Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)

A Graph Model for Recommender Systems

Hong Chen, Mingxin Gan*†, Mengzhao Song
Dongling School of Economics and Management
University of Science and Technology
Beijing, China
chenhongbeibei@126.com, Corresponding author:ganmx@ustb.edu.cn, mengzhaosong@126.com

***Abstract*—With the constant enlargement of the scope and coverage of the Internet, the traditional search algorithms just help to filter data, without considering the needs of individuals. Therefore, various recommender systems employing different data representations and recommendation methods are currently used to cope with these challenges. In this paper, inspired by the network-based user-item rating matrix, we introduce an improved algorithm which combines the similarity of items with a dynamic resource allocation process. To demonstrate its accuracy and usefulness, this paper compares the proposed algorithm with collaborative filtering algorithm using data from MovieLens, and finally verifies the results. The evaluation shows that, the improved recommendation algorithm based on graph model achieves more accurate predictions and more reasonable recommendation than collaborative filtering algorithm or the basic graph model algorithm does. Meanwhile, the algorithm can effectively mitigate the sparse of the rating matrix.**

Keywords-recommender systems;graph model; collaborative filtering; resource allocation matrix

I. INTRODUCTION

With the continuous advancement of information technology and the exponentially rapid growth of network data, information overload makes users faced with the embarrassing issue of “Resource Isotropic”. Thus it becomes a most serious challenge for the users to search for what they are really interested in. Therefore, many researchers focus on how to provide users with the personalized active information service and recommend the potentially valuable content in recent years.

In the efforts of scholars, recommendation algorithms, which is the core of recommender systems, has developed into four methods, namely collaborative filtering recommendation, content-based recommendation, graph model recommendation and hybrid recommendation [1].

Now the graph model recommendation is relatively new, which makes use of the selection relations between users and items to dynamically allocate the items’ resources. But the existing recommender systems also have their limitations. The current CF algorithm exploit all items’ joint ratings to calculate the similarity between users, but rating matrix is usually very sparse, greatly reducing the recommended accuracy [2]. The content-based algorithm has difficulty in extracting multimedia information features, and only recommends users to buy rather similar items [3]. For the sustainable development of e-commerce websites, these problems need to be solved urgently.

Sparsity is a difficult problem for several mainstream recommendation algorithms. Usually each user only gives evaluations for few items, and the rating matrix is very sparse. As a result, the user similarity is not accurately calculated. Sarwar [4], Tang [5] put forward their own effective methods to lower the sparsity. With the increase of users and resources, the scalability problem should also be considered in the system’s long-term development. Tao [6] puts forward a solution that uses the clustering algorithm to divide users into clusters. Huang [7] comes up with dynamic programming method in a more extensive data set.

While the recommendation algorithm based on graph model uses the user-item selection relations of graph structure to more effectively solve the problems above. Chen [8] puts forward a CF algorithm based on three weighted graphs, taking users, items and labels into account to solve users’ cold start problem partly. Liu [9][9] introduces degree index to regulate the algorithms’ scalability. Li [10] transforms evaluations given by users into graph to bond the similarity of users and items.

However they all haven’t reached to the unity of accuracy, diversity and novelty. Owing to this, our main focus in this paper is to do research on the graph model-based recommendation system, in which users and items’ characteristics are regarded as abstract nodes so that user-item rating matrix can be modeled as a bipartite graph model [11][11].

Therefore, the recommendation algorithm based on graph model has an advantage on solving the sparsity problem with both high accuracy and little computation time. The main contribution of this dissertation is that according to the characteristics of user-item rating matrix, an improved graph model is presented with a comprehensive consideration over user-item selection relations and the items’ similarity, which draws lessons from Pearson similarity projects them into a resource allocation matrix calculation. It verifies that this algorithm produces more accurate and reasonable recommendations.

II. METHOD

A. Problem Description and Similarity Calculation

A recommender system mainly includes three types of data. The definition of the corresponding data is as follows.

