ICIC Express Letters
Volume **10**, Number **10**, October **2016**

ICIC International ©2016 ISSN 1881-803X
pp. 2493–2498

RESEARCH ON COLLABORATIVE FILTERING RECOMMENDATION BASED ON K-MEANS CLUSTERING

Liyan Dong[1,2], Gaozi Zhu[1], Qi Zhu[1] and Yongli Li[3]

[1]College of Computer Science and Technology
[2]Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education
Jilin University
No. 2699, Qianjin Street, Changchun 130012, P. R. China
dongly@jlu.edu.cn

[3]School of Computer Science and Information Technology
Northeast Normal University
No. 2555, Jingyue Ave., Changchun 130117, P. R. China
liyl603@nenu.edu.cn

Received March 2016; accepted June 2016

Abstract. *In accordance with the inaccuracy of searching neighbors in traditional collaborative filtering algorithms, we narrow down the space of neighbor searching by means of partition clustering to improve the real-time performance of recommendations. Then by taking the similarity measurement into consideration, we convert the high-dimensional sparse matrix of user rating to the low-dimensional dense matrix of user interest to enhance the accuracy of recommendations. Experiments proved that combining the partition clustering based on user interests and collaborative filters could contribute to increasing real-time speed and the accuracy of recommendation effectively.*

Keywords: Recommendation, Collaborative filtering, Partition clustering, Neighbor search

1. **Introduction.** With the vigorous development of the Internet and communication technology, people have passed the era that lacks information to the era that is overloaded [1]. At present, the major methods of combating information explosion are information retrieval and information filtering. Search engine, as a representative of information retrieval technology, has gradually matured. Personalized recommendations, served as a great supplement of search engines, are the most visible information filtering application which has been widely used in various kinds of fields, such as Electronic Commerce, Social Network, Location Based Service (referred to as LBS), and Personalized Advertisements. The common methods and approaches applied to personalized recommendations are collaborative filtering, content-based recommendations and graph-based recommendations. Particularly, collaborative filtering [2] is the most popular algorithm in this area. The motivation for collaborative filtering comes from the idea that people often get the best recommendations from someone with similar tastes to themselves. Collaborative filtering explores techniques for matching people with similar interests and then makes recommendations. Unfortunately, when the overwhelming amount of data occurs, collaborative filtering would be troubled with some problems like sparse data, cold start, inaccurate recommendation and the expansibility of algorithms.

In this paper, on the basis of the recommendation system, we apply partition clustering into the enhancement of neighbor search in collaborative filtering and propose a novel collaborative filtering algorithm based on k-means clustering named as UIC-CF, short for User-Interest Clustering Collaborate Filtering. Experiments show that the improved algorithm has many advantages in the real-time performance and the accuracy of recommendation.

2. **Related Work.**

2.1. **User-based collaborative filtering.** Personalized recommendation systems have many forms. A specific series of the algorithm is User-Based Collaborate Filtering [3], referred to as UCF. It can be described as follows.

Step 1. Data preprocessing

The scores of those items rated by the users are represented as user-item-rating matrix. For instance, Table 1 lists the statistics of a simple user-item-rating matrix, and especially the score is zero if the user had not scored on the item.

Table 1. User-item-rating matrix

user/item	i_1	i_2	...	i_n
u_1	1	0	...	3
u_2	0	4	...	0
...	...	...	...	...
u_m	1	0	...	0

Step 2. Searching neighbors set

The computation of user neighbors is based on the user similarity. Given the user-item-rating matrix, a ranking of users in decreasing order of user similarity is returned. The user neighbors set is made up of the top K users of this ranked list. To be specific, it is assumed that each user can be represented as an n dimensional score vector. What is more, traditional similarity measurement [4] among users contains Cosine Similarity, Adjusted Cosine Similarity and Pearson Correlation Coefficient. In this paper, cosine similarity is introduced for its simplicity, which defines the similarity between two users u and v as

$$\mathrm{Sim}\left(\vec{u},\vec{v}\right)=\cos\left(\vec{u},\vec{v}\right)=\frac{\vec{u}\times\vec{v}}{|\vec{u}|\times|\vec{v}|} \tag{1}$$

Step 3. Predicting and recommending the top N items of a ranked list

The user similarity is used to combine and weight the preferences of user neighbors. Thus, the prediction for the target user can be calculated as

$$P_{ui}=\bar{R}_u+\frac{\sum_{v=1}^{K}\mathrm{Sim}(u,v)*(R_{vi}-\bar{R}_u)}{\sum_{v=1}^{K}\mathrm{Sim}(u,v)} \tag{2}$$

where P_{ui} represents the predictive score of item i rated by user u, R_{vi} means the real score of item i rated by user v, $\bar{R}_u$ is the average score of all items rated by user u [5], K is the size of user neighbors set, and $\mathrm{Sim}(u,v)$ is served as the similarity between users u and v. Finally, UCF returns a ranking of ungraded items in decreasing order of P_{ui} in order to recommend the top N items to the target user u.

