International Journal of Hybrid Information Technology
Vol.9, No.4 (2016), pp. 437-446
http://dx.doi.org/10.14257/ijhit.2016.9.4.38

A Slope One and Clustering based Collaborative Filtering Algorithm

An Gong[1], Yun Gao, Zhen Gao, Wenjuan Gong, Huayu Li, Hongfu Gao

School of Computer & Communication Engineering, China University of Petroleum, Qingdao 266580, China

Abstract

Collaborative filtering is the most successful and widely used technology in E-commerce recommendation system. However, the traditional collaborative filtering recommendation algorithm faces severe problems of sparse user ratings and poor scalability. Slope One algorithm can reduce the sparsity of ratings, improve the recommendation accuracy, but with the growth of users and items, the running time increases rapidly. In this paper, we first introduce the feature similarity into Slope One algorithm, then combine it with ants clustering algorithm, thus reliving the influence of rating sparsity, improving the searching speed, and reducing the searching costs. Experimental results show that the new algorithm can efficiently improve recommendation quality.

Key words *Slope One; clustering; recommendation system; collaborative filtering; score prediction*

1. Introduction

With the dramatic increase of Internet and E-commerce, the "information overload" problem is getting worse and worse. It's difficult for users to find what they really need from the mass of information. Recommendation systems can suggest users items that may fit their interests like sales, and therefore it has been widely used in E-commence [1]. Collaborative filtering (CF) is the most successful and promising recommendation technology which maintains a database of preferences for items by users [2]. However, there are several limitations associated with the traditional collaborative filtering algorithm: the user-item rating matrix is typically very sparse since most users do not rate most items which causes inaccuracy in searching neighbors, thus reducing the real-time and accuracy, and further affecting the quality and efficiency of recommendation, that is to say, the accuracy and scalability of the collaborative filtering algorithm need to be improved urgently.

Many scholars have proposed a variety of methods into recommendation systems to improve performance. Melville et al. [3] use content-based methods to fill the empty values and ease data sparsity, thereby improving the quality of recommendation. A Combined recommendation method is proposed in [4] to reduce the impact of sparsity. Luo et al. [5] introduce the concept of local user similarity and global user similarity, and propose a collaborative filtering framework based on both of these user similarity measures. Lee et al. [6] illustrate a cooperative prediction design between user-based CF and item-based CF using a similar prediction procedure but different styles of information. The authors in [7] present a collaborative filtering algorithm based on item and user for Multiple-interests and Multiple-content which provides better recommendation quality than collaborative filtering based on user and collaborative filtering based on item dramatically. For scalability, Gong [8] proposes a personalized recommendation approach joins the user clustering technology and item clustering technology. A method utilizes SVD and demographic data at various points of the filtering

[1] Corresponding author.

Email address: gongan0328@sohu.com (An Gong) procedure is proposed in [9] in order to improve the quality of the generated predictions. Lemire et al. [10] introduce Slope One schemes which are easy to implement, dynamically updateable, efficient at query time. Mi et al. [11] propose a new approach in improving the sparsity and scalability of recommendation systems by using the clustering algorithms based on the slope one scheme.

In this paper, we introduce item attributes similarity into Slope One to reduce the matrix sparsity, and then cluster users using ant colony algorithm [12] to lessen the searching space of neighbors and thus enhancing the scalability. Moreover, we validate the effectiveness of Slope One-based algorithm, Clustering-based algorithm and our method respectively. The experimental results show that the method performs well.

2. Related Work

User-based collaborative filtering algorithm is based on the hypothesis that if two users' score scheme on some items is similar, then their ratings for others will probably be about the same. The basic idea is firstly getting the similarity between users through their previous ratings, secondly choose the Top K users as neighbors of object user, and then use the neighbors' ratings to foresee the rating of the user on unrated items, at last, show the N top-scoring items to the user. In a typical CF scenario, the data structure can be defined as $D = \{U, I, R\}$, where U is a set of m users $\{u_1, u_2, \dots, u_m\}$, *I* is a set of *n* items $\{i_1, i_2, \dots, i_n\}$ and *R* is user-item matrix with values R_{ij}being the rating of user *i* to item *j*. Note that there are a lot of null values in *R* since most of the users do not rate most of the items.