Symbols definition:

User set: Users： $U = \{U_1, U_2, \dots, U_n\}$;

Item set: Objects： $O = \{O_1, O_2, \dots, O_m\}$;

Rating set: Ratings: $R_{n\times m}$ is generally shown as a $n \times m$ matrix. However in practice the ratings is rather scarce, so in order to generate recommendations, the recommender system has to predict the null values in the rating matrix and then gives the target user a recommendation.

This paper adopts the similarity between users to do further research. Here are three basic methods of the existing nine similarity calculation methods [12]: Cosine similarity, the correlation of Pearson and Jaccard similarity.

Cosine Similarity. The similarity between users can be measured by the cosine angle of the vectors.

$$sim(U_i, U_j) = cos(\vec{a}, \vec{b}) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}|\cdot|\vec{b}|} = \frac{\sum_{k=1}^{n} R_{ik}\cdot R_{jk}}{\sqrt{\sum_{k=1}^{n} R_{ik}^2}\cdot\sqrt{\sum_{k=1}^{n} R_{jk}^2}}, \quad (1)$$

where R_{ik} and R_{jk} respectively represent the rating item O_k that user U_i and user U_j have rated.

Correlation of Pearson. If items that user U_i and user U_j both rated are expressed by I_{ij}, their similarity $sim(U_i, U_j)$ is defined as

$$sim(U_i, U_j) = \frac{\sum_{k\in I_{ij}}(R_{ik}-\overline{R_i})(R_{jk}-\overline{R_j})}{\sqrt{\sum_{k\in I_{ij}}(R_{ik}-\overline{R_i})^2}\cdot\sqrt{\sum_{k\in I_{ij}}(R_{jk}-\overline{R_j})^2}}, \quad (2)$$

where $\overline{R_i}$ and $\overline{R_j}$ respectively represent the average rating that user U_i and user U_j have given.

Jaccard Similarity. The similarity index which Jaccard put forward a hundred years ago can be defined as

$$sim(U_i, U_j) = \frac{|T(i)\cap T(j)|}{|T(i)\cup T(j)|}, \quad (3)$$

where $T(i)$ and $T(j)$ respectively represent the neighbors of user U_i and user U_j, which show the concept of degree in the graph model.

This paper uses a weighted average of the nearest neighbor scores to predict the rating, from which we can judge the Top-N recommendation. Rating prediction formula is

$$r_{ij} = k\sum_{u\in\hat{I}} sim(i,u)\cdot r_{uj} \quad (4)$$

where r_{ij} means the probable rating of item O_j that the system predicts user U_i has rated. r_{uj} means user U_i's nearest neighbors rated the item O_j. k is a normalization factor, $\mathrm{k} = 1/\sum_{u\in\hat{I}}|sim(i,u)|$, $\hat{I}$ is a set made up of users having relatively high similarity to user U_i. $sim(i,u)$ indicates the similarity between user U_i and U_u.

B. The Improved Graph Model Algorithm

The proposed graph model, which includes the nodes of users and items and the edges on behalf of the ratings users give to the items. Accordingly, this chapter will describe the basic concepts in detail, and then outline the concrete steps of improved recommendation algorithm.

The bipartite graph $\mathrm{G} = (X, E, Y)$ is shown as Fig.1.

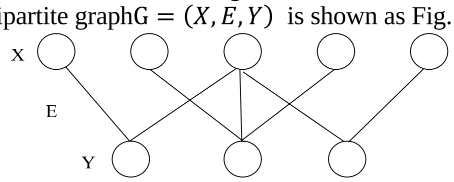

Figure 1. Bipartite graph

In Fig.1, these two-tier nodes X, Y are respectively on behalf of the item set and user set, and the edge set E means the rating set. In fact, the elements in the same layer have internal relations, namely the similarity between two users or items. Besides, the input rating matrix of recommender system will be able to use an $\mathrm{n}\times\mathrm{m}$ adjacency matrix $\{a_{ij}\}$ to describe. If user U_i collects the item O_j, $a_{ij} = 1$, otherwise, $a_{ij} = 0$.

$$a_{ij} = \begin{cases} 1, & U_i, O_j \in E; \\ 0, & Others; \end{cases} \quad (5)$$

In fact there is an assumption that the user's rating score is not less than the predetermined threshold (such as 3 points), which means the user collects the item. The principle of resource allocation is to allocate resources to all connected nodes. Tao Zhou [13] proposes a dynamic resource allocation process with three steps, after that the resources of all nodes in the graph can be represented by resource allocation matrix.

