J. Service Science & Management, **2009, 2: 400-403**
doi:10.4236/jssm.2009.24048 Published Online December 2009 (www.SciRP.org/journal/jssm)

A Personalized Recommendation Algorithm Based on Associative Sets

Guorui JIANG, Hai QING, Tiyun HUANG

School of Economics and Management, Beijing University of Technology, Beijing, China.
Email: jianggr@bjut.edu.cn

Received August 11, 2009; revised September 13, 2009; accepted October 24, 2009.

ABSTRACT

During the process of personalized recommendation, some items evaluated by users are performed by accident, in other words, they have little correlation with users' real preferences. These irrelevant items are equal to noise data, and often interfere with the effectiveness of collaborative filtering. A personalized recommendation algorithm based on Associative Sets is proposed in this paper to solve this problem. It uses frequent item sets to filter out noise data, and makes recommendations according to users' real preferences, so as to enhance the accuracy of recommending results. Test results have proved the superiority of this algorithm.

Keywords: *Frequent Itemsets, Associative Sets, Collaborative Filtering, Recommendation Technology*

1. Introduction

How to help users quickly and effectively access to the information they really need when facing abundant resources becomes a challenging task and also a hot topic of current academic study. Personalized recommendation system is one of the effective tools to solve this problem. A helpful method is to develop intelligent recommendation system to provide personalized service [1], that is to recommend products to users according to their preferences or demands, so as to help them finish the purchasing process.

Nearest neighbor collaborative filtering approach is a recommendation technique that is the most widely used right now [2]. Its basic idea is to generate recommendations for target users according to the rating data of nearest neighbors that have given similar ratings. As items' (movies, music, etc.) ratings given by the nearest neighbors are quite similar to those given by target users, items' ratings given by target users can be estimated by the weighted average of the ratings given by the nearest neighbors. The advantage of collaborative filtering approach is that it can adapt to the rapid updating of users' information. It caculates the tightness among users according to the latest data every time, so as to make recommendations. However, the consequent disadvantage is that it is quite slow to get K nearest neighbors within large amounts of data. Meanwhile, results would not be satisfactory when sparse data is dealt with, especially for new products and new users. At the same time, its scalability is not very good [3].

On the basis of traditional collaborative filtering algorithm, our paper proposes a personalized recommendation algorithm based on Associative Sets. This algorithm first supposes user rating matrix as transaction sets, while every transaction is a user's rating set. Then it generates frequent itemsets through frequent itemsets generation algorithm, puts frequent itemsets into a series of Associative Sets according to one user's rating record, and performs collaborative filtering among Associative Sets so as to improve the accuracy and scalability of the algorithm.

2. Traditional Collaborative Filtering Algorithm and Its Analysis

2.1 Traditional Collaborative Filtering Algorithm

Collaborative filtering algorithm is the most widely used approach in personalized recommendations, which can forecast target users' interests and preferences according to neighbor users' interests and preferences. It first finds neighbors that have the same preferences with target users under the help of statistical techniques, and then makes recommendations to target users according to their neighbors' preferences. It includes three stages [4]:

1) Representation

Inputting data can usually be expressed as an $m \times n$ user rating matrix, where m represents the number of users, n represents the number of items, and R_{ij} represents the rating given by user i to item j. Such ratings can have several scales just as Table 1.

Table 1. User/item rating matrix

User	Item					
	I_1	I_2	···	I_j	···	I_n
U_1	R_{11}	R_{12}	···	R_{1j}	···	R_{1n}
U_2	R_{21}	R_{22}	···	R_{2j}	···	R_{2n}
···	···	···	···	···	···	···
U_i	R_{i1}	R_{i2}	···	R_{ij}	···	R_{in}
···	···	···	···	···	···	···
U_m	R_{m1}	R_{m2}	···	R_{mj}	···	R_{mn}

2) Neighbor Generation

For user *u*, generate a “nearest neighbor” set according to the level of similarity between neighbors. The calculation of similarity values between neighbors can be performed through vector space similarity calculation methods that are widely used currently, such as cosine method, pearson similarity method and so on. There are two ways to determine neighbors, one is to determine the similarity threshold through cosine method first, and then select users whose similarity values are greater than the similarity threshold as neighbor users; the other is to determine the number of neighbor users *N* first, and then select the first *N* users whose similarity values are greater as neighbor users.

