Weight Based KNN Recommender System

Bin Wang, Qing Liao, Chunhong Zhang
Beijing Key Laboratory of Network System Architecture and Convergence
Beijing University of Posts and Telecommunications
Beijing, China
wangbinjs@126.com, {liaoqing, zhangch}@bupt.edu.cn

***Abstract*-Today, the personalized recommendation is one of the most important technologies in the Internet and e-commerce system, along with the increasing number of users and commodities. Among personalized recommendation algorithms, CF (Collaborate Filtering) has been researched for many years. The similarity computation method, which is the key in personalized recommender, like cosine theorem or pearson correlation coefficient, does not consider the distinguish degree of the items. In this paper, we will propose weight based similarity algorithm, called IR-IUF++, which updates pearson correlation coefficient. IR-IUF++ performs better than traditional similarity algorithm in our experiment.**

Keywords-Collaborate Filtering; KNN; Similarity Computation; IR-IUF++;

I. INTRODUCTION

The rapid growth of Internet and e-commerce has brought the problem of information overload: excessive information makes it difficult to pick up what we really want. It reduces the efficiency of information usage. Many searching engines or e-commerce online platforms, as shown in table 1, already start to use personalized recommender in searching results. And personalized recommendation results perform better than traditional ones, especially on e-commerce online. Recommender system is an important method of an information filter.

A recommender system needs to complete two tasks [11]: In the first period, we use recommendation algorithms and users' history logs to predict item ratings [12], which is not marked by the user before. In the second period, we generate a list of recommendation items for the user on basis of first period, and a number of factors, such as diversity, novelty, credits must be considered.

or a recommender system, the first period is key. And the core technology is recommendation algorithm. In fact, many algorithms are derived from collaborative filtering,

TABLE I. COMPANY WITH PERSONALIZED RECOMMENDATION

Field	Company
e-commerce	Amazon EBay Dangdang Taobao 360buy
music community	pandora.com last.fm Genius (iTunes)
books community	douban.com librarything.com
movie community	netflix.com movieLens
Social networking	Twitter Facebook Google+ weibo

which calculates the k-nearest neighbor(KNN). In this paper, we address a new method for getting the KNN. Comparing to former CF-KNN algorithm, it adds weight on every item, and the similarity between users will be more reliable. At last, we have conducted an experiment to test and verify this new method with movielens dataset.

II. RELATED WORKS

The recommender system is used to provide forecasts and recommendation for an item, which is shown as figure 1.

Commonly, there are three methods for the system.

Rule-Baed recommender system. In this way, we recommend the current hottest items or special items for some special days(can be used for cold users). And it is not personalized, the recommendation rules are provided by artificial.

Cluster recommendation. For example, item i and item j is of great similarity, then the user, who consume item j, is likely consumes item i.

Collaborative filtering (CF) algorithm [4]. It is proposed by Goldberg. User–Item relationship is the basis of the recommendation system. The goal of CF is to suggest the new items to users by predicting a score of an item (the score can be 0/1 or 0-5, etc.) for a particular user based on his consume log of some old items as well as the log from other users with similar tastes. "Consume" can be understood as a "read" or "buy". Memory-Based CF algorithm has two models, one is user-based, and another is item-based [3][6]. The basic idea of user-based CF is that if the user A like item a, the user B like item a, b, c, user C like item a and c, so that user A and user B and C are similar, and c is recommended to the user A. The item-based CF

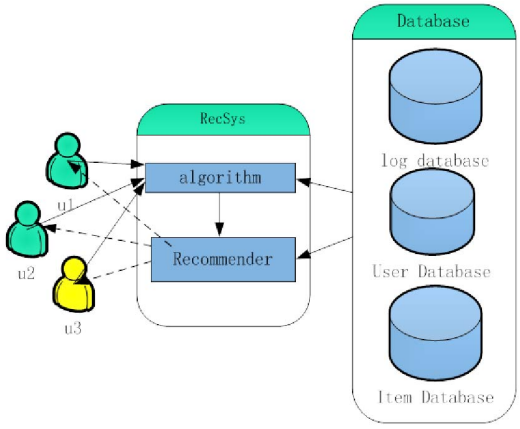

Figure1. Normal Recommender System .

978-0-7695-5011-4/13 $26.00 © 2013 IEEE
DOI 10.1109/IHMSC.2013.254

was proposed by amazon [7], its basic idea is to calculate the similarity between items with all the history of the user's preference data and the user is recommended with similar items.

Model-based CF. Models are developed using data mining, machine learning algorithms to find patterns based on training data. These are used to make predictions for real data. There are many model-based CF algorithms.

