A Kind of Collaborative Filtering Algorithm Based on User Clustering and Time Stamp

Shuqin Li[1, a] Xiaohua Yuan [2,b] Huaimei Han[3,c]

[1]Beijing Information Science& Technology University ,Beijing, P.R. China,

[2]Beijing Information Science& Technology University,Beijing, P.R. China

[3]Shanghai Oceanic University,Shanghai, P.R.China

[a] Lishuqin_de@126.com , [b.]xhyuan@shou.edu.cn,[c] 763219676@qq.com

Keywords: Collaborative filtering; user clustering; time stamp; user recommendation

Abstract: In the case of user rating data is very large and sparse, the effect of traditional collaborative filtering algorithm in use recommendation is unsatisfactory. In this paper, a collaborative filtering algorithm based on user clustering and time stamp is proposed. Which first clusters the users according to user's preferences for different types of items, thus user will only recommended items to members belong to his/her own cluster, thus can largely reduce the input data of recommendation algorithm, and can improve recommendation effectiveness. Then taking account the factor that user will change his/her interest, in the traditional collaborative filtering algorithm, time factor is added to realize a real-time user recommendation. Experimental results show that the proposed algorithm can improve the accuracy and recall rate of user recommendation.

1 INTRODUCTION

Item-based Collaborative Filtering Algorithms(ItemCF）[1] is widely used in recommendation system, the basic idea of it is to calculate the similarity between items through analyzing the records of user behaviors, and recommend to user some similar items they liked before. In practical application, in the recommendation system there usually exist problems such as data sparse, cold start, and lack of extensibility [2]. With the more deeply application of recommended system, many researchers have put forward their new methods to improve the shortcomings of the recommendation system, among which: from item properties and the link of rating, paper[3-4] proposes item comprehensive similarity calculation method, to predict the element of user-item rating matrix; Paper [5] obtains user properties through user classification, and uses association rules algorithm to analyze the history record of user, thus to predict user's future possible behaviors; And to solve the large data expansion problem, paper [6] proposes a MapReduce based matrix decomposition recommendation algorithm.

Aiming at movie recommendation, taking advantage of the distributed computing framework of MapReduce, this paper proposes a real-time collaborative filtering algorithm based on users clustering, which try to reduce the amount of input data in the recommendation, thus to improve the response rate and the accuracy of the recommendation.

2 REAL-TIME RECOMMENDATION ALGORITHM BASED ON USER CLUSTERING

Taking account that in the existing recommendation system, there exists the problems of data sparsity and lack of extensibility, this paper adopt a mutual recommendation, in which user only recommend item to use who has the same interest with her/his, this can: largely reduce the dimension and sparsity of input data, relieve the problem of extensibility, lessen the time of calculation, and as a result, efficiency of the recommendation system can be improved largely.

The proposed recommendation firstly cluste users according to their preference degree of each item, then extract user's original rating scores by taking one cluster as a unit, and then recommend item to user by ItemCF. In order to ensure the effect continuity of recommended list on the prediction of user interest, and take account in the change of user interest, we add time factor in the proposed recommendation algorithm, thus to balance the short-term behavior and long-term behavior,

as a result, our user cluster based recommendation become a real-time one. The main processes of the recommendation are shown in figure 1, which include data preprocessing, preference calculation user clustering, and collaborative filtering.

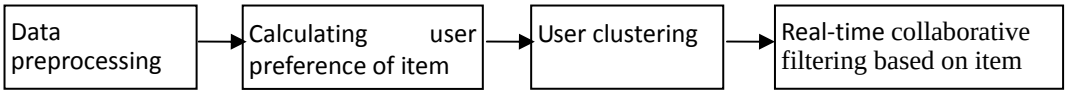

Fig.1 Flow graph of user clustering

3 IMPLEMENTATION OF REAL-TIME RECOMMENTATION ALGORITHM BASED ON USER CLUSTERING

3.1 DATA PREPROCESSING

In the experiment, one data set provided by MovieLens (http://www.grouplens.org/node/73) is used, the size of the data set is 1mb, it contains: 1000209 rating records of 3883 movies rated by 6040 users, the attribute information of the movies, and the personal statistics information about the rating users.

