Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering

Jianrui Chen¹ · Chunxia Zhao² · Ulji² · Lifang Chen²

Received: 24 September 2017 / Accepted: 16 September 2019 / Published online: 23 October 2019
© The Author(s) 2019

Abstract

In recent years, application of recommendation algorithm in real life such as Amazon, Taobao is getting universal, but it is not perfect yet. A few problems need to be solved such as sparse data and low recommended accuracy. Collaborative filtering is a mature algorithm in the recommended systems, but there are still some problems. In this paper, a novel collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering is presented. Firstly, score matrix is pre-processed with normalization and dimension reduction, to obtain denser score data. Based on these processed data, clustering principle is generated and dynamic evolutionary clustering is implemented. Secondly, the search for the nearest neighbors with highest similar interest is considered. A measurement about the relationship between users is proposed, called user correlation, which applies the satisfaction of users and the potential information. In each user group, user correlation is applied to choose the nearest neighbors to predict ratings. The proposed method is evaluated using the Movielens dataset. Diversity experimental results demonstrate that the proposed method has outstanding performance in predicted accuracy and recommended precision.

Keywords Collaborative filtering · User correlation · Dynamic evolutionary clustering · Rating prediction · Recommendation

Introduction

The development of the internet and e-commerce makes our life more convenient as billions of required products are searchable online. Meanwhile, we must face the problem of information overload in daily life. Under the circumstances, it is much harder for us to dig out relevant object that we really want than ever before [1]. Many researchers have done plenty of research on the recommendation system, making progress about this particular issue. But data sparseness has always been an important reason for the recommendation low accuracy. To make full use of existing information, researchers have proposed more and more excellent algo-

rithms [2], such as neighborhood-based CF(Collaborative Filtering) and model-based CF.

Neighborhood-based CF algorithms are further classified into two categories: user-based CF [3] and item-based CF [4]. And the basic principle of them is interlinked. For instance, user-based CF considers two users to be similar when their neighbors are similar. Obviously, selection of nearest neighbor is significant. Correspondingly, choosing the appropriate similarity will be helpful for the improvement of the recommended accuracy and the applicability to the recommended algorithm [5]. The researcher takes into account the score information that best reflects the user's preferences. They came up with similarity measures such as cosine similarity [6], pearson correlation coefficient [7], adjusted cosine [8]. And others put forward such as Salton similarity [6] and Jaccard similarity [9], which is taken into account the number of items of self and their neighbors. A vertex similarity index CosRA is proposed, which combines both advantages of cosine index and resource-allocation index [1]. This fusion enhances the overall performance of personalized recommendation.

Model-based CF algorithms learn a model from the training data and subsequently, the model is utilized for rec-

✉ Jianrui Chen
jianrui_chen@snnu.edu.cn
Chunxia Zhao
zhaochunxia911@163.com

¹ School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

² College of Science, Inner Mongolia University of Technology, Hohhot 010051, China

ommendations [10], such as matrix decomposition model [11–13], clustering model [14–16]. In Ref. [11] presented a collaborative filtering recommendation algorithm based on SVD smoothing. Their approach predicts item ratings that users have not rated by the employ of SVD technology, and then uses Pearson correlation similarity measurement to find the target users neighbors, lastly produces the recommendations, which can alleviate the sparsity problems of the user item rating dataset. Kindly rephrase the sentence “Their approach predicts item” Superiority of CF algorithms with clustering techniques relieve the impact of data sparseness and cold start problems, which have been verified by many research experiments [17,18]. Recently, many scholars have also proposed dynamic evolutionary clustering algorithms [19–21], compared to the classical Kmeans algorithm, which not only reduced time-consuming and complexity, but also have indeterminate classification category. Liao et al. [22] have presented an approach which applied the user-product rating matrix without the necessity of collecting extra attributes about customers and products to cluster, and clusters are formed automatically. Shang et al. proposed a novel fuzzy double trace norm minimization method for recommender systems with faster running time [23]. Recently, we have presented two novel recommender algorithms based on heterogenous network model [24] and filled matrix [25]. They combined dynamic clustering with traditional collaborative filtering to gain better recommendation results.

Based on the above work, a novel recommended algorithm is proposed in this paper. Main motivations are given as follows. Firstly, to reduce complexity and time-consuming, the following two steps are executed on the score matrix: normalizing and reducing the dimension. Then, clustering principle is constructed and a novel dynamic evolutionary clustering model is proposed to imitate the changes of different nodes in network model. Secondly, to better grasp the highest similar interest neighbors, a new similarity index which combines advantage of degrees with potential information of nodes. Thirdly, ratings are predicted in each group based on user correlation.

The rest of the paper is structured as follows. Section “Evolutionary clustering algorithm” gives the evolutionary clustering model. In section “Similarity indices”, we review existing similarity indices and put forward a new similarity index. Full description of our algorithm is presented in section “The UCEC&CF algorithm flow”. Section “Experiment and studies” shows experimental results and comparisons with existing algorithms, and presents advantages of our algorithm. Section “Conclusion” concludes the paper.

Evolutionary clustering algorithm

Many complex social, biological, and information systems can be described by networks, where nodes indicate individuals, and links represent the relationships or interactions between nodes [26]. And the study of link prediction has become a common focus of science. A link between two nodes can be found by the topology information of network, or the feature vectors of nodes. In this paper, the network $G(V, E)$ is constructed, where V is the set of nodes containing M users and N_1 items, and E is the set of scores of items from users. Firstly, clustering method is presented to gather similar interest users into the same groups.

Data pre-processing

As we all know, clustering is roughly divided into three steps: deal with data at first, then choose the appropriate distance function to measure the similarity between objects, and at last obtain the groups. Naturally, the most important step of clustering is to choose optimal clustering principle. Ba et al. [12] proposed a method to cluster all users by calculating the user characteristic value with the attributes of users. But this method requires much computational cost. So, in this paper, to reduce the complexity, only the score matrix is adopted to gather users into groups.

To effectively apply scores information, Ref. [22] normalized the ratings of user u_i according to Eq. (1). By this, the influences in habit of people giving ratings will be reduced. So, we take the same approach, original score matrix $R = (r_{ij})_{M \times N_1}$ will be processed by (1).

$$Q_i = \sum_{j=1}^{m_i} r_{ij}, \quad i = 1, \dots, M, \quad (1)$$

$$\tilde{r}_{ij} = \frac{r_{ij}}{Q_i}, \quad 1 \leq j \leq m_i.$$

Here, r_{ij} denotes the user i rating of the item j , m_i denotes the number of items from user u_i .

At the same time, to gain better recommendation results, items with fewer scores are removed. And in this way, the sparse scoring matrix will become relatively dense. Here, the number of items is changed from N_1 to N , $N_1 > N$. By doing this, the new relationship