ICIC Express Letters
Volume **11**, Number **8**, August **2017**

ICIC International ©2017 ISSN 1881-803X
pp. 1339–1350

A MULTI-VIEW SIMILARITY BASED CONTEXT-AWARE RECOMMENDATION ALGORITHM

Li Yang[1] and Guirong Shao[2]

[1]Public Department of Computer Teaching
[2]Department of Physics and Electronic Engineering
Yuncheng University
No. 1155, West Fudan Street, Yuncheng 044000, P. R. China
yli809@163.com

Received January 2017; accepted April 2017

Abstract. *To solve the problem that the existing recommendation algorithms have low predictive accuracy which is caused by data set sparsity when getting users' preference, a multi-view similarity based context-aware recommendation algorithm is proposed. This method first gets the direct similarity among users' historic preference behavior by the conception similarity and Pearson correlation coefficient. The calculation is based on users' historic preference behavior on the same and similar commodities. Then the method gets the incidence relation between commodities and preference commodities set of users who need to get their prediction based on the commodity property context. And it combines similarity of the two views to acquire the nearest neighbor users, based on which it elicits users' preferences by adopting improved user-based collaborative filtering. The authors present empirical experiments by using a real extensive data set. The experimental results show that the proposed algorithm can achieve better prediction accuracy and diversity compared with collaborative filtering algorithm and context-aware algorithm.*
Keywords: Recommendation system, Multi-view similarity, Behavior similarity

1. **Introduction.** With the advent of service-oriented and pervasive computing, the amount of information resources is increasing far more quickly than our ability to process it. If only depending on users to process it, the expenditure of time is unacceptable, and the problem of information overload is getting worse. Many technologies, such as information retrieval and search engine, can alleviate the problem to some extent, and reduce the time expenditure of users to find their preference information. However, most of those technologies are suffering from low intelligence and high error rate; especially, they cannot offer personalized recommendation service when facing users who have different educational backgrounds and working environments. Under this situation, recommender systems emerge in response to those problems and have been successful and widely used in electronic commerce. Their main goals are to help users find personalized preference through recommendation algorithms based on users' historical preference behavior. Over the years, many researchers have been proposed different ways to realize recommendation, which can be categorized into two classes: (1) content-based recommendation methods [1], which use mathematical models and data mining to get prediction; (2) collaborative filtering [2], which gets users' preference by filtering the uncorrelated items based on a collection of similar users' rating history. However, those methods always suffer from cold-start problem and sparsity problem, among which the former means recommender systems cannot provide accurate prediction for users who have no preference behavior in those recommender systems, and the other means that most users only have very few behaviors in recommender systems which are insufficient for those systems to build their preference models. To solve those problems, Bobadilla et al. [3] present a new similarity

measure method using optimization based on neural learning, which exceeds the best results obtained with current metrics. Gao et al. [4] incorporate the weight and rank of a user into the computation of item similarities and differentials, improving the recommendation results of the typical Adjusted Cosine and Slope One item-based CF approaches. Tang et al. [5] propose a method of location-aware collaborative filtering to recommend Web services to users by incorporating locations of both users and services, and concentrate on users physically near to the target user. Jiang et al. [6] account the personalized influence of services when computing similarity measurement between users and personalized influence of services, which improves accuracy of recommendation of Web services significantly. Xu [7] proposes a method to make context-aware and trip similarity based travel recommendations by mining CCGPs. Zheng et al. [8] study how context similarity can be integrated into the sparse linear method and matrix factorization algorithms.

Almost all of those approaches get users' predictions based on the original development trend of resources or the historic preference. However, since users' preference behaviors have interaction effects on the real environment, to only base on those preference behavior without considering the interaction effect may have a negative influence on the accuracy of prediction. In this paper, we present a multi-view similarity based context-aware recommendation algorithm. This method first gets the direct similarity among users' historic preference behavior by the conception similarity and Pearson correlation coefficient method. The calculation is based on the historic preference behavior which is on the same and similar commodity. Then the method gets the incidence relation between commodities and preference commodities set of users who need to get their prediction based on the commodity property context. And it combines similarity of the two views to acquire the nearest neighbor users. Finally, a weighted prediction computation algorithm based on improved user-based collaborative filtering is employed to get the recommendation.

2. A Multi-View Similarity Based Context-Aware Recommendation Algorithm.

2.1. Context-aware recommender system.

At present, there is still no standard definition of context. Most researchers use the definition given by Dey [8] that context is the information which expresses the condition of environment by an explicit or implicit way. It can be user, location, or any correlation object. The definition of k different types of context can be given by

$$C = (C_1, C_2, C_3, \dots, C_k) \tag{1}$$

where $C_i \subset C$ $(i \in [1, k])$ represents any type of one-dimensional context vector, and $C_i = \{c_{i_p} \,| p \in [1, q_i]\}$. q_i is the number of the information of this type. An example is as below

$$\begin{array}{l} C_i : \mathit{Emotion}\,(\mathit{happy}[1], \mathit{gloomy}[2], \mathit{sad}[3], \mathit{angry}[4]) \\ C_j : \mathit{Location}\,(\mathit{office}[1], \mathit{communication}[2], \mathit{home}[3], \mathit{school}[4], \mathit{restaurant}[5]) \end{array} \tag{2}$$

In the condition of context, the traditional user-item prediction model can be extended to a multidimensional prediction model R : *User* $\times$ *Context* $\times$ *Item* $\to$ *Rating* [9], and another example is as follows

$$\begin{array}{l} \mathit{User} \subseteq \mathit{UName} \times \mathit{Address} \times \mathit{Income} \times \mathit{Age} \\ \mathit{Item} \subseteq \mathit{IName} \times \mathit{Type} \times \mathit{Price} \\ \mathit{Context} \subseteq \mathit{Location} \times \mathit{Emotion} \times \mathit{Time} \times \mathit{Environment} \end{array} \tag{3}$$