Now, consider the basic idea of the algorithm. First of all, the two vertices in the bipartite graph respectively represent the user set and item set, which can visually express the preference of users. Secondly, the initial allocation of item resources in the recommender systems with the help of the bipartite graph is described as a resource allocation matrix. Thirdly, items that have not been selected are sorted so as to generate a recommendation list. Finally, the principle of algorithm is at the same accuracy level, the shorter the length of the recommendation list is, the better; of the same preference, unpopular items have greater recommendation significance than the popular ones.

Based on the analyses above, it can be concluded that applying the similarity in users and the selection relations between users and items to the dynamic resource allocation process results in a more reasonable personalized recommendation. As a result, the main steps of the improved algorithm based on graph model are as follows:

Input: rating matrix $R_{n\times m}$ and the target user set U_i;

Output: Recommendation list matrix F;

Steps:

Step1: Build a bipartite graph. The recommender system has n users and m items. Accordingly the pre-set threshold (3 points), the $n\times m$ rating matrix R can be converted into an adjacency matrix $\mathrm{A} = \{a_{ij}\}_{n\times m}$, and then we can establish a bipartite graph model containing $\mathrm{n}+\mathrm{m}$ nodes.

Step2: Calculate the item similarity. A $m\times m$ similarity matrix $SimP$ can be obtained by computing the Pearson correlation coefficient.

Step3: Calculate resource allocation matrix W in an integrated method. The formula is as follows:

$$w_{ij} = \frac{1}{D_j}\sum_{k=1}^{n}\frac{a_{ki}\cdot a_{kj}}{D_k} \quad (7)$$

$$W = SimP.\left(w_{ij}\right)_{m\times m} \quad (8)$$

The weight w_{ij} represents the resources item O_j would like to distribute to O_i, D_j means the degree of item O_j and the matrix W represents the resource allocation matrix we are looking for.

Step4: Naturalize $W_{m\times m}$. After transformation with normalized linear function, we obtain the normalized resource allocation matrix W'.

$$W' = \left(w'_{ij}\right)_{m\times m} \tag{9}$$

$$w'_{ij} = \frac{w_{ij} - min\{w_{ij}\}}{max\{w_{ij}\} - min\{w_{ij}\}} \tag{10}$$

Step5: Compute resource allocation for the target user. Initial resource allocation for user U_i is expressed with $f_i = (a_{i1}, a_{i2}, \dots, a_{im})$, while f'_i represents the final resource allocation of m items. The formula is as follows:

$$f'_i = W \cdot f_i \tag{11}$$

Step6: Get recommendation list matrix F according to f'_i. Firstly remove the items which have been chosen by the target users, and then sort the results in descending order, finally obtain the recommendation list matrix $F_{m\times n}$.

III. NUMERICAL RESULTS

In the experimental stage, we use MovieLens data set to assess the performance of the improved recommendation algorithm based on graph model. MovieLens website data set is collected by the GroupLens Research Project Team 错误!未找到引用源。, including 1682 movies and 943 users. Actually, the rating scores of this data set are integers from 1-5, fairly dense.

The experimental data set is randomly divided into two groups, mainly training set and test set, respectively accounting for 80% and 20% of the entire data set. We test the algorithm by using two metrics: r and hit.Sorting accuracy r is a fitting degree measurement of recommendation list [14][14]. This study uses a mean sort point to describe the sorting accuracy of recommendation algorithm. The formula is as follows:

$$r_j = \frac{L_j}{N} \tag{12}$$

N represents the total number of items without rating in the training set, while $L_j$represents the predicted position of the item j in the recommendation list. The smaller r is, the more accurate the algorithm is. It is worth mentioning that the calculation of r does not require any additional parameters, and doesn't need to know the rating in advance.