2.1. Similarity Measuring Methods

There are mainly three ways to compute the similarity between items:

(1) Cosine-based similarity: In this case, two users are thought of as two vectors in the *n* dimensional item-space. Vector ***u*** denotes the ratings of user *u* and vector ***v*** denotes the ratings of user *v.* And similarity between user *u* and user *v*, denoted by sim(*u*, *v*)is given by:

$$\mathrm{sim}(u, v) = \cos(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u} \bullet \boldsymbol{v}}{\|\boldsymbol{u}\| \bullet \|\boldsymbol{v}\|} \tag{1}$$

Correlation-based similarity: In this case, we only consider items rated by both *u* and *v,* and we let alike items compose the set $I_{u,v}$, then the correlation similarity is given by:

$$\mathrm{sim}(u, v) = \frac{\sum_{i \in I_{u,v}} (R_{u,i} - \bar{R}_u)(R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_{u,v}} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (R_{v,i} - \bar{R}_v)^2}} \tag{2}$$

, where $\bar{R}_u$ and $\bar{R}_v$ denote the average rating of user *u* and user *v*, respectively.

(2) Adjusted-cosine similarity: Cosine-based similarity does not take into account differences in rating scale between different users. The adjusted cosine similarity overcomes this drawback by subtracting the corresponding item average from each co-rated pair. Formally, the similarity between user *u* and user *v* is given by:

$$\mathrm{sim}(u, v) = \frac{\sum_{i \in I_{u,v}} (R_{u,i} - \bar{R}_u)(R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_u} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{i \in I_v} (R_{v,i} - \bar{R}_v)^2}} \tag{3}$$

2.2 Prediction Computation

User-Based CF calculates the predictive value $P_{u,i}$ for an object user *u* and a specific item *i* by the following Equation:

$$P_{u,i} = \bar{R}_u + \frac{\sum_{a=1}^{k} (R_{a,i} - \bar{R}_a) sim(u,a)}{\sum_{a=1}^{k} sim(u,a)} \tag{4}$$

,where $\bar{R}_u$ and $\bar{R}_a$ are average rating of user *u* and user *a* respectively, $sim(u, a)$ is the similarity between *u* and *a*, *k* is the size of nearest neighbors.

After above steps, we recommend the Top N items with biggest $P_{u,i}$ to object user.

3. A new Algorithm based on Slope One and Clustering

In this paper, we first introduce the item feature similarity into Slope One algorithm, then combine it with user clustering algorithm based on user attributes, finally apply to collaborative filtering recommendation. The method mainly solve two issues: the reduce of accuracy with the increasing sparsity resulted from new users and items entering the system; the worse scalability is due to the surge of nearest neighbor searching space caused by the expansion of system.

3.1. Slope One based on item attribute similarity

The Slope One algorithm is first proposed by Doctor Daniel Lemire in 2005 [13]. Its basic form of the predictor is $f(x) = x + b$. Suppose U_i and U_j represent the sets of users rated item *i* and item *j* respectively, the process to calculate R_{uj}can be divided into two steps:

(1) Suppose the ratings of item *i* and *j* by user *v* meet a linear relationship $R_{vj} = R_{vi} + d_{ji}$, where d_{ji} can be formulated as follows:

$$d_{ji} = \sum_{v \in U_i \cap U_j} \frac{R_{vj} - R_{vi}}{|U_i \cap U_j|} \tag{5}$$

(2) Suppose the rating of item *i* by user *u* is represented by r_{ui}, we can take advantage of Eq.(5) to predict rating by *u* to *j* as $R_{uj} = r_{ui} + d_{ji}$.

From above steps, we can see that the Slop One algorithm is an incremental model has features of simple, efficient and can continue self-learning with the joining of users and items. Slope One algorithm can reduce matrix sparsity and improve recommendation accuracy to some extent, but the time consumption increases with the system expanding. Moreover, the predicting process only considers the similarity between users' preferences, doesn't take into account similarity between item attributes, and resulting the prediction lack of correlation. So in this paper, on the basis of weighted Slope One algorithm, introducing the item attributes features, we only select rating data of the Top KN items whose comprehensive similarity is much higher with the object item to exclude the interference from non-neighbors; besides, the algorithm can reduce the size of item neighbors reducing time-consuming.