The formal definition of context-aware recommender system is as follows. If $UD_{pre} = \{D_{j1}, \dots, D_{j(Ent_l)}\}$ $(j \in [1, U], Ent_l \subseteq S_{vr})$ is the initial goal space, $D_{pre} = \{D_{i1}, \dots, D_{i(Ent_k)}\}$ $(i \in [1, U], Ent_k \subseteq S_{vr})$ is the result goal space, $UD_{pre} \cap PD_{pre} = \emptyset$, and utility function $F_{uty}(\cdot)$ is used to compute users' preference Pre_{jis} under multidimensional context information, then the primary purpose of recommender system is to find the favorite items $d_{i1}, \ldots, d_{i(Ent_k)}$ for $d_{j1}, \ldots, d_{j(Ent_l)}$. The definition is as below

$$\begin{aligned}&\forall(d_{j1}, \ldots, d_{j(Ent_l)}) \subseteq UD_{pre},\\ &(d_{i1}, \ldots, d_{i(Ent_k)}) = \arg\max F_{uty}(UD_{pre}, D_{pre})\end{aligned} \tag{4}$$

2.2. **Multi-view similarity based recommendation algorithm.** Inspired by the theory of multi-view learning, a preference elicit method, based on the views of user and item is proposed to get the users' trust on items. (1) From the point of users, we compute the similarity among users based on the historic preference behavior, and get the interactive affection among users' preference behavior because the method uses the real historic preference records which lead to more accurate prediction. (2) From the point of items, we obtain the incidence relation among items based on the inherent properties of items, and combine the historic preference on the given items to acquire the relevance among items. (3) The similarity of the two views is combined to acquire the nearest neighbor users, based on which users' preferences are elicited by the improved user-based collaborative filtering. The whole process of the proposed method is shown in Figure 1, the method uses direct similarity and similarity among users to generate prediction based on users' historical preference, and the detail is described in Subsections 2.2.1-2.2.3.

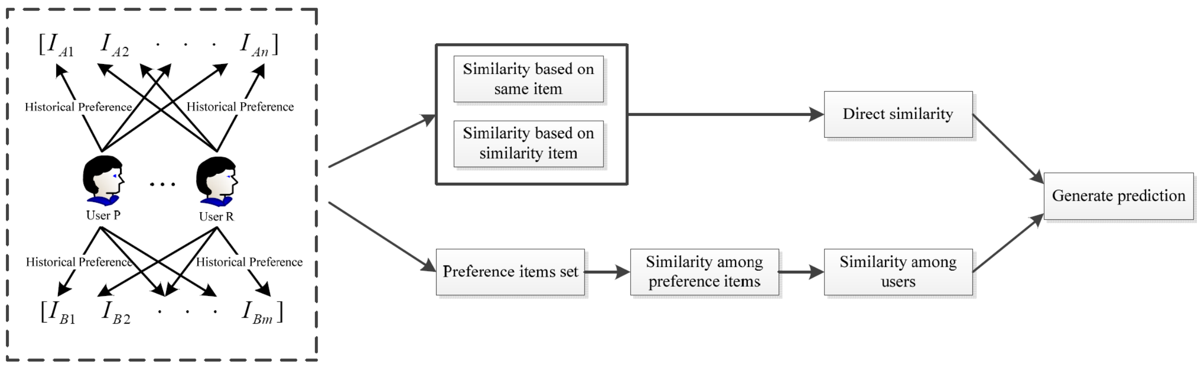

FIGURE 1. The process of the proposed method

2.2.1. *The measure method of the similarity among users.* The interactive affection among users which is conditioned by human emotion has a significant influence on the users' preference model [10]. The behavior similarity means the similarity among users' preference for the same items or the congeneric items. We employ the direct similarity and indirect similarity which use Pearson correlation coefficient and conception similarity to get behavior similarity. This method can measure the similarity from individual user's preference behavior and the interaction affection among users' preference behavior. The referred definitions are as follows.

Definition 2.1. *Direct Similarity, it is only based on the real historical behavior to get the similarity, which includes the following two steps.*

(1) For users' behaviors which have preference records on the same items, it uses Pearson correlation coefficient to measure the similarity. I_{AB} is the common preference set between user U_A and U_B, which measures the similarity by the differential degree between the two users' historic preference values on I_{AB} and their historic preference average values on other items. It is given by

$$S_1(U_A, U_B) = \frac{\sum\limits_{i\in I_{AB}} \left(P_{Ai} - \overline{P_i}\right)\left(P_{Bi} - \overline{P_i}\right)}{\sqrt{\sum\limits_{i\in I_{AB}} \left(P_{Ai} - \overline{P_i}\right)^2 \cdot \sum\limits_{i\in I_{AB}} \left(P_{Bi} - \overline{P_i}\right)^2}} \tag{5}$$

where P_{Ai} and P_{Bi} are the values of user U_A and U_B for items that belong to I_{AB} respectively, $\overline{P_i}$ is the average value of other users on item I_i.