3) Recommendation

As “nearest neighbor” set is generated, we can forecast one certain user’s rating for each item, and then make recommendations to that user according to the level of forecasting ratings.

2.2 Problem Analysis

Traditional collaborative filtering algorithm considers users’ entire historical information as its preference information and uses such information to find its nearest neighbors. However, users’ preferences are often formed exploringly and progressively in reality. It is a historical progress, and during this progress, users often try many times and become stable gradually, so as to form their real interests. Even though users’ interests have already been formed, they would sometimes try other items in daily search process for various reasons. Such items cannot be seen as their interests and the supporting evidences for recommendations. Therefore, if we want to gain real preference information of users, we must filter out the occasional search information to reduce interference. Find nearest neighbors according to users’ real preference information, and then make recommendations, while the results of recommendations can become better.

3. A Personalized Recommendation Algorithm Based on Associative Sets

As we have analyzed above, we can gain associative items through frequent itemsets. These associative items constitute the foundations of different interests. As for current users, we use their entire information to filter associative items, and then merge the associative items after filtering to form their interest sets.

3.1 Algorithm Descriptions

1) Set all items as $I = \{I_1, I_2, I_3, ..., I_n\}$, *n* as the number of items. See every user’s rating record as one item of transaction, $T_i \subseteq I$, which represents the rating set of user *i*, wherein $i \in \{1, 2, 3, ..., m\}$, *m* represents the number of users. So, user rating matrix can be seen as transaction set $T = \{T_1, T_2, T_3, ..., T_m\}$.

2) Use Apriori algorithm to generate the frequent itemsets F of transaction set T, whose support level is support.

- In the first iteration of the algorithm, each item of *I* is a member of the set of candidate 1-itemsets. The algorithm simply scans all of the transactions *T* in order to count the number of occurrences of each item.
- Select the candidate 1-itemsets, which satisfies minimum support *support*, to consist the set of frequent 1-itemsets L_1.
- Use $L_1 \times L_1$ to generate a candidate set of 2-itemsets, and prune using apriori property---All nonempty subsets of a frequent itemset must be frequent. Then, scan all of the transactions *T* in order to count the number of occurrences of each item in candidate set of 2-itemsets.
- Select the candidate 2-itemsets, which satisfies minimum support *support*, to consist the set of frequent 2-itemsets L_2.
- Constantly use $L_{k-1} \times L_{k-1}$ to generate a candidate set of k-itemsets, and prune it. Then, scan all of the transactions T in order to count the occurrence of each

item in candidate set of k-itemsets. Select the candidate k-itemsets, which satisfies minimum support *support*, to consist the set of frequent k-itemsets L_k.

- If the candidate set of k-itemsets is null, all frequent itemsets are gained.

3) Gain rating items I^* of current user a, and merger the frequent itemsets, which contains some items of I^* and also the number it contains is more than parameter num in F, as associative sets *C*.

4) Use Pearson correlation coefficient algorithm to calculate the similarity between user *a* and any other user *b* in associative sets *C*.

$$w^C(a,b)=\frac{\sum_{j\in C}(R_{aj}-\overline{R_a^C})(R_{bj}-\overline{R_b^C})}{\sqrt{[\sum_{j\in C}(R_{aj}-\overline{R_a^C})^2][\sum_{j\in C}(R_{bj}-\overline{R_b^C})^2]}} \tag{1}$$

where *j* is the item in associative sets *C*, R_{aj} is the rating given by user *a* to item *j*, R_{bj} is the rating given by user *b* to item *j*, $\overline{R_a^C}$ and $\overline{R_b^C}$ are average ratings of user *a* and user *b* separately.