The item attributes matrix can be written as bellow:

Table 1 Item-attribute matrix

Item	s_1	s_2	⋯	s_l
i_1	0	1	⋯	0
i_2	1	0	⋯	1
…	⋯	⋯	⋯	⋯
i_n	0	1	⋯	0

The row of the matrix represents item, the column represents attribute, the value 1 represents the item has the attribute and 0 represents no. So that each item *i* can be denoted as $i = \{p_1, p_2, \ldots, p_l\}$ where $p_t \in \{0,1\}, 1 \leq t \leq l$.

The process of Slope One based on item attribute similarity is shown below

Step 1 Calculate attribute similarity between item *i* and *j* according to Eq. (2), denoted as $sim_s(i,j)$. It can be seen from the formula, the more the same attributes, the higher the similarity is.

Step 2 Calculate rating similarity between items based on matrix *R* (mentioned in Section 2) using Eq. (2), denoted as $sim_r(i,j)$.

Step 3 Use the results obtained from *Step 1*and *Step 2* to calculate comprehensive similarity $sim(i,j)$, formulated as bellow:

$$sim(i,j) = \alpha sim_s(i,j) + (1 - \alpha) sim_r(i,j) \tag{6}$$

Step 4 Sort $sim(i,j)$ in descending order; take the top KN items as neighbor set.

Step 5 Based on item neighbors, apply weighted Slope One scheme on object users, then fill vacancies in R with the predicted value.

3.2. Clustering based on user attributes

Traditional collaborative filtering algorithm searches neighbors for object user according to ratings of all users, which regards each score as equally important. This operation neglects the differences of data, and will introduce bigger error and reduce the precision of recommendation. For example, an 80-year-old and a 20-year-old gave 5 points to the same film, but we think the meaning is different although the score is the same. In contrast, we believe it's much more meaningful if both a man of 20 and a 30 gave 5 points to the same movie. To achieve dimension reduction, this article employs ant colony algorithm [14] based on self-organization, introduces the user's own characteristic factors, and divides users into different groups by clustering. Thus, neighbor searching can just conduct in a cluster, improving response rate, increasing the proportion of similar users' ratings, and further improve the accuracy of the system.

We outline the clustering algorithm based on self-organization as follows:

Step 1. Consider users as data objects, namely ants in colony, denoted as $a_i, i = 1,2, \dots, m$, m is the size of colony. Initially, assume that in an ant nest, there are k roots forming k branches to which an ant adheres, these ants compose the initial cluster centers, denoted as $C_j, j = 1,1, \dots, k$.

Step 2. The ants entered later will compute similarities between themselves and all cluster centers and choose the biggest one to adhere. User attributes of experimental data used in this paper include age, gender and occupation. The attribute similarity $sim_f(u, v)$ between user u and v can be expressed as following:

$$sim_f(u,v) = \alpha sim_a(u,v) + \beta sim_g(u,v) + (1 - \alpha - \beta) sim_o(u,v) \qquad (7)$$

, where $sim_a(u,v)$, $sim_g(u,v)$ and $sim_o(u,v)$ denote the similarity of age, gender and occupation, respectively. α and β are adjustment factors indicating how important the characteristics are to comprehensive similarity.

Step 3. Update cluster centers, and then repeat from Step 2 until the clusters is not changing, then output the user clusters. We describe the updating process as follows:

i. Calculate the average age of each cluster and figure out the mode of gender and occupation in every cluster;
ii. Consider the 3 values of each cluster as their temporary center;
iii. Compute distance of all users from each temporary center on the three attributes.
iv. Choose user with minimum distance in each cluster as new cluster

We can see that all users are classified into a cluster, without omissions. But the algorithm has drawbacks of predefined number of clusters and random initial centers.