For a certain recommendation list whose length is L, hitting can be described as the items tested appearing in it:

$$hitting\ rate = \frac{M}{L} \tag{13}$$

where M means hitting times. Obviously hitting rate will be more effective when comes to the length of recommendation list.

During the experiment, the described algorithms are written and carried out in MATLAB. In order to fully prove the accuracy of the improved algorithm, the paper carries out twelve kinds of configuration experiments to be comparisons, including three kinds of similarity calculation methods based on users. The evaluation of collaborative filtering recommendation experiments with rating matrix and adjacency matrix are as Table I below:

TABLE I. CF CONFIGURATION EXPERIMENTS RESULTS

	Cosine		Pearson		Jaccard	
	r	***hit***	***r***	***hit***	***r***	***hit***
0-5 rating matrix	**0.1749**	0.0517	0.1812	**0.0520**	0.1770	0.0509
0-1 adjacency matrix	0.1798	0.0515	0.1803	0.0519	0.1763	0.0509

By comparing the data in Table I, we can see that the sorting accuracy r which uses 0-5 rating matrix to calculate Cosine similarity is the best, and that hitting rate that uses 0-5 rating matrix calculating Pearson similarity is the best. Thus, it comes to a preliminary conclusion that with the same data set, Cosine and Pearson bring more accuracy in collaborative filtering than Jaccard. Furthermore, using 0-5 rating matrix as algorithm input data is more practical, and it can improve algorithm's accuracy indirectly.

Combining the experiment results in Table I and the former basic algorithm based on graph model, the improved algorithm results are shown in Fig.2. It is notable that the sorting accuracy r of the improved algorithm based on graph model is higher than others.

Through calculating, the mean r of collaborative filtering that based on similarity is 0.1777, and that of the former graph model is 0. 1060, while that of the improved algorithm based on graph model is 0.0960. So the sorting accuracy r of the improved algorithm is the smallest, that is, its accuracy is the highest. And the improved one has improved by 9.43% to the former one. To sum up, the improved graph model algorithm has large advantage over traditional collaborative filtering algorithm and former graph model algorithm.

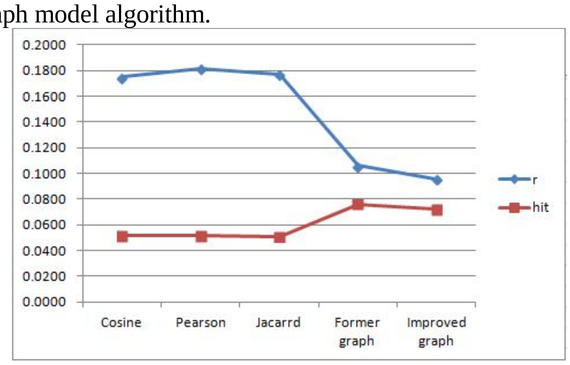

Figure 2. Comparison between the improved graph model and other algorithms

The graph model doesn't require users to give definite scores on items, which decreases an estimation that may cause error. Meanwhile, this improved algorithm has improved the calculating method of resources allocation with the help of the similarity between items, thus finding more accurate and similar neighbor items to recommend, which gives it an advantage over basic graph model algorithm in terms of ranking accuracy. Finally, we can learn from the figure that when data sets are in the same density level, other algorithms are far less accurate than graph model in the similarity calculation of the nearest neighbor, therefore the algorithm based on graph model can solve the sparsity problem to some extent.

IV. CONCLUSIONS AND DISCUSSION

This paper proposes the improved algorithm based on graph model, which needs us to make comprehensive use of similarity of items and selection relations between users and items in the process of resources allocation, and then gain a relatively reasonable recommendation list for the target users. Our several comparative tests verify the validity and accuracy of bipartite graph and the improved graph model algorithm, and main conclusions are as follows:

1) Among three methods of similarity calculation, the experiments show that Cosine and Pearson perform better than Jaccard in sorting accuracy and hitting rate.