The horting model [5] is based on graph theory. The vertices are the users, and the edges represent the similarity between two users. We search for a neighbor vertices of one vertice in horting graph and then consolidate to form the final recommendations. Horting graph can skip intermediate vertice to find the nearest neighbors. It fully considers similarity relationship propagation between vertices. Therefore, the accuracy is better than KNN collaborative filtering.

The matrix factorization (MF) technique [13] is gradually applied to personalized recommendation algorithm. The MF algorithm does a global optimization on object, so MF performs better than CF in RMSE. Collaborative filtering mainly captures a small part of a strong correlation of the data and ignore the global information. Koren [9][10] introduced the matrix factorization model temporal model, which is called time SVD++ in the Netfixt data contest, significantly reducing the Root Mean Squared Error (RMSE). Xiong proposed Tensor decomposition algorithm based on Bayesian probability, the model applies to the case of shopping preferences, the experiment shows the recommendation algorithm which adding context, can improve the recommendation accuracy. Mohsen Jamali introduced MF with trust propagation. Trust propagation has been shown to be a crucial phenomenon in the social sciences in social network analysis and in trust-based recommendation. The author demonstrates that trust propagation leads to a substantial increase in recommendation accuracy, in particular for cold start users.

Singular value decomposition (SVD) is introduced to release data sparsity problem. SVD can reduce the dimension of the user-item space. The experiment results show that SVD can solve the synonym problem, and significantly improve the recommender system scalability.

However, in traditional similarity algorithm, when calculating the similarity between users, it rarely considers the global information (the distinguishing degree of items) In this paper, we will propose a new algorithm to calculate the similarity of users, which is called IR-IUF++

III. WEIGHT-BASED KNN ALGORITHM

A. Traditional KNN Personalized Recommendation

In recommender systems, we have a set of users $U = \{u_1, u_2, ..., u_n\}$ and a set of items $I = \{i_1, i_2, ..., i_n\}$. The ratings expressed by users on items are given in a rating matrix $R = Matrix[R_{ui}]_{m\times n}$, the element R_{ui} denotes the rating of user u on item i, and R_{ui} can be any real number but often ratings are integers in the range $[1;5]$. Traditional KNN-CF algorithms generally consist of the following steps:

Step1), similarity computation. Suppose R_{ui} (R_{vi}) is the value of rating by user u (v) on item i. $\overline{R}_u$ ($\overline{R}_v$) is average value of ratings by u (v) on all rated items. $I_{uv} = I_u \bigcap I_v$, where I_u (I_v) is the set of items rated by u (v). We will get the following correlation similarity,(we use the " pearson correlation coefficient [8]"):

$$Sim_{uv}(u,v)_0 = \frac{\sum_{i\in I_{uv}} (R_{ui} - \overline{R}_u)(R_{vi} - \overline{R}_v)}{\sqrt{\sum_{i\in I_{uv}} (R_{ui} - \overline{R}_u)^2}\sqrt{\sum_{i\in I_{uv}} (R_{vi} - \overline{R}_v)^2}} \quad (1)$$

Step2), neighborhood computation. For each u, find the k users that are most similar to u according to Sim_{u*}

Step3), rating prediction. Supposing the k nearest neighbors of user u is S_{knnu}, we can get rating $\hat{R}_{ui}$, which is predicted rating on item i.

$$\hat{R}_{ui} = \overline{R}_u + \frac{\sum_{v\in S_{knnu}} sim_{uv}(u,v)\times(R_{vi} - \overline{R}_v)}{\sum_{v\in S_{knnu}} |\, sim_{uv}(u,v)\,|} \quad (2)$$

The three steps are popular in recommender systems. However, in step 1, traditional similarity formula does not consider the global information of an item, the table 2 is an *User – Item* table .

With the table 2 information, we can get the similarity between U_3 and U_1 is equal to similarity between U_3 and U_2 in step1. In fact, "Xinhua Dictionary" is commonly in daily life. It may be consumed by everyone and cannot be used to distinguish the users' taste. While "Introduction to Algorithms" will not be consumed for everyone. The book can easily distinguish the users' interest. So, we propose a new algorithm for similarity computation.

B. IR-IUF++ similarity algorithm

John S.Breese [1] proposed the penalty function to punish hot items. They used the following formula.

$$Sim_{uv}(u,v)_1 = \frac{\sum_{i\in I_{uv}} (R_{ui} - \overline{R}_u)(R_{vi} - \overline{R}_v)\dfrac{1}{\log(1+N(i))}}{\sqrt{\sum_{i\in I_{uv}} (R_{ui} - \overline{R}_u)^2}\sqrt{\sum_{i\in I_{uv}} (R_{vi} - \overline{R}_v)^2}} \quad (3)$$

$N(i)$ is the total number of users who have consume item i. IR-IUF++ (Item Rating-Inverse User Frequency++) is based on TF-IDF (Term Frequency-Inverse Document Frequency).The following is IR-IUF++ algorithm.