3.3. A Slope One and Clustering based Collaborative Filtering Algorithm

The algorithm proposed in this paper is based on Section 3.1 and Section 3.2, introduces the item feature similarity into Slope One algorithm to fill rating matrix, reducing sparsity; cluster users based on user features similarity; choose some users in the same cluster as neighbors the with object user, and then calculate the prediction; At last, get final recommendation. The framework is described as follows:

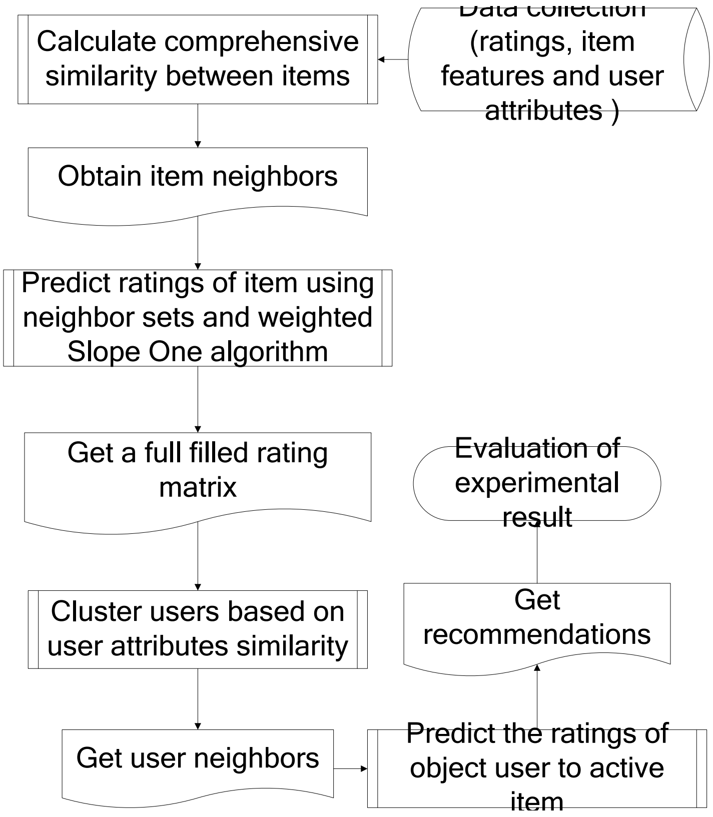
Calculate comprehensive similarity between items
Data collection (ratings, item features and user attributes)
Obtain item neighbors
Predict ratings of item using neighbor sets and weighted Slope One algorithm
Get a full filled rating matrix
Evaluation of experimental result
Cluster users based on user attributes similarity
Get recommendations
Get user neighbors
Predict the ratings of object user to active item

Figure 1. Framework of proposed recommendation algorithm

4. Experimental Results and Evaluation

4.1. Datasets

We uses the Movielens data set including 100,000 ratings for 1682 movies by 943 users. The ratings in the data set are integers on a numerical scale from 1 to 5, with the bigger integer representing the stronger preference. In order to compare the effect, we employ 5-fold cross validation method, that is, randomly extract 250 users and 250 movies to compose a training set and a test set corresponding to the training set. Extract a total of five groups of such data sets.

4.2 Evaluation Metrics

For evaluating the quality of a recommender system, researchers have used several measures, shown in [15]. Among them, the mean absolute error (MAE) is one of the most widely used metrics. MAE is easy to understand and can measure the quality of recommendation systems in an intuitive way. In this paper, we used MAE as the evaluation metrics. The MAE is computed by first summing the absolute errors of the N corresponding ratings-prediction pairs and then averaging the sum. Formally,

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{8}$$

,where N is the total number of true and predicted rating pairs, q_i is the true rating value and p_i is the predicted rating value by CF method.

4.3. Experimental procedure

Our experiments are divided into three parts. The first part validates Slope One collaborative filtering algorithm based on item attributes similarity. In the second part, we verify collaborative filtering algorithm based on attributes clustering. Finally, we analyze the performance of collaborative filtering algorithm based on Slope One and clustering.

The experimental process using 5 groups of datasets mentioned in Section 4.1, experiments on them respectively, and uses the averaged results as experimental results.