(2) For users' behavior which have preference records on the similar items, it combines the intrinsic similarity of items based on their properties and users' preference behavior on similarity items. The intrinsic similarity is measured by conception similarity, which means if any two properties belong to a same conception, then the two properties are similar. For example, chrysanthemum and rose are two different things, but they all belong to conception 'Flower', then we can say chrysanthemum and rose are similar. The conception similarity is given by

$$S_2(U_A, U_B) = \omega_{item}(I_i, I_j) \times \omega_{user}(U_A, U_B) \tag{6}$$

$$\omega_{item}(I_i, I_j) = \frac{Card(I_i \cap I_j)}{Card(I_i \cup I_j)} \tag{7}$$

$$\omega_{user}(U_A, U_B) = \frac{|P_{Ai} - P_{Bj}|}{\overline{P_{U_A}} + \overline{P_{U_B}}} \tag{8}$$

where $\omega_{item}(I_i, I_j)$ represents the similarity between I_i and I_j. $\omega_{item}(U_A, U_B)$ is the rating difference between U_A and U_B. $Card(I_i \cap I_j)$ denotes the number of conceptions between I_i and I_j. P_{Ai} is the preference value of user U_A on item I_i. P_{Bj} is the preference value of user U_B on item I_j. $\overline{P_{U_A}}$ and $\overline{P_{U_B}}$ are the average preferences values of user U_A and U_B respectively.

Then the direct similarity among users can be defined by

$$S(U_a, U_b) = \alpha \cdot S_1(U_A, U_B) + (1 - \alpha) \cdot S_2(U_A, U_B) \tag{9}$$

where $S(U_a, U_b)$ is the direct similarity between user U_A and U_B, α denotes the weight parameter of similarity.

2.2.2. *The measure method of the intrinsic similarity among items.* When forecasting users' preference, the properties of items and users' historic preference are two core influence factors, which are also the only available data for recommender systems under sparsity problem. Because users' preference will not change very frequently and largely [11], items selected from historic preference behavior may have much bigger influence and value.

Firstly, we screen the items which have preference records by user (U_u, for example), based on which we acquire the preference commodities set (represented by I_{Pre}). Then we compute the similarity degree between the preference commodities and other items based on the intrinsic properties and historic preference behavior, and finally get the preference degree for other items. The method to get the preference commodities is set as follows

$$I_i \in I_{Pre} \Leftrightarrow P_{ui} \geq P_{Pre} \tag{10}$$

$$P_{Pre} = (1 + \lambda) \cdot \frac{\sum\limits_{i \in I_u} P_{ui}}{Card(P_u)} \tag{11}$$

where P_{Pre} represents the values of users for items which belong to preference commodities set. Only when the preference value of the preference commidities set is bigger than P_{Pre}, can items join them. λ is the decision parameter to control the size of preference commodities and the value of P_{Pre}. $Card(P_u)$ represents the number of U_u's preference behavior, which is equal to the number of items having preference behavior by U_u.

After acquiring the preference commodities set, we measure the correlation relation between preference commodities and other items based on the items' properties and historic preference. It contains two influence factors, (1) the number of same properties among items, and (2) the similarity among properties. The data used to measure the similarity include two parts, (1) the number of items which contain both properties, (2) the number of items which contain the properties respectively. If S_a and S_b are any two properties, their similarity can be measured as below

$$Sim(S_a, S_b) = \frac{Card(S_a, S_b)}{Card(S_a) + Card(S_b) - Card(S_a, S_b)} \tag{12}$$

where $Card(S_a, S_b)$ represents the number of items which contain properties S_a and S_b, $Card(S_a)$ and $Card(S_b)$ are the numbers of items which contain properties S_a and S_b respectively.

After we get the similarity among properties, the intrinsic similarity among items can be measured by

$$Sim_1(I_i, I_j) = \frac{\sum\limits_{S_a \in S_{i-j}, S_b \in S_{j-i}} Sim(S_a, S_b) + Card(S_{ij})}{Card(S_{I_i}, S_{I_j})} \tag{13}$$

where S_{i-j} represents the properties set included by I_i but not by I_j. S_{j-i} is in the properties set which is included by I_j but not by I_i. $Card(S_{ij})$ denotes the number of common properties between item I_i and item I_j. $Card(S_{I_i}, S_{I_j})$ is the number of properties included by item I_i and item I_j.

The similarity among items reflected by users' historic preference behavior has bigger probability according with users' preference than the similarity reflected by items' properties. The similarity based on users' historic preference can be measured by

$$\begin{aligned} & Sim_2(I_i, I_j) \\ = & \frac{1}{\frac{\sum\limits_{U_c \in U_{ij}} |P_{ci} - P_{cj}|}{Card(U_{ij})} + \frac{\sum\limits_{U_d \in U'_{ij}} |P_{di} - P_{dj}|}{Card(U'_{ij})} + 1} \\ = & \frac{Card(U_{ij}) \cdot Card(U'_{ij})}{Card(U'_{ij}) \cdot \sum\limits_{U_c \in U_{ij}} |P_{ci} - P_{cj}| + Card(U_{ij}) \cdot \sum\limits_{U_d \in U'_{ij}} |P_{di} - P_{dj}| + Card(U_{ij}) \cdot Card(U'_{ij})} \end{aligned} \tag{14}$$

where U_c and U_d represent any two users who have preference behavior on I_i or I_j. U_{ij} is the users set which both has preference behavior on item I_i and I_j, U'_{ij} denotes the users set which has preference behavior on item I_i or I_j, $Card(U_{ij})$ and $Card(U'_{ij})$ are the numbers of users in U_{ij} and U'_{ij} respectively.