For the succeeding study and data usage, we pre-process the original data set as flowing: from the data set, some fields respectively from the user table and the movie table, are selected to construct one new table called as user-item type table, which includes fields such as user id, movie id, rating, and movie type, as shown in table 2. In the user-item type table, under the field of movie type, there stored the id of movie type, namely, number between 1 and 19, in case a movie belongs to more than one types, the corresponding ids are separated by a segmentation character. For example, for a movie belongs to types of action and adventure as well, in the pre-processed table, its type value is assigned as 2|3.

User table

Field Name	Identifier
User ID	user id
Movie ID	movie id
Rating value	rating
Timestamp	timestamp

Table Merging

Movie table

Field Name	Identifier
Movie ID	Movie id
Movie Title	Movie title
Release Date	Release date
Video Release Date	Video release date
Link in IMDB	IMDB URL
Type of Movie Unknown	unknown
…	Other type
Type of Movie Western	western

User-Item table

Filed Name	Identifier
User ID	user id
Movie ID	movie id
Rating value	rating
Type of Movie	movie type

Fig. 2 Result of data pre-processing

3.2 USER PREFERENCE OF ITEM

User always prefer some type of movies, for example, some one like comedies, and some others like action movies. Based on the algorithm of Term Frequency and Inverse Documentation Frequency(TFIDF) which was used to measure words feature in the content based recommendation, we process the data of user-item table, thus to obtain user's preference of movie type. The output of the data processing take the format as {user id, movie type1:pref1, movie type2: pref2,…, movie type19: pref19}, in which, the preference degree of user u for movie type i is defined as

$$Adherence(u,i) = TF_{ui} * IDF(i) \qquad (1)$$

where $$TF_{ui} = \frac{|T(u,i)|}{\sum_{j=1}^{S} T(u,j)} \qquad (2)$$

$$IDF(i) = \log \frac{\sum_{i=1}^{S}\sum_{k=1}^{N} T(i,k)}{\sum_{k=1}^{N} T(i,k)} \qquad (3)$$

and $|T(u,i)|$ is the number of movie type i that has been watched by user u, $\sum_{j=1}^{S} T(u,j)$ represents the total number of movies watched by user u, $\sum_{i=1}^{S}\sum_{k=1}^{N} T(i,k)$ represents the total number of movies in the movie set, and $\sum_{k=1}^{N} T(i,k)$ is the total number of movies belong to type i.

Function (3) represents the distribution of movie i in the whole movie set, thus introduce it in to the system can efficiently prohibit the excessive effort of some movie type of which the movies quantity is too large or too small. In all the movies that has rated by user u, the number of movie type i is larger, it indicates that user u is more like movie type i.

3.3 USER CLUSTERING

In user clustering, we take the method that combined Canopy clustering algorithm and K-means. The typical characteristic of Canopy clustering algorithm is that it needs not to specify the value of K in advance, and compared with other clustering algorithms, although its precision is low but it is very fast.

In this paper, we use Canopy clustering algorithm to finish a "coarse clustering", which quickly determine the K value for the K-means classification, and at the same time, locate the original center for the clusters. Then, we use K-means clustering algorithm to fulfill a "precise clustering ". In the implementation, we use the combined algorithm of Canopy clustering algorithm and K-means that integrated on Aphache Mahout0.8, in the calling of the algorithm, there needs to set two parameters, T_1 and T_2. In this paper, making use of the quantities calculated in section 3.2, we take movie type and user preference data as the input data, in which, each record in the user-item table is taken as one point vector, and using formula (4) and (5), we calculate the values of T_1 and T_2, respectively.

$$T_2 = \frac{\sum_{n}^{j=1}\sum_{n}^{i=1} \frac{x_i \cdot x_j}{x_i^2 + x_j^2 - x_i \cdot x_j}}{n} \qquad (4)$$

$$T_1 = 2 * T_2 \qquad (5)$$

where x_i and x_j represent the point vectors wait to be clustered, and n is the number of the point vectors.

After run the Canopy clustering algorithm and K-means algorithm, we can obtain one result file which includes all the cluster id, and user id that corresponding to each cluster id, and this result is just the distribution of users in the clusters.