5) For *a*, arrange all the users according to the value of $w^C(a,b)$, and select the first *M* users that have greater values as neighbor users $Neighbor_a$ of user *a*.

6) Forecast the rating of user *a* to item *j*. The forecasting formula is:

$$P_{a,j}=\overline{R_a}+\frac{\sum_{b\in Neighbor_a} w^C(a,b)(R_{b,j}-\overline{R_b})}{\sum_{b\in Neighbor_a} w^C(a,b)} \tag{2}$$

where $P_{a,j}$ is the forecasting rating of user *a* to item *j*, $R_{b,j}$ is the rating of user *b* to item *j*, $\overline{R_a}$ and $\overline{R_b}$ are average ratings of user *a* and user *b* to all items.

7) Arrange items according to the value of $P_{a,j}$, and select the first *N* items that have greater $P_{a,j}$ as recommendatory items.

3.2 Algorithm Explanations

1) In addition to the original rating records, there are four other parameters in this algorithm: support for the calculation of frequent itemsets *support*, threshold for the selection of associative sets from frequent itemsets *num*, number of nearest neighbors *M* and number of items that are recommended to users *N*. Wherein, *support* and *num* are used to determine Associative Sets, but what is the right combination needs to be tested. Usually, different data sets have different proper combination of *support* and *num*. Therefore, it will take more time to learn this algorithm.

2) Step 1 and step 2 in algorithm description are mainly used to generate frequent itemsets, which will take much more time. However, as it is performed offline, instant recommendations cannot be influenced.

3) As frequent itemsets have to be merged (Step 3) before collaborative filtering, it will take more time online than traditional algorithm will take, but its accuracy can be improved greatly. As the duration of merging frequent itemsets relies on the number of frequent items, to reduce frequent itemsets through offline activities can shortern online duration. In addition, because associative filtering items for every user are somewhat less than all the items, the duration of collaborative filtering process itself will be reduced. Through optimization, online duration of algorithm can be reduced accordingly.

4) Frequent itemsets include the complete set of frequent itemsets, the closed frequent itemsets, maximal frequent itemsets and so on [5]. The frequent itemsets used in this paper are maximal frequent itemsets, which can reduce the number of frequent itemsets greatly. If other frequent itemsets are used, we can calculate the importance of different items when calculating nearest neighbors with the help of support when merging frequent itemsets, which can improve the accuracy further more.

4. Test Process

4.1 Data Set and Evaluation Standard

Data set MovieLens is used to test this algorithm, which is provided by the GroupLens research lab at the University of Minnesota. The data was collected through the MovieLens web site (movielens.umn.edu) during a seven-month period. MovieLens includes 100000 records of ratings given by 943 users to 1682 movies. A rating is a number from 1 to 5, optionally supplemented by the number of seconds which the user spent reading the movie. Users are encouraged to assign ratings based on how much they liked the movie, with 5 highest and 1 lowest. Each user has given ratings to 20 moves at least. You can get the date set at www.grouplens.org.

Average Absolute Error (MAE) is used to evaluate the forecasting accuracy of this algorithm. MAE is the deviation average of the actual value and the predictive value of the ratings given by all users to the items. The lower the value of MAE is, the better the recommendations are. Supposing user rating set is $\{p_1,p_2,...,p_N\}$, and the actual user rating set is $\{q_1,q_2,...,q_N\}$, MAE is defined as follows [6]:

$$MAE=\frac{\sum_{i=1}^{N}|p_i-q_i|}{N} \tag{3}$$

4.2 Test Results and Remarks

We will compare Associative Sets Based Collaborative Filtering (ASBCF) and traditional User Based Collaborative Filtering (UBCF) during the test process. In order to verify the results, we will test in two dimensions.

One is to test in different sparse degrees. Here we select the first 200 users and first 500 items in MovieLens rating records, and then deduct some records every time randomly. In the end, we gain rating records under

200*500, 13270 pieces, 7976 pieces, 4525 pieces and 2162 pieces separately, and their sparse degrees are 0.13, 0.08, 0.043, 0.021662 separately, see Figure 1.