After we acquired the similarity based on users' historic preference and items' properties, the final similarity computation method was as follows

$$Sim(I_i, I_j) = \frac{Sim_1(I_i, I_j) + Sim_2(I_i, I_j)}{\sqrt{Sim_1^2(I_i, I_j) + Sim_2^2(I_i, I_j)}} \tag{15}$$

2.2.3. *The computation method of prediction.* Based on the similar users' preference behavior, we combine the items which are similar to the preference commodities set to get prediction. To get the prediction, an improved user-based collaborative filtering method is as follows,

$$P_{ui} = \left(1 + \frac{\sum\limits_{U_m \in UN_u} \left|P_{mi} - \overline{P_m}\right| \cdot S(U_u, U_m)}{Card(UN_u)}\right) \cdot \overline{P_i} \cdot (1 + Sim(I_i, I_j)) \tag{16}$$

where UN_u represents the nearest neighbors of U_u based on the users' historic preference behavior. $Card(UN_u)$ is the number of users in UN_u, $\overline{P_m}$ denotes the average preference values of U_m to other items, and $\overline{P_i}$ is the average preference of other users to item I_i.

3. **Experimental Evaluation.**

3.1. **Data set.** There are no existing data sets which include context information; therefore, we generate a simulative data set *MBookCrossing* by adding reasonable context information based on the real-world data set *BookCrossing* [12]. The *BookCrossing* data set is Cai-Nicolas Ziegler which uses crawlers to get the behavior records about 278858 users on 271379 books. The *MBookCrossing* mainly includes four parts.

(1) MBX-Users, which includes users' ID, location, age;

(2) MBX-Books, which includes books' ISBN, title, author, publisher, abbreviation;

(3) MBX-Book-Ratings, which includes users' ratings on books;

(4) MBX-Book-Contexts, that includes time, location, state and so on, in which time includes morning, afternoon, and night; location includes office, home, school, and restaurant; state include work, learning, and amusement.

3.2. **Evaluation metrics.** Recommender systems have different measures to evaluate the quality of recommendation. In this paper, we employ the MAE [13] to measure our prediction. MAE is a representative example of a statistical accuracy criterion, which measures the accuracy by contrasting the real rate and predicted rate gotten from our method. Assume the predicted rating set as $\{s_1, s_2, \ldots, s_N\}$, and the actual rating set as $\{d_1, d_2, \ldots, d_N\}$, then MAE is defined by

$$MAE = \frac{\sum\limits_{i=1}^{N} |s_i - d_i|}{N} \tag{17}$$

Another evaluation method is diversity [14], meaning the dissimilarity among items in recommendation list, in which higher values denote bigger probability that the recommended items cover users' preference. If $s(i, j) \in [0, 1]$ represents the similarity between item I_i and I_j, $|G|$ denotes the length of the recommendation list, then the diversity of the recommendation which represents the average values of all the users' diversity can be defined as follows.

$$Diversity = \frac{1}{|U|} \sum_{u \in U} Diversity_u \tag{18}$$

$$Diversity_u = 1 - \frac{\sum_{I_i, I_j \in G} s(i, j)}{\frac{1}{2} |G| (|G| - 1)} \tag{19}$$

$$S_{item}(i, j) = \frac{\sum\limits_{a \in U_{ij}} \left(P_{ai} - \overline{P_a}\right) \left(P_{aj} - \overline{P_a}\right)}{\sqrt{\sum\limits_{a \in U_{ij}} \left(P_{ai} - \overline{P_a}\right)^2 \cdot \sum\limits_{a \in U_{ij}} \left(P_{ai} - \overline{P_a}\right)^2}} \tag{20}$$

where U_{ij} denotes the users' set which both has rated item I_i and I_j, P_{ai} is the preference value of user U_a on item I_i, $\overline{P_a}$ represents the average historical preference values of user U_a on other items.

3.3. **Experiment results and analysis.**

Experiment 1: Impact of parameter α

We examined the influence of parameter α in Formula (9) on the prediction accuracy based on *MBookCrossing*. In this experiment, we measured the influence of α on users' behavior similarity, and got the users' nearest neighbor by the preference behavior similarity to generate the recommendation list. To improve the accuracy of the experiment, we used ABO method to conceal data in *MBookCrossing* randomly. Since *MBookCrossing* is changing with every operation, we ran the algorithm five times on each weight coefficient, and got the average value as the final accuracy. After repeating test and experiment, we chose several representative series weight coefficients as Table 1, and set the training set as 30% and 70% in this experiment. The experimental results are shown in Figure 2 and Figure 3.