TABLE II. USER-ITEM MATRIX EXAMPLE

User	Item/Rating
U_1	Xinhua Dictionary/5
U_2	Introduction to Algorithms/5
U_3	Xinhua Dictionary/5; Introduction to Algorithms/5

Firstly, compute η_{ui} , and $\eta_{ui} = \frac{s_{ui}}{\sum_k s_{uk}}$, where s_{uk} is rating on item k by u .

Second, compute $\mu_i = \log \frac{|Set_u|}{|N(i)|}$, where $|Set_u|$ is the number of users.

Third, get the IR-IUF value $\lambda_{ui} = \eta_{ui} \times \mu_i$

With the upper parameter, we update the similarity compution .

$$Sim_{uv}(u,v)_2 = \frac{\sum_{i \in I_{uv}} (R_{ui} - \overline{R}_u)(R_{vi} - \overline{R}_v) \times \frac{1}{1 + (\lambda_{ui} - \lambda_{vi})^2}}{\sqrt{\sum_{i \in I_{uv}} (R_{ui} - \overline{R}_u)^2} \sqrt{\sum_{i \in I_{uv}} (R_{vi} - \overline{R}_v)^2}} \quad (4)$$

We consider the variance of item rating. Indeed, the item's distinguishing degree is relevant to the item's variance. If the ratings of an item by all users are similar, this item hardly distinguishes the users interest .For example, a film is famous and good reputation, many audience give it a high ratings, but not because they like it or its type very much .

Based on the above truth, we use the variance to reveal the item's distinguishing degree .

$$\sigma_i^2 = \frac{\sum_{i=1}^{N} (R_i - \overline{R}_i)^2}{N} \quad (5)$$

where N is the total number of ratings of item i.

Combining with $Sim_{uv}(u,v)_2$, we get the final similarity formula.

$$Sim_{uv}(u,v)_3 = \frac{\sum_{i \in I_{uv}} (R_{ui} - \overline{R}_u)(R_{vi} - \overline{R}_v) \times \frac{\sigma_i^2}{1 + (\lambda_{ui} - \lambda_{vi})^2}}{\sqrt{\sum_{i \in I_{uv}} (R_{ui} - \overline{R}_u)^2} \sqrt{\sum_{i \in I_{uv}} (R_{vi} - \overline{R}_v)^2}} \quad (6)$$

IV. EXPERIMENT

We have evaluated the proposed methods on real world and synthetic data sets. The movielens dataset (100K) was collected through the MovieLens web site during the seven-month period from September 19th, 1997 through April 22nd, 1998. This data has been cleaned up – users who had less than 20 ratings or did not have complete demographic information were removed from this data set. It consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies.

A. Evaluation

Generally, there are two ways to evaluate the prediction error[2]. Supposing TestSet is the set of all pairs (u; i) in the test data. One is Mean Absolute Error (MAE), and defined as follow.

$$MAE = \frac{\sum_{(u,i) \in TestSet} |\hat{R}_{ui} - R_{ui}|}{|TestSet|} \quad (7)$$

Another is root mean square error (RMSE), it is defined as:

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in TestSet} (\hat{R}_{ui} - R_{ui})^2}{|TestSet|}} \quad (8)$$

B. Experimental Setup

We perform 5-fold cross validation in our experiments. In each fold we have 80% of data as the training set and the remaining 20% for test set. In experiment, to evaluate the performance of our method we consider two comparison partners.

- Normal CF, whose similarity formular is shown in formula(3), which does not take the global item information into account.
- Our new algorithm for similarity computation as shown in formula (6).

C. Experimental Results

In experiment, we first compare the pearson correlation coefficient(pcc) and IR-IUF++ with RMSE evaluation. The figure 2 shows the result. The x axis stands for the number of neighbors and it increases from 3 to 20. And the result shows that IR-IUF++ method has improved 3.19% than pearson correlation coefficient method.

The figure 3 shows the comparation result between pearson correlation coefficient(pcc) and IR-IUF++ with MAE evaluation .The IR-IUF++ MAE is lower 2.94% than pearson correlation coefficient MAE .