(1) Validation of Slope One algorithm based on item attributes similarity

In this part, we apply **S**lope One algorithm based on item attributes similarity and traditional Slope One scheme into collaborative filtering, respectively, and then compare the effect. The number of object users is 15; the number of neighbors is from 10 to 50 increased by 10. Finally, the experimental results are shown in Figure 2

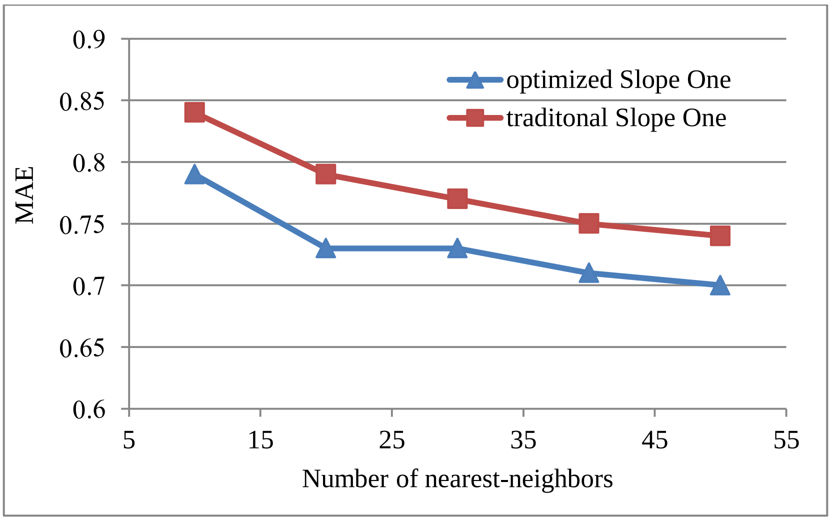

Figure 2. Comparison of traditional Slope One algorithm and Slope One based on item attributes similarity

We can observe that the recommendation errors of Slope One based on item attributes similarity are significant less than the traditional Slope One algorithm.

(2) Validation of clustering based on user attributes

Here we compare the MAE of collaborative filtering algorithm based on attributes clustering and traditional collaborative filtering algorithm. We first divide users into 7 clusters according to their age, and then keep iterating until the clusters are not changing. The parameter settings are the same as Part (1) and the results is shown in Figure 3.

Experimental results show that the introduction of clustering based on user attributes can clearly reduce recommender errors and get a better recommendation results just using a fewer neighbors. However, with the increases of nearest-neighbors, the error is unstable, we think the reason is that the clustering process resulted the user space smaller, and the number of neighbors is too large and the actual similarity between users is too low to represent the true meaning of the nearest neighbor, thus affecting the recommendation result. Therefore, we shouldn’t choose too many neighbors in actual applications.

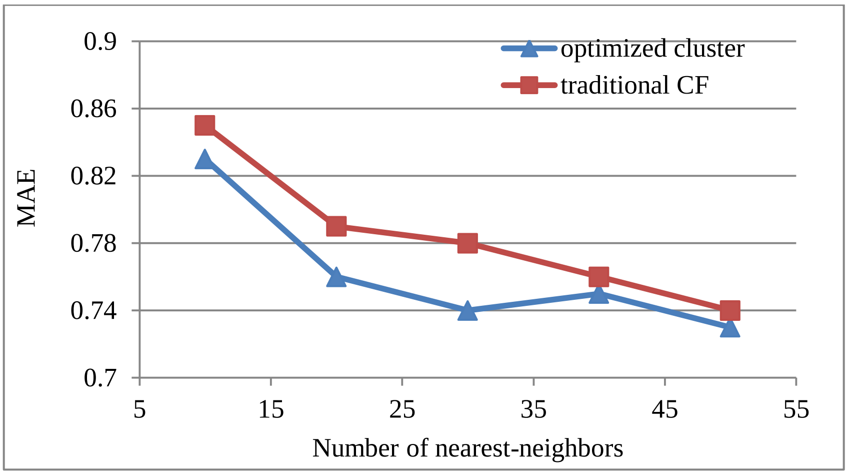

Figure 3. Comparison of traditional CF algorithm and CF introduced clustering based on user attributes

(3) Validation of CF based on Slope One and clustering

In this part, we compare the CF based on Slope One and Clustering with traditional CF algorithm. We adopt increasing number of object users to experiment for each data set, separately. Here we display cases of 10 and 15 object users, shown in Figure 4 and Figure 5, respectively.