Table 1. Experiment weight series of α

	Sequence 1	Sequence 2	Sequence 3	Sequence 4	Sequence 5
α	0.2	0.3	0.5	0.7	0.9

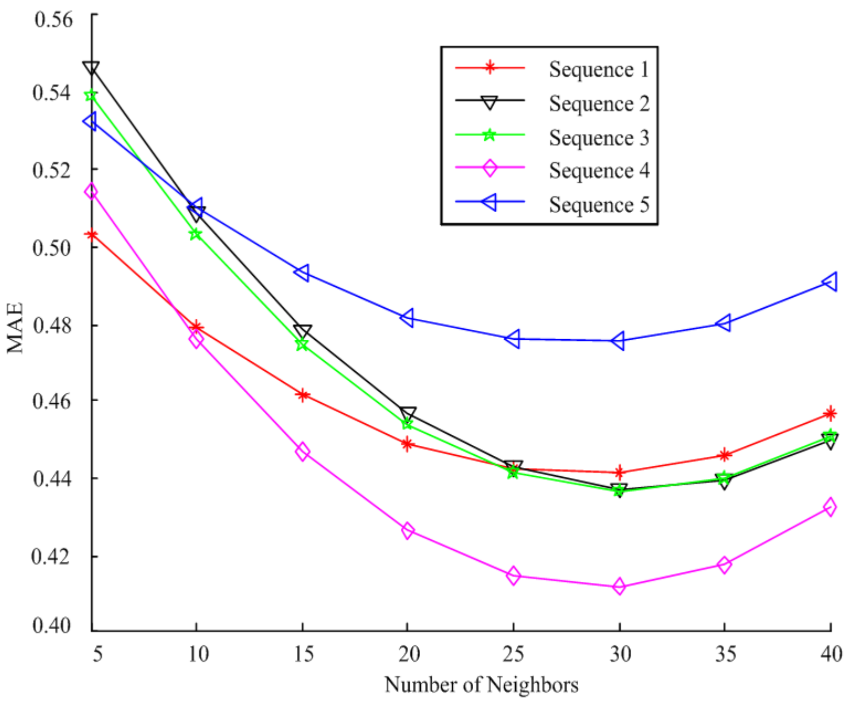

Figure 2. Experimental results by using 30% training set

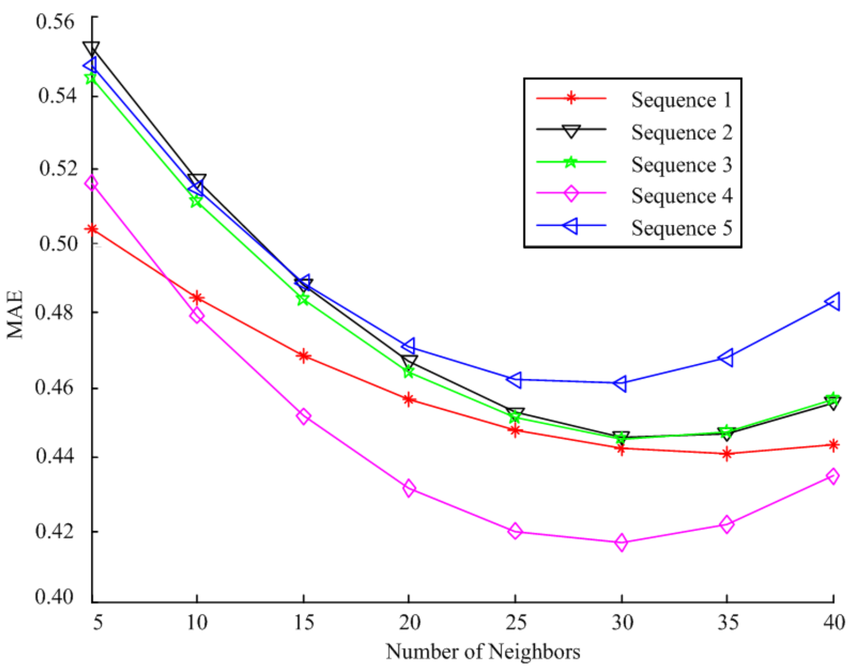

Figure 3. Experimental results by using 70% training set

As shown in Figure 2 and Figure 3, the prediction accuracy first increases and then decreases. When $\alpha = 0.7$ the proposed algorithm achieves the best prediction accuracy, the algorithm also has better results when the training set is 70%. The experimental results show that the same items have large influence on users behavior than similar items.

Experiment 2: Impact of parameter λ

This experiment measured all the possible values of λ on MBookCrossing. To maintain the objectivity of the experimental results, we adopted different ratios of training set and test set. To test the influence of λ on users' preference items set, we used the most similar items obtained by the item similarity method as the recommendation list, and based on which we measured the influence of λ on recommendation accuracy. The process of data set was same as Experiment 1. After repeated tests and experiments, we chose several representative series weight coefficients as Table 2, and we set the training set as 10%, 30%, 50% and 70%. In this experiment, the experimental results are shown in Figure 4 and Figure 5. From the experimental results, we can see that with the adding of λ the prediction accuracy first increases and then decreases, and when $\lambda = 0.48$, the proposed algorithm achieves the best prediction accuracy.

TABLE 2. Experiment weight series of λ

	Sequence 1	Sequence 2	Sequence 3	Sequence 4	Sequence 5
λ	0.24	0.33	0.48	0.53	0.72

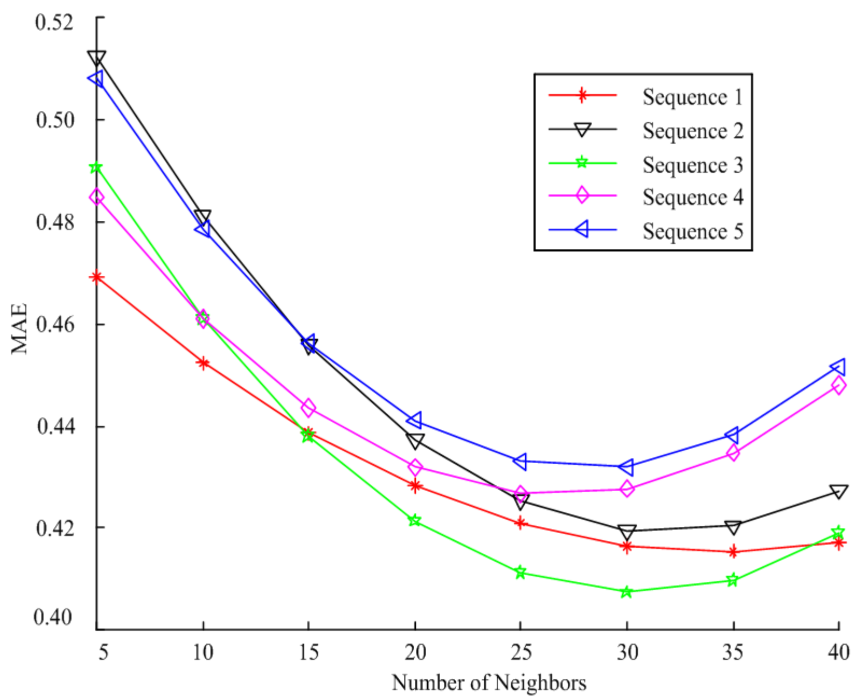

FIGURE 4. Experiment results by using 30% training set

Experiment 3: Comparison with other methods

(1) The comparison of accuracy

Based on Experiment 1 and Experiment 2, we set $\alpha = 0.7$, $\lambda = 0.48$; after repeated test and experiment, we select 30% and 70% as the ratio of training set. Since the famous collaborative filtering algorithm CF [16] is the most classic and widely used recommend algorithm, and context-aware based recommend algorithm MICF [17] owns the same hottest research direction as ours, we compared our algorithm with the CF and MICF.