2.2. **K-means clustering.** Clustering is the process of splitting data into clusters. After clustering, objects in the same cluster share high similarity with each other while objects in different clusters are extremely different from others. K-means clustering is a classical clustering algorithm based on partitioning with the advantages of high efficiency and fast convergence [6].

2.3. **User interest.** Each item in recommendation system has its own attribute characteristics. Our focus is to obtain useful information such as the item attributes to improve the accuracy of recommendations. Referring to the important method of evaluating words in the collection of documents, which is named as TF-IDF, short for term frequency-inverse document frequency [7], we put forward the definition of user interest in the item attributes. The interest of user u in the item properties p is defined as

$$\mathrm{Interest}(u,p)=TF(u,p)*IDF(p) \tag{3}$$

where

$$TF(u,p) = \frac{N_{up}}{\sum_{i=1}^{N} N_{ui}} \tag{4}$$

$$IDF(p) = \log \frac{\sum_{u=1}^{m} \sum_{i=1}^{N} N_{ui}}{\sum_{u=1}^{m} N_{up}} \tag{5}$$

Here, N is the total number of item attributes, m represents the amount of users in the system, and n is the total quantity of items. N_{up} is the amount of items which contain attribute p and interest user u as well. $\sum_{i=1}^{N} N_{ui}$ is the total times of each attribute occurring in the items liked by user u. n_p defines the number of items containing attribute p in the system. IDF is a measure of how much information the attribute provides, that is, whether the attribute is common or rare across all items. It is obvious that it can reduce the weights of popular attributes and improve the discrimination of the specific attributes.

3. **UIC-CF.** The measurement of the neighbors for the target user in the traditional collaborative filtering algorithm needs to traverse all users in the system. When faced with a large amount of data, the method of iterating through all the users has become impractical. Thus, the technology of collaborative filtering has encountered unprecedented challenges on the real-time performance and scalability of recommendation system. If we search the neighbors set by means of clustering, the scope of search could be narrowed down to the cluster, whose users share the similarity with the target user, in order to improve the efficiency of neighbor search. As a result, the clustering method applied to collaborative filtering became an effective solution to deal with the high dimensional issues. On the basis of user-item-rating information, in this section, we fully excavate the potentiality of user interest [8], and propose the novel collaborative filtering algorithm (UIC-CF). Finally, a detailed discussion focused on the real-time performance and recommendation accuracy of the improved algorithm was conducted.

3.1. **UIC-CF overview.** UIC-CF converts the high-dimensional sparse matrix of user rating to the low-dimensional dense matrix of user interest in order to achieve much higher accuracy of measurement on user similarity. It contributes to dominating the performance bottleneck of collaborative filtering algorithm in recommendation accuracy. Meanwhile, UIC-CF searches the neighbors set using k-means clustering, so the scope of search could be narrowed to the cluster or nearby clusters, so as to improve the efficiency of neighbor search. Consequently, it deals with the problem of poor real-time performance of recommendation. The main process of UIC-CF can be described as: 1) Calculate the user-interest matrix R_{mN}; 2) Clustering for R_{mN}; 3) Searching neighbors set on the basis of cluster; 4) Top N recommendation based on the neighbors set of the target user.

To clarify the description, we translate the user-item-rating matrix into the user-interest matrix R_{mN}. Each user is described as $\vec{u}$ with his interest, and thus the user set is represented as U. C represents the set of clusters, and CC means the set of clusters' centers. Then, the clustering for R_{mN} is described as Algorithm 1.

After clustering, users shared with high similarity were split into the same cluster. Then the top K search for neighbors set is shown as Algorithm 2.

Finally, UIC-CF predicts the score of ungraded item and then recommends the top N items of a ranked list to the target user.

3.2. **Evaluation metrics.**

Real-time performance: [9] introduced the space-searching rate, named as Ratio, to specify the real-time improvement. It is defined as

$$\text{Ratio} = \frac{NU_1}{NU_2} \tag{6}$$

$$NU_1 = C_1 \cup C_2 \cup \ldots \cup C_k \tag{7}$$

where NU_1 represents the neighbors set of the improved collaborative filtering algorithm while NU_2 represents the neighbors set of the traditional collaborative filtering algorithm searching space. If the space-searching rate is less than 1, then the real-time performance has been improved. The less the ratio is, the better the result of the improvement is.