3.4 REAL-TIME COLLABORATIVE FILTERING ALGORITHM BASED ON ITEM

3.4.1 TIME FACTOR

Recommendation system in real-time not only requires for real-time processing of user behavior records, but request the recommendation algorithm itself be real-time also. In this paper, in the itemCF algorithm, we adopt a time attenuation function to reduce the effect of user's long before preference to some recommendation. The attenuation function $f(|t_{ui}-t_{uj}|)$ we used is as follows

$$f(|t_{ui}-t_{uj}|)=\frac{1}{1+\partial|t_{ui}-t_{uj}|} \qquad (6)$$

where t_{ui} is time at which user action on movie i, and ∂ is the time attenuation parameter. The intention of function $f(|t_{ui}-t_{uj}|)$ is that, the time interval between user's actions on movie i and on movie j is longer, then the value of $f(|t_{ui}-t_{uj}|)$ is less.

In the experiments of this paper, time is represented by timestamp, thus t_{ui} is equal to the value of the related timestamp after divided by $24*60*60$. To select a suit value of ∂, there need to perform many a few tests, in this paper, we set the value of ∂ as 0.5.

3.4.2 REAL-TIME ItemCF algorithm

The main steps of the real-time recommendation algorithm based on user clustering are as follows:

Step 1, according to the clustering results, from the original rating data, extract the rating data of users belong to each cluster.

Step 2, calculate the similarity between each pair of items by formula (7), store the calculated results in the HDFS file on Hadoop platform, the data fields include {movie id, movie id, similarity degree}.

$$sim(i,j)=\frac{\sum_{u\in N(i)\cap N(j)} f(|t_{ui}-t_{uj}|)}{\sqrt{|N(i)||N(j)|}} \qquad (7)$$

where: $N(i)$ represents users that have scored movie i, $|N(i)|$ is the number of these users, and $f(|t_{ui}-t_{uj}|)$ is the attenuation function that an be calculated by (6).

Step 3, according to the calculated similarity degree of movie, for each user, calculate his predictive rating for the movie that he has not scored yet. The prediction score formula is as follows.

$$p(u,i)=\sum_{j\in M(u)} sim(i,j)\frac{r(u,j)}{1+\beta|t_0-t_{uj}|} \qquad (8)$$

Where $r(u,j)$ is the rating value of user u for movie j, $M(u)$ is the set of movies that user u has rated, and t_0 represents the current time.

Step 4, for each user, sort the movies list according to his predicted score related, and recommend the top n movies to the user.

4 EXPERIMENTAL RESULTS AND ANALYSIS

In the experiments, we sort the rating data according to timestamp in an ascending order, and divide

the data into 10 sets. For each data set, we take the first 80% data as the train set, the last 20% as the test set, and repeat the experiment 10 times.

On the foundation of section 3.4, this section valid the real-time recommendation based on user clustering, by comparing the results with that of the distributive and collaborative filtering algorithm on Mahout0.8. In the experiments, we

1) Successively pre-process the training data set, and construct the user preference file

2) Calculate the value of T_1 and T_2 by formula (4) and (5), by taking the preference file outputted by 1) as the input file.

3) Cluster user by the algorithm described in section 3.3, taking the preference file as input file again.

Figure 3 display the result of one clustering, in which T_1 and T_2 are assigned as 1.06 and 0.53, respectively.

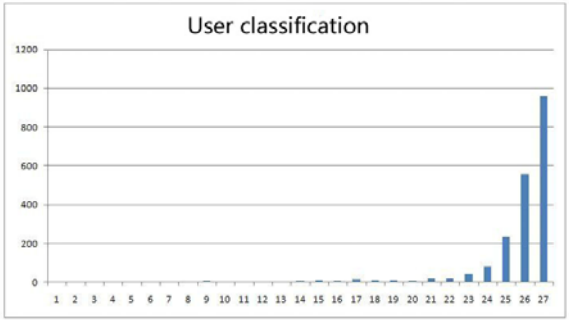

Fig.3 Results of one test of user clustering

In Figure 3, the 3253 users are clustered into 27 clusters, after sorting the clusters according the number of belonged users, we can find that: the user distribution in figure 3 embodies a long-tail form, that is, in most clusters, there are little users, but most of the users belong to a small number of clusters, this indicates that most of the users have similar preference. And in the results of other clustering, there exist a similar tendency.

4) From each of the clustering results, we select the user id of 10 clusters, taking each of the 10 cluster as the unit, we extract the rating data belong to those user ids, and take the extracted data as input data of the real-time recommendation algorithm, that is, all the rating data of the users that belong to the 10 clusters were used as the input data of the distributive ITCF algorithm on Mahout0.8. The results of average precision and recall ratio of each experiment are shown in Figure 4 and Figure 5, from which we can see that, in the 10 experiments, the recommended strategy proposed in this paper can increase the recommendation precision and recall ratio, by 1.07% and 1.56%, respectively.