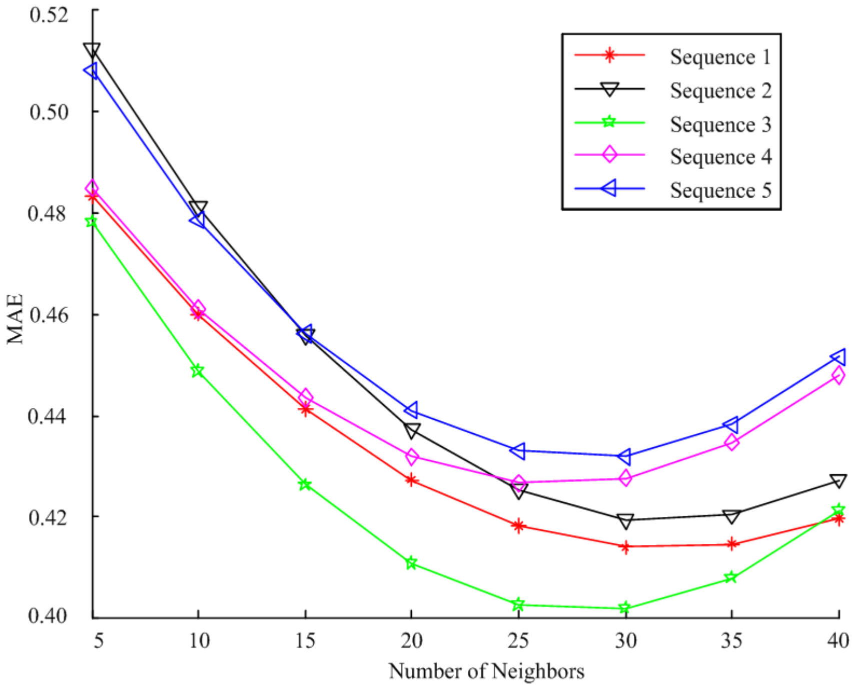

FIGURE 5. Experiment results by using 70% training set

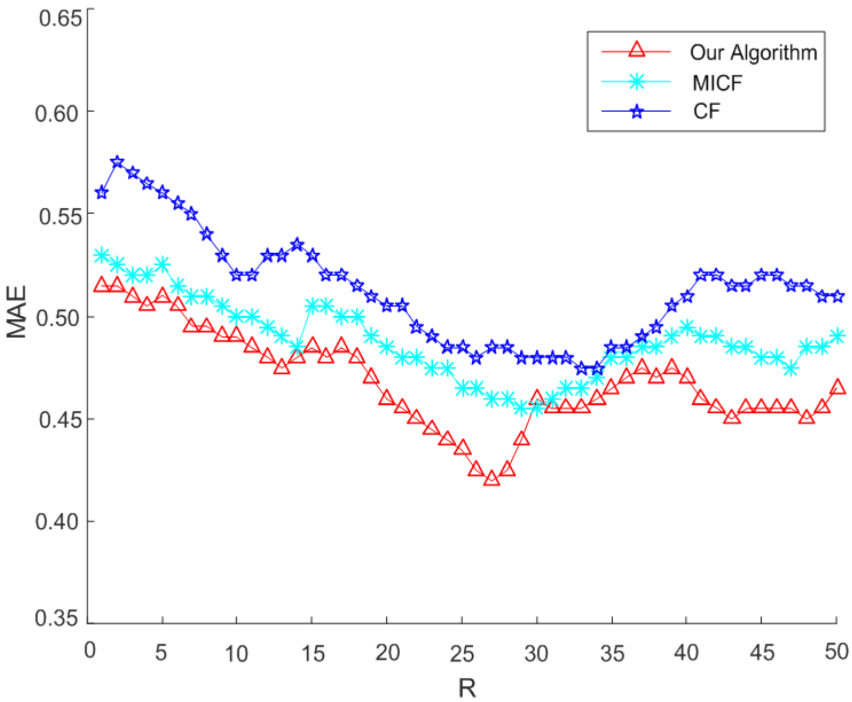

FIGURE 6. The comparison of accuracy with other methods on 30% training set

CF only considers the influence of user-item, which can only use the MBX-Book-Ratings. The experimental results are shown in Figure 6 and Figure 7.

By computing the results shown in Figure 6 and Figure 7, under different training set rates, the MAE of the three methods are increasing with the adding of training set, and the optimal MAE of our algorithm, MICF, CF are 0.42, 0.45, 0.47 respectively, and in the mass, our method has achieved a better prediction accuracy than MICF and CF about 5.38% and 10.52% respectively. It can also be seen from the experimental results that by