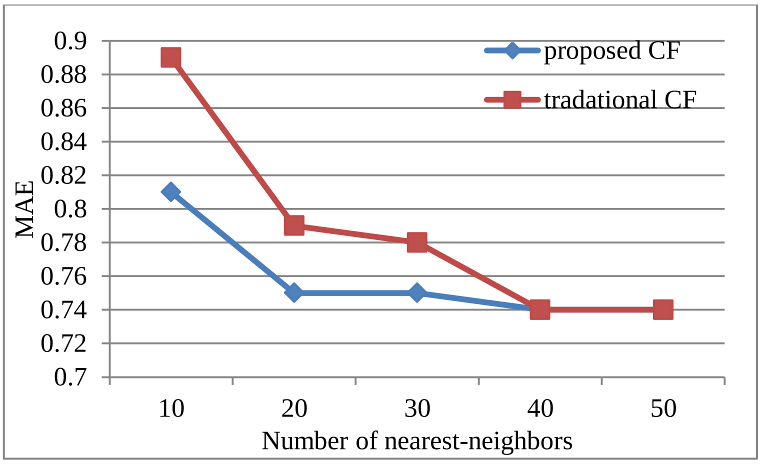

Figure 4. Comparison of traditional CF and CF introduced optimized Slope One and clustering when chosen 10 object users

As can be seen from Figure 4, the proposed algorithm is better than or similar to the traditional CF regardless of the number of neighbors. Especially in the case of a small number of neighbors, the effect is even more pronounced. From Figure 5, we can see that with the increase in the number of neighbors, the proposed algorithm is superior to the traditional CF algorithm, and when there are just 20 neighbors, the recommendation has been relatively stable.

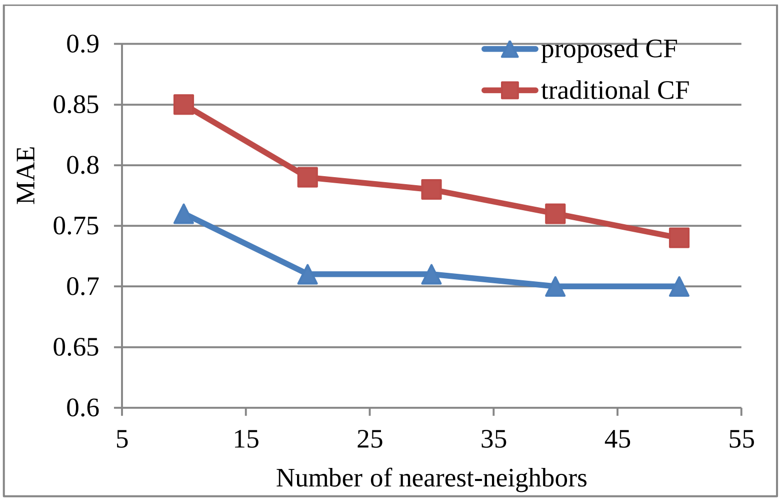

Figure 5. Comparison of traditional CF and CF introduced optimized Slope One and clustering when chosen 15 object users

Finally, we compare all the four algorithms including traditional CF algorithm, the CF based on optimized Slope One, the CF based on optimized clustering and the proposed CF algorithm, shown in Figure 6.