From the results, we can see that in different sparse degrees, MAE of ASBCF is lower than that of UBCF, 5.2206% in average. That is to say, ASBCF performs better in every sparse degree than UBCF. However, as the data is sparse extremely, ASBCF's accuracy will be reduced accordingly.

The other is to compare values of MAE with different numbers of nearest neighbors. Here we select 10, 20, 30, 40 and 50 nearest neighbors, and the test results can be seen in Figure 2.

From Figure 2, we can see that ASBCF also performs better than UBCF with all kinds of nearest neighbor numbers. However, along with the increasing of neighbor number, their gap becomes smaller and smaller. Maybe it is because ASBCF can find nearest neighbors more effectively than UBCF, and when the number of neighbors increases, users that are a little further from current user are also selected, which can increase error. That is to say, ASBCF is more effective than UBCF.

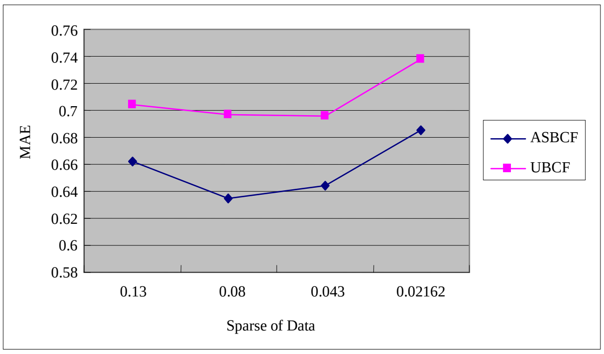

Figure 1. MAE of ASBCF and UBCF under different sparse degrees

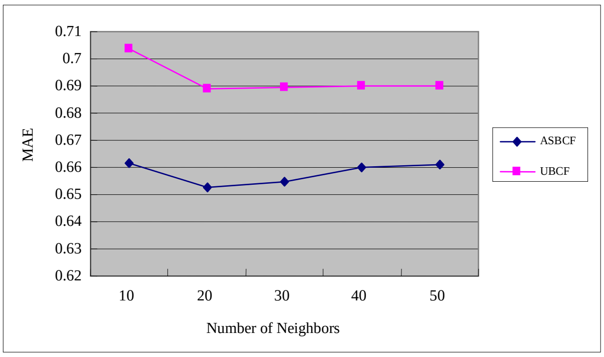

Figure 2. MAE of ASBCF and UBCF with different numbers of nearest neighbors

5. Conclusions

This paper proposes a personalized recommendation (collaborative filtering) algorithm based on Associative Sets. It generates a series of frequent itemsets through frequent itemsets generation algorithm, and then filters out some noise items that have little relevence with users by merging, so as to make collaborative filtering algorithm more effective. It is proved that this algorithm is better than traditional algorithm in recommendation accuracy. Although it takes more time to generate frequent items, it will not influence instant recommendations, as the generation can be performed offline. Support of frequent itemsets owns one kind of new information, which represents different items' importance. If such information is used in collaborative filtering, forecasting accuracy will be improved, and this is the breakthrough point for further research.

REFERENCES

[1] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-Commerce recommendation applications," Journal of Data Mining and Knowledge Discovery, pp. 115–153, 2001.

[2] J. Breese, D. Hecherman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98), pp. 43–52, 1998.

[3] L. Zhao, N. J. Hu, and S. Z. Zhang, "Algorithm design for personalization recommendation systems," Journal of Com- puter Research and Development, pp. 986–991, August 2002.

[4] Y. Li, L. Liu, and X. F. Li, "Research on personalized recommendation algorithm for user's multiple interests," Journal of Computer Integrated Manufacturing Systems, pp. 1610–1615, December 2004.

[5] J. W. Han and M. Kamber, "Data mining concepts and techniques (Second Edition)," Ming Fang, Xiaofeng Meng translated, China Machine Press, pp. 149–161, March 2007.

[6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item based collaborative filtering recommendation algorithms," In Proceedings of the Tenth International World Wide Web Conference, pp. 285–295, 2001.