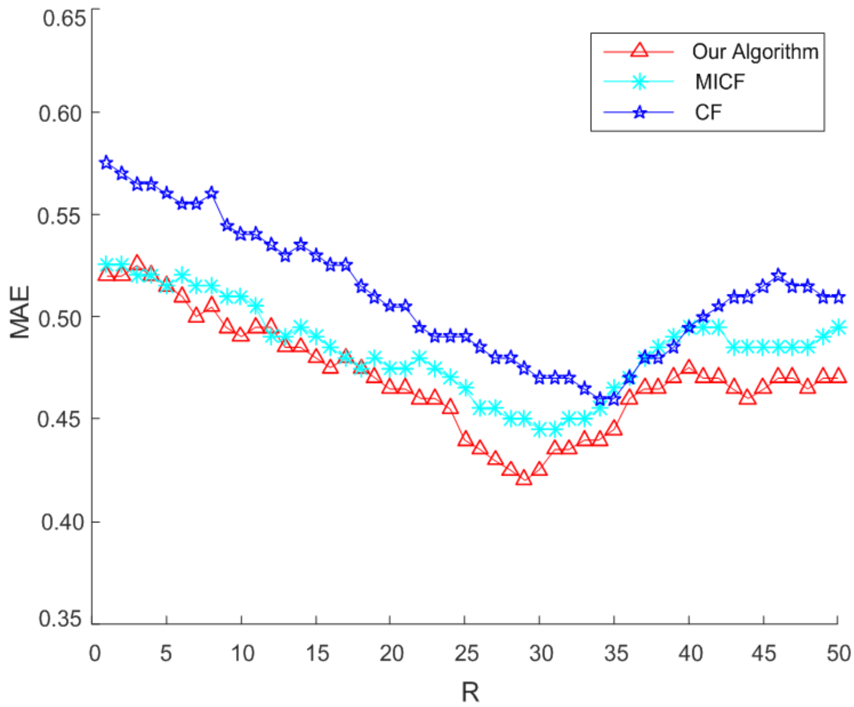

FIGURE 7. The comparison of accuracy with other methods on 70% training set

combining the similarities among users and the intrinsic similarities among items, better recommendation quality can be achieved.

(2) The comparison of diversity

The diversity represents the degree of difference among items in recommendation list. It is another evaluation criterion of recommendation systems which is different from the accuracy. The difference is mainly reflected in the following aspect: the accuracy measures the correlation of the recommendation items and their preference model built by users' historical preference records. The diversity is based on the variety of users' preference, whose higher values denote bigger probability that the recommended items cover users' preference. For instance, user A is fond of horrid fiction and romantic fiction, but most of his/her historical preference records are horrid fiction. If the recommendation systems just consider the accuracy, they will only recommend the corresponding horrid fiction, and the feedback of the evaluation criterion will support this process. However, the user A's real demands may be on the opposite. In this situation, the diversity needs to be considered together with accuracy. The comparion results are shown in Figure 8 and Figure 9.

As shown in Figure 8 and Figure 9, which denotes that the three methods' average diversity relay on the scale of training sets and length of recommendation list, but there are no linear relationship among those factors. And we have found that our method has advantages over other two methods, the possible reasons are analyzed as follows. Our method obtains the prediction from the views of user, item and the relationship among user-user, and user-item, with the consideration of the influence factors which relate to users' preference. It diminishes the influence of users' historical preference on recommendation list relatively, and increases the influence of the own properties of user and item.

4. **Conclusion.** In this paper, we focus on the study of recommender approaches to solving the problems in the existing approaches' prediction accuracy which is often influenced by the sparse data set. We proposed an improved recommendation algorithm

