{c}sim(u,i)}$$

Where A_u is the average rating of the target user u to the items, Rit is the rating of the neighbour user i to the target item t, Am is the average rating of the neighbour user i to the items, sim(u, i) is the similarity of the target user u and the neighbour user i, and c is the number of the neighbours.

IV. Experimental evaluation and results

A. Data set

For the experiment, we use MovieLens collaborative filtering data set to evaluate the performance of proposed algorithm. MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. The historical dataset consists of 100,000 ratings from 943 users on 1682 movies with every user having at least 20 ratings. Therefore the lowest level of sparsity for the tests is defined as 1 – 100000/943*1682=0.937. The ratings are on a numeric five-point scale with 1 and 2 representing negative ratings, 4 and 5 representing positive ratings, and 3 indicating ambivalence.

B. Performance measurement

The metrics for evaluating the accuracy of a prediction algorithm can be divided into two main categories [7,8]: statistical accuracy metrics and decision-support metrics. Statistical accuracy metrics evaluate the accuracy of a predictor by comparing predicted values with user provided values. Decision-support accuracy measures how well predictions help user select high-quality items. In this paper, we use decision-support accuracy measures.

Decision support accuracy metrics evaluate how effective a prediction engine is at helping a user select high-quality items from the set of all items. The receiver operating characteristic (ROC) sensitivity is an example of the decision support accuracy metric. The metric indicates how effectively the system can steer users towards highly-rated items and away from low-rated ones. We use ROC-4 measure as the evaluation metric. Assume that p1, p2, p3, ..., pn is the prediction of users' ratings, and the corresponding real ratings data set of users is q1, q2, q3, ..., qn. See the ROC-4 definition as following:

$$ROC\text{-}4=\frac{\sum_{i=1}^{n}u_i}{\sum_{i=1}^{n}v_i}$$

$$u_i=\begin{cases}1, & p_i\geq 4 \text{ and } q_i\geq 4\\ 0, & otherwise\end{cases}$$

$$v_i=\begin{cases}1, & p_i\geq 4\\ 0, & otherwise\end{cases}$$

The larger the ROC-4, the more accurate the predictions would be, allowing for better recommendations to be formulated.

C. Comparing the proposed CF with the traditional CF

We compare the proposed CF that combining the user-based CF and the item-based CF with the traditional CF. The result is that our proposed is better than the traditional CF which includes the decision support accuracy metrics of ROC-4 for the two comparing methods.

V. Conclusions

Collaborative filtering can divide into two main technologies as user-based CF and item-based CF. In this paper, we have proposed a cooperative prediction design between the two CF methods which have a similar prediction procedure but use different styles of information. We use the information of user-item matrix both vertically and horizontally and then combined two prediction values using UBCF and IBCF. Lastly, experimental results show that this algorithm can increase the accuracy of the predicted values, resulting in improving quality of the collaborative filtering recommender system.

References

[1] Lee, J.-S., & Olafsson, S., Two-way cooperative prediction for collaborative filtering recommendations, Expert Systems with Applications (2008), doi:10.1016/ j.eswa.2008.06.106.

[2] Songjie Gong, Hongyan Pan, Personalized Recommendation in Short Message Serivices, In: Proceeding of 2008 International Pre-Olympic Congress on Computer Science, World Academic Press, 2008, pp.69-72.

[3] Manos Papagelisa, Dimitris Plexousakis, Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents, Engineering Applications of Artificial Intelligence 18 (2005) 781–789.

[4] Songjie Gong, Chongben Huang, Employing Fuzzy Clustering to Alleviate the Sparsity Issue in Collaborative Filtering Recommendation Algorithms, In: Proceeding of 2008 International Pre-Olympic Congress on Computer Science, World Academic Press, 2008, pp.449-454.

[5] Hyung Jun Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Sciences 178 (2008) 37-51.

[6] Chong-Ben Huang, Song-Jie Gong, Employing rough set theory to alleviate the sparsity issue in recommender system, In: Proceeding of the Seventh International Conference on Machine Learning and Cybernetics (ICMLC2008), IEEE Press, 2008, pp.1610-1614.

[7] Gao Fengrong, Xing Chunxiao, Du Xiaoyong, Wang Shan, Personalized Service System Based on Hybrid Filtering for Digital Library, Tsinghua Science and Technology, Volume 12, Number 1, February 2007,1-8.

[8] SongJie Gong, The Collaborative Filtering Recommendation Based on Similar-Priority and Fuzzy Clustering, In: Proceeding of 2008 Workshop on Power Electronics and Intelligent Transportation System (PEITS2008), IEEE Computer Society Press, 2008, pp. 248-251.