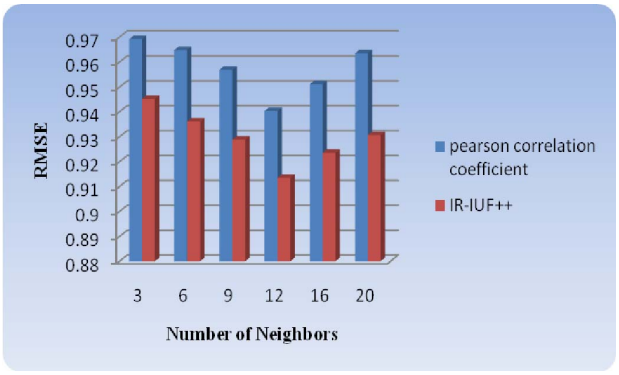

Figure2. Comparation between pcc and IR-IUF++ in RMSE

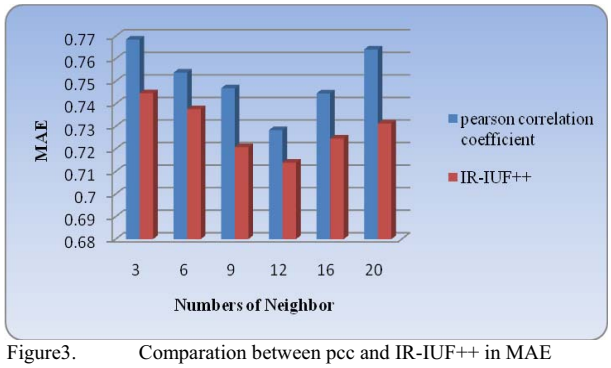

Figure3. Comparation between pcc and IR-IUF++ in MAE

V. CONCLUSION

KNN-CF is an effective way to solve the personalized recommender when the dataset is not huge. In this paper, we analysis the commonly similarity algorithm, and find the shortcoming of Pearson correlation coefficient, which does not include item's global information. Then we propose weight-based KNN recommendation algorithm. This algorithm fully considers the item's distinguish degree while computing the users similarity. It is derived from TF-IDF algorithm, and works well in the experiment. IR-IUF++ KNN can decrease by about 3% in RMSE(3.19%) or MAE (2.94%) . The items' λ_{ui} can be computed offline. It will not reduce the similarity computation speed. We believe this new user similarity method is a good choice.

REFERENCES

[1] Breese J, HechermanD, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the 14th Conference on Uncertaintyin Artificial Intelligence(UAI'98).1998, Pages 43-52.

[2] Hill W, Stead L, Rosenstein M, Furnas G. Recommending and evaluating choices in a virtual community of use. Proceedings of the CHI'95.1995,Pages 194~201.

[3] Sarwar B, Karypis G, Konstan J, Riedl J. Item-Based collaborative filtering recommendation algorithms. Proceedings of the 10th International World Wide Web Conference .2001, Pages 285~295.

[4] Sarwar B, Karypis G, Konstan J, Riedl J. Analysis of recommendation algorithms for E-commerce. ACM Conference on ElectronicCommerce.2000.Pages 158~167.

[5] Wolf J, Aggarwal C, Wu K-L, YuP. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. Proceedings of the ACM SIGMOD International Conference on Knowledge Discovery and Data Mining, SanDiego, 1999. Pages 201~212.

[6] laypool M, Goldaale A, Miranda T, Murnikov P, Netes D, Sartin M. Combining content-based and collaborative filters in an online newspaper. ACM SIGIR'99 Workshop Recommender Systems: Algorithms and Evaluation. New York : ACM press, 1999

[7] Pazzani M. A framework for collaborative, content-based, and demographic filtering. Artificial Intelligence Review,1999, 13(5-6): 393-408.

[8] Robertson S. Threshold setting and performance optimization in adaptive filtering. Information Retrieval,2002, Pages 239-256.

[9] Y.Koren. Factorization meets the neighborhood: a multifaceted collaborative ltering model. ACM SIGKDD'08, 2008, pages 426 - 434.

[10] Y.Koren.collaborative ltering with temporal dynamics. ACM KDD'09: the 15th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, 2009, pages 447-456.

[11] G. Adomavicius and A. Tuzhilin. Recommender Systems Handbook, Springer, 2010.

[12] J. Wang, A. Vries, and M. Reinders. A user-item relevance model for log-based collaborative filtering. In LNCS. In Advances in Information Retrieval. Vol. 3936, pp.37-48.

[13] J. Davidson, B. Liebald, and Junning Liu. The YouTube Video Recommendation System. In RecSys '10: Proceedings of the fourth ACM Conference on Recommender Systems. 2010.