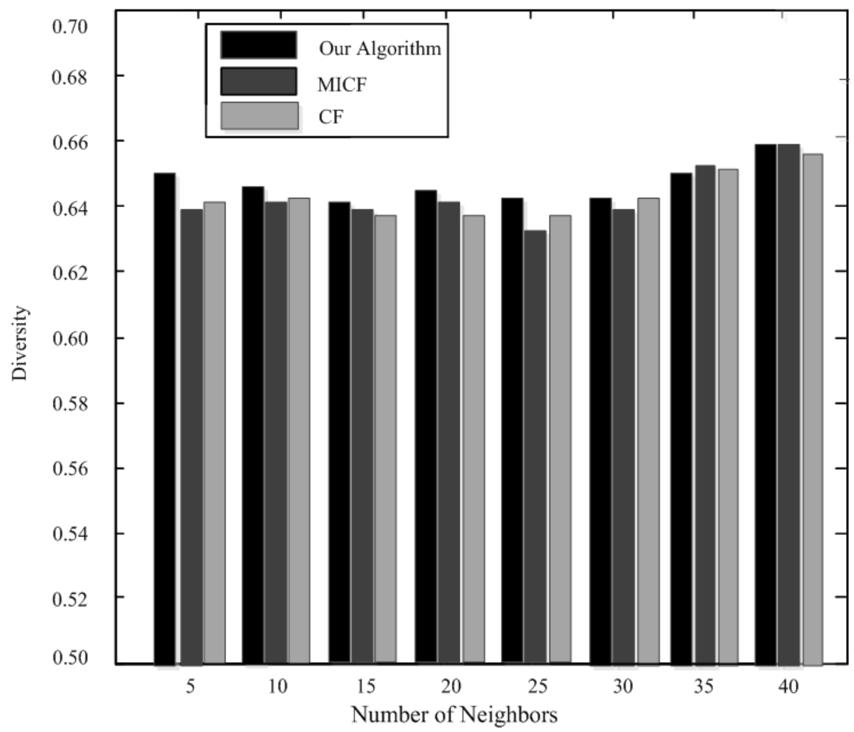

Figure 8. The comparison of accuracy with other methods on 30% training set

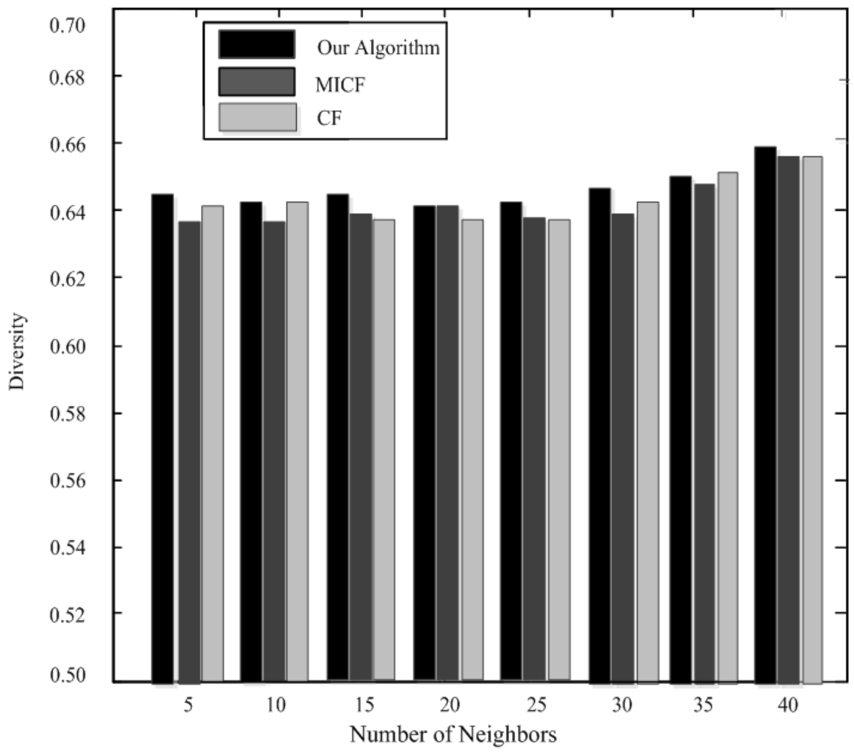

Figure 9. The comparison of accuracy with other methods on 70% training set

based on multi-view similarity. This method first gets the direct similarity among users' historic preference behavior by the conception similarity and Pearson correlation coefficient. The calculation is based on the historic preference behavior on the same and similar commodities. Then the method gets the incidence relation between commodities and preference commodities set of users who need to get their prediction based on the commodity property context. Moreover, it combines similarity of the two views to acquire the nearest neighbor users, based on which it elicits user preferences by adopting an improved user-based collaborative filtering. The experiments in a real world data set shows that our prediction can achieve better prediction accuracy and diversity. Our future work will focus on the context-aware recommender systems to achieve better recommendation accuracy.

REFERENCES

[1] S. Wei, N. Ye, S. Zhang, X. Huang and J. Zhu, Collaborative filtering recommendation algorithm based on item clustering and global similarity, *The 5th International Conference on Business Intelligence and Financial Engineering (BIFE)*, pp.69-72, 2012.
[2] K. Niemann and M. Wolpers, A new collaborative filtering approach for increasing the aggregate diversity of recommender systems, *Proc. of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.955-963, 2013.
[3] J. S. Bobadilla, F. Ortega, A. Hernando et al., A collaborative filtering approach to mitigate the new user cold start problem, *Knowledge-Based Systems*, vol.26, pp.225-238, 2016.
[4] M. Gao, Z. Wu and F. Jiang, Userrank for item-based collaborative filtering recommendation, *Information Processing Letters*, vol.111, no.9, pp.440-446, 2015.
[5] M. Tang, Y. Jiang, J. Liu et al., Location-aware collaborative filtering for QoS-based service recommendation, *IEEE the 19th International Conference on Web Services (ICWS)*, pp.202-209, 2014.
[6] Y. Jiang, J. Liu, M. Tang et al., An effective web service recommendation method based on personalized collaborative filtering, *IEEE International Conference on Web Services (ICWS)*, pp.211-218, 2011.
[7] Z. Xu, Trip similarity computation for context-aware travel recommendation exploiting geotagged photos, *International Conference on Data Engineering*, pp.330-334, 2014.
[8] Y. Zheng, B. Mobasher, R. Burke et al., Similarity-based context-aware recommendation, *Web Information Systems Engineering*, pp.431-447, 2015.
[9] A. K. Dey, Understanding and using context, *Personal and Ubiquitous Computing*, vol.5, no.1, pp.4-7, 2001.
[10] S. C. Kim, K. J. Sung, C. S. Park et al., Improvement of collaborative filtering using rating normalization, *Multimedia Tools and Applications*, pp.1-12, 2013.
[11] N. Sahoo, P. V. Singh and T. Mukhopadhyay, A hidden Markov model for collaborative filtering, *MIS Quarterly*, vol.36, no.4, pp.1329-1356, 2012.
[12] C. N. Ziegler, S. M. McNee, J. A. Konstan et al., Improving recommendation lists through topic diversification, *Proc. of the 14th International Conference on World Wide Web*, New York, pp.22-32, 2015.
[13] H. Wu, K. Yue, X. Liu et al., Context-aware recommendation via graph-based contextual modeling and postfiltering, *International Journal of Distributed Sensor Networks*, vol.2015, no.3, pp.1-10, 2015.
[14] A. Barragansmartinez, E. Costamontenegro and J. Burguillo, A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition, *Information Sciences*, vol.180, no.22, pp.4290-4311, 2016.
[15] H. Liu, Z. Hu and A. Mian, A new user similarity model to improve the accuracy of collaborative filtering, *Knowledge Based Systems*, pp.156-166, 2014.
[16] K. Choi and Y. Suh, A new similarity function for selecting neighbors for each target item in collaborative filtering, *Knowledge-Based Systems*, vol.37, pp.146-153, 2013.
[17] S. Rana, A. Jain and V. K. Panchal, Most influential contextual-features [MICF] based model for context-aware recommender system, *Proc. of International Conference on Emerging Trends in Communication Control*, Bengaluru, India, pp.1-6, 2013.