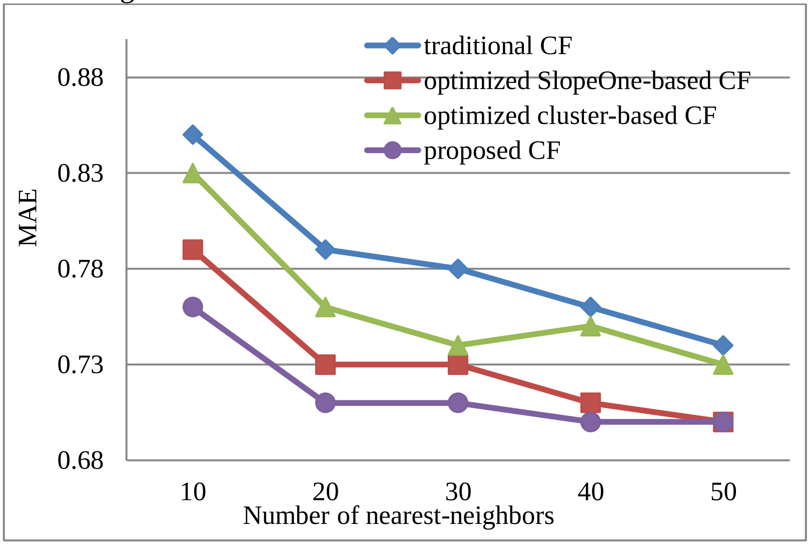

Figure 6. Comparison of four algorithms: traditional algorithm, the CF based on optimized Slope One, the CF based on optimized clustering and the proposed CF algorithm

We can see that both the second and the third algorithm are better than the first algorithm in accuracy, but with the increase of neighbors, the recommendation quality is unstable. This is because the data is extremely sparse, although the effects of irrelevant users can be excluded by Slope One or clustering to some extent, the actual degree of similarity has become very low due to the reduction of data dimension, resulting in reduced quality of recommendation. The proposed algorithm makes up for the shortcomings of the two algorithms, reducing the sparsity of data, narrowing the search space of neighbors, not only improves the efficiency of recommendation, but also improves the accuracy, makes more effective recommendation.

5. Conclusion

We proposed a Slope One and clustering based collaborative filtering algorithm against the traditional CF's problem of sparsity and scalability. We introduce the item feature similarity into Slope One algorithm to reduce rating sparsity, then combine it with ants clustering based on user attributes to lessen the search space and exclude the effects of irrelevant users. Experimental results on Movielens show that the new algorithm can efficiently improve recommendation quality.

Acknowledgment

This work is supported by the Fundamental Research Funds for the Central Universities (with number: 14CX02030A).

References

[1] Yang X, Guo Y, Liu Y, et al. A survey of collaborative filtering based social recommender systems[J]. *Computer Communications*, 2014, 41: 1-10.

[2] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.

[3] Melville, P., Mooney, R. J., & Nagarajan, R. (2002, July). Content-boosted collaborative filtering for improved recommendations. In AAAI/IAAI (pp. 187-192).

[4] Garcia, I., Sebastia, L., & Onaindia, E. (2011). On the design of individual and group recommender systems for tourism. Expert Systems with Applications, 38(6), 7683-7692.

[5] Luo, H., Niu, C., Shen, R., & Ullrich, C. (2008). A collaborative filtering framework based on both local user similarity and global user similarity.Machine Learning, 72(3), 231-245.

[6] Lee, J. S., & Olafsson, S. (2009). Two-way cooperative prediction for collaborative filtering recommendations. Expert Systems with Applications,36(3), 5353-5361.

[7] Li, Y., Lu, L., & Xuefeng, L. (2005). A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce.Expert Systems with Applications, 28(1), 67-77.

[8] Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. *Journal of Software*, 5(7), 745-752.

[9] Vozalis, M. G., & Margaritis, K. G. (2007). Using SVD and demographic data for the enhancement of generalized collaborative filtering. *Information Sciences*, 177(15), 3017-3037.

[10] Lemire, D., & Maclachlan, A. (2005, April). Slope One Predictors for Online Rating-Based Collaborative Filtering. In SDM (Vol. 5, pp. 1-5).

[11] Mi, Z., & Xu, C. (2012). A recommendation algorithm combining clustering method and slope one scheme. In *Bio-Inspired Computing and Applications*(pp. 160-167). Springer Berlin Heidelberg.

[12] Dorigo, M. (1992). Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*.

[13] Lemire, D., & Maclachlan, A. (2005, April). Slope One Predictors for Online Rating-Based Collaborative Filtering. In *SDM* (Vol. 5, pp. 1-5).

[14] Colorni, A., Dorigo, M., & Maniezzo, V. (1991, December). Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life* (Vol. 142, pp. 134-142).

[15] Herlocker, J. L., Konstan, J. A., Terveen, L. G., John, & Riedl, T. (2004). Evaluating collaborative filtering recommender systems. *Acm Transactions on Information Systems, 22*, 5--53.