Algorithm 1 CreateCluster (U, R_{mN}, k)

```
PROCESS-INIT_INPUT   // $CC = CC_1, CC_2, \ldots, CC_k \leftarrow k$ random user in $U$
                     // $C = C_1, C_2, \ldots, C_k \leftarrow \{\}, \{\}, \ldots, \{\}$
Repeat:
foreach user $\vec{u} \in U$
    foreach Cluster Center $cc \in CC$
        calculate Sim$(\vec{u}, cc)$
    end foreach
    Sim$(\vec{u}, CC_m)$ = Max(Sim$(\vec{u}, CC_1)$, Sim$(\vec{u}, CC_2)$, ..., Sim$(\vec{u}, CC_k)$)
      $C_m = C_m \cup \vec{u}$
end foreach
foreach Cluster Center $cc \in CC$
    update $cc$
end foreach
Until $CC_1, CC_2, \ldots, CC_k$ no change
return $C$, $CC$
```

Algorithm 2 TopKSearch $(\vec{u}, K, C, CC)$

```
PROCESS-INIT_INPUT   // KNN = $\emptyset$, UserCluster = $\emptyset$, KUserSet = $\emptyset$
foreach Cluster $cc \in CC$
    calculate Sim$(\vec{u}, cc)$
end foreach
UserCluster $\leftarrow$ Sort$(\vec{u}, CC)$
foreach Cluster $c \in$ UserCluster
    K_Temp = K_Temp + Count$(c)$
    KUserSet = KUserSet$\cup c$
    if (K_Temp $> K$)
        break
end foreach
foreach user $\vec{u}_i \in$ KUserSet
    calculate Sim$(\vec{u}, \vec{u}_i)$
end foreach
KNN $\leftarrow$ Sort(Sim$_u$, $K$)   // user similarity in decrease order
return KNN
```

Accuracy degree: The MAE, abbreviation for mean absolute error [10], means the absolute deviation between the projected ratings recommended by the system and the actual ratings by users. The less the MAE is, the less the error differences are, and the more accurate the predictions are.

4. **Experiments.**

4.1. **Experiment data sets.**

MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota, which are one of the most popular data sets in the area of recommendation. In this paper, we use the ml-100k version. Fifty percent cross validation method is used across the experiments. Therefore, the average value of the results from five time experiments was used as the final result of the experiment.

4.2. Experimental results.

Real-time performance

Figure 1 shows that the improved algorithm (UIC-CF) could find the target user's neighbors set in a small percentage of the user space. At the same time, the process of clustering can be done offline, and therefore it helps to improve the real-time performance of recommendation.

Accuracy of recommendation

Cosine similarity is introduced for its simplicity in this paper. Figure 2 compares the accuracy performance between UIC-CF and UCF. Here, abscissa represents the size of neighbors set, and ordinate shows the MAE of recommendation.

As shown in Figure 2, with different neighbor sizes, the MAEs of UIC-CF are all less than UCF's. Consequently, UIC-CF could contribute to enhancing the performance accuracy of recommendation effectively.

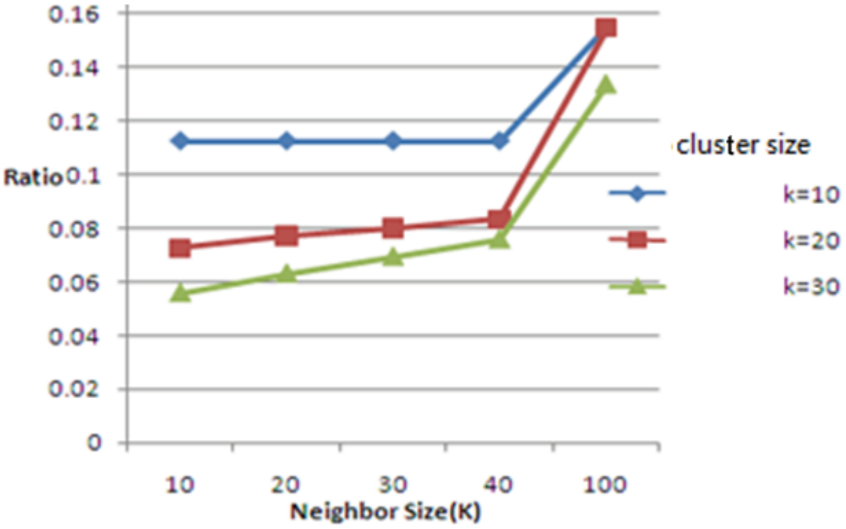

Figure 1. The ratio of UIC-CF

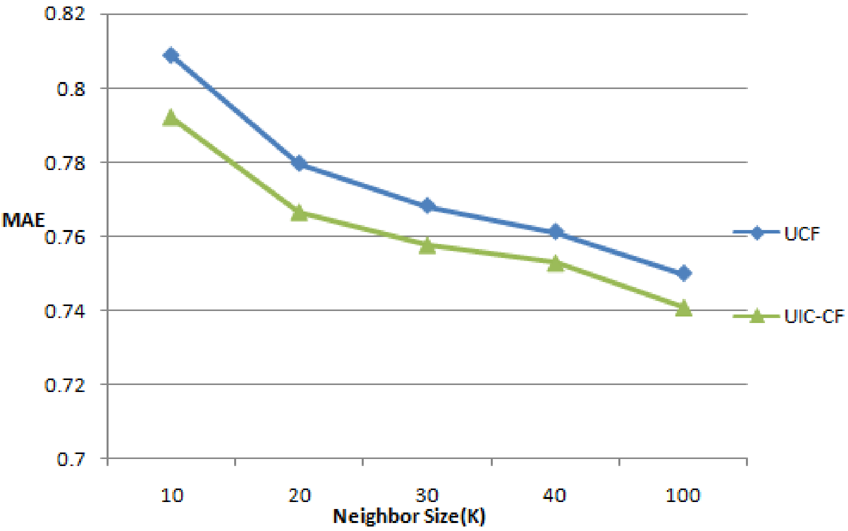

Figure 2. The competition between UCF and UIC-CF on accuracy

5. **Conclusions.** UIC-CF searches the neighbors set using k-means clustering in order to improve the efficiency of neighbor search, so as to enhance the real-time performance of recommendation. It also converts the high-dimensional sparse matrix of user rating to the low-dimensional dense matrix of user interest for reaching much higher accuracy of determination of user's neighbors set, and improves the accuracy of recommendation consequently. However, the clustering effect of the classic k-means clustering on the sparse matrix was not ideal, and we could try to use other matrix clustering algorithms to improve the collaborative filtering, such as the clustering algorithm based on density. So applying the idea of clustering to the item-based collaborative filtering is also a research direction.

Acknowledgment. This research is supported by the National Natural Science Foundation Project (61272209).

REFERENCES

[1] M. Nilashi, D. Jannach, O. Ibrahim and N. Ithnin, Clustering- and regression-based multi-criteria collaborative filtering with incremental updates, *Information Sciences*, vol.293, pp.235-250, 2015.

[2] B. M. D. Ekstrand, J. T. Riedl and J. A. Konstan, Collaborative filtering recommender systems, *The Adaptive Web, Methods and Strategies of Web Personalization*, vol.9, pp.45-46, 2015.

[3] M. Elahi, M. Braunhofer, F. Ricci and M. Tkalcic, Personality-based active learning for collaborative filtering recommender systems, *AI*IA 2013: Advances in Artificial Intelligence*, pp.360-371, 2013.

[4] M. R. Steuerer, Implementing K-means clustering and collaborative filtering to enhance sustainability of project repositories, *Proc. of the 47th ACM Technical Symposium on Computing Science Education*, p.724, 2016.

[5] J. Zhang, Y. Lin, M. Lin and J. Liu, An effective collaborative filtering algorithm based on user preference clustering, *Applied Intelligence*, pp.1-11, 2016.

[6] J. Aligon, E. Gallinucci, M. Golfarelli, P. Marcel and S. Rizzi, A collaborative filtering approach for recommending OLAP sessions, *Decision Support Systems*, vol.69, pp.20-30, 2015.

[7] Y. Park, S. Park, W. Jung and S. G. Lee, Reversed CF: A fast collaborative filtering algorithm using a k-nearest neighbor graph, *Expert Systems with Applications*, vol.42, pp.4022-4028, 2015.

[8] G. M. Dakhel and M. Mahdavi, A new collaborative filtering algorithm using K-means clustering and neighbors' voting, *The 11th International Conference on Hybrid Intelligent Systems*, pp.179-184, 2011.

[9] N. Polatidis and C. K. Georgiadis, A multi-level collaborative filtering method that improves recommendations, *Expert Systems with Applications*, vol.48, pp.100-110, 2016.

[10] U. Kużelewska and K. Wichowski, A modified clustering algorithm DBSCAN used in a collaborative filtering recommender system for music recommendation, *Theory and Engineering of Complex Systems and Dependability*, pp.245-254, 2015.