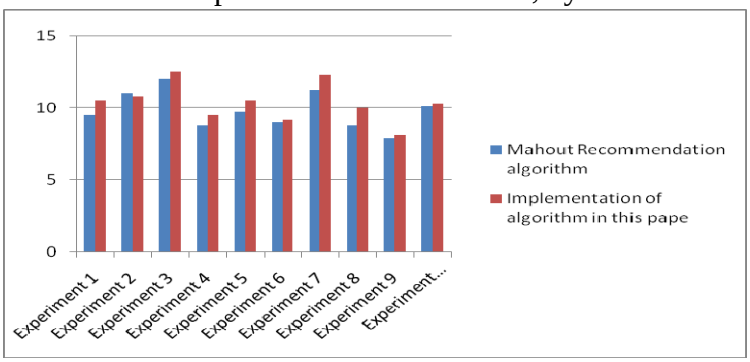

Fig. 4 Comparison of Recommendation accuracy

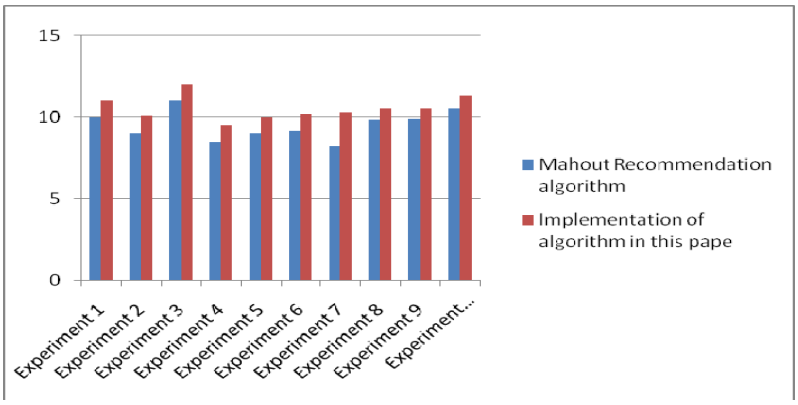

Fig. 5 comparison of Recommended recall ratio

5 Conclusions

Taking into account that for users with the same interests, if we firstly pre-process the rating records and make the recommendation based on the pre-processed result, the recommendation based on ItemCF will be more effective, thus in this paper, we propose an improved ItemCF based on user clustering and time stamp, in which we firstly cluster the users according to their preference to different movie types, then taking cluster as the unit, we extract the related rating records from the original score table, then take the extracted result as well as the timestamp information as the input of the collaborative filtering algorithm, thus to effectively reduce the input data of recommended algorithm, and improve the response speed and the recommendation accuracy. In the experiments, there needs to assign some parameters, the values given in this paper are for reference only. The experimental results show that, compared with the recommendation algorithm provided by mahout0.8, our recommendation can increase the precision and recall ratio at some extent.

Acknowledgements

This paper is jointly sponsored by the Network culture and digital dissemination (Beijing Key Laboratory of research funding ICDD201507), and by supported by National Natural Science Foundation of China (61502039) ,and by Sensing & Computation Intelligence Joint Laboratory.

Reference

[1] G. Linden, B. Smith, and J. York, Amazon.com recommendations: item-to-item collaborative Filtering[C]. IEEE Internet Computing, 2003, vol. 7, no. 1, 76-80

[2] Shao hua Sun. Study on the sparsity and cold start of collaborative filter system [D]. Zhejiang University, 2005, Hangzhou, Zhejiang Province, China, 2005.

[3] Vincent S-Z, Boi Faltings. Using hierarchical clustering for learning the ontologies used in recommendation systems[C]. Proceedings of the 1301 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Jose, California, United States, 2007:599-608

[4] Shi Peng. Study on Collaborative filtering algorithm based on user preference and item feature. Thesis of Central South University, Changsha, Hunan Province, China, 2012.

[5] Horng Jinh Chang, Lun Ping Hung, Chia Ling Ho. An anticipation model of potential customers' purchasing behavior based on clustering analysis and association rules analysis[C]. Expert Systems with Applications. 2007(32):753–764

[6] ZHANG Yu, CHENG Jiujun. Study on Recommendation algorithm with matrix factorization method Based on mapReduce. [J]. Computer Science,2013,40(1)：19-22