2) Compared with 0-1 adjacency matrix, 0-5 rating matrix is more effective as a data input form of recommendation algorithm.

3) Bipartite graph model not only can represent data flexibly, but also can diminish the effect caused by data sparsity to a certain extent. It gains better recommendations.

4) The improved algorithm based on graph model makes a linear combination of the similarity in items and the user-item selection relations into resource allocation matrix, so as to generate a recommendation list that is more accurate than the former graph model algorithm and collaborative filtering algorithm. It significantly improves the effectiveness of recommender systems, that's to say, acquires better personal recommendations.

Although this paper compares the performance of graph model algorithms and other traditional collaborative filtering algorithms, there are still some limitations in this paper. When building a graph model, to simplify the model, we neglect the influence of time, for instance, without distinguishing users' long-term interests and short-term interests.

In addition, there is still a cold start problem existing, resulting from difficultly gaining the resource allocation matrix. So far, there is no algorithm found to overcome this difficulty perfectly. In future I suppose to use some characteristics or labels of new users, combining the degree of user-item relations to generate recommendation for new users.

ACKNOWLEDGMENT

This work was partly supported by the National Natural Science Foundation of China under Grants No. 71101010, and the Fundamental Research Funds for the Central Universities under Grant No. FRF-BR-11-019A.

REFERENCES

[1] Jianguo Liu, Tao Zhou, and Binhong Wang, Research process of personalized recommendation system, Progress in Natural Science, 2009, 19(1): 115.

[2] Liu Fangxian, Song Shunlin, Improved collaborative filtering recommendation algorithm, Computer Engineering and Application, 2011, 47(8): 72–75.

[3] Yan Zhu and Zenan Lin, Review of Fine personalized recommendation method in E-Commerce, China Soft Science, 2009, (2).

[4] Sarwar B, Karypis G, Konstan J A, Riedl J, Application of dimensionality reduction in recommender systems, a case study[C], 2000: 203–216.

[5] Xiaobo Tang and Jing Fan, Commercial recommendation based on customer clustering, Journal of Information, 2009, 28(6).

[6] Jianwen Tao and Hongyan Pan, Collaborative filtering algorithm based on item similarity and user rating, Information Learned Journal, 2008, 27(2): 199–204.

[7] Chuangguang Huang, Jian Yin, Jing Wang, Yubao Liu and Jiahai Wang, Collaborative filtering algorithm based on indeterminate neighbors, Chinese Journal of Computers, 2010, 33(8).

[8] Chao Chen, Yingchao Zhang and Jin Miu, Collaborative filtering algorithm based on three graph network, Journal of Ningde Teachers College, 2010, (4).

[9] Zhaoxing Liu, Ning Zhang and Jiming Li, Personalized recommendation algorithm based on collaborative filtering and network structure, Complex System and Complexity Science, 2011, 8(2).

[10] Yongli Li, Chong Wu and Kunshen Wang, Online recommendation method based on graph theory and information to maximize retention, theory and practice of systems engineering, 2011, 31(9).

[11] Zan Huang, Wingyan Chung and Hsinchun Chen, A Graph Model for E-Commerce Recommender Systems, Journal of the American Society for Information Science & Technology, 2004, 55(3): 259–274.

[12] Tao Zhou, Linyuan Lu and Yicheng Zhang, Predicting missing links via local information, the European Physical Journal B, 2009: 623–630.

[13] Tao Zhou, Jie Ren, Matus Medo and Yicheng Zhang, Bipartite network projection and personal recommendation, Physical Review,2007, 76(4).

[14] Jianguo Liu, Tao Zhou, Qiang Guo and Binhong Wang, Review of evaluation method for personalized recommendation system, Complex System and Complexity Science, 2009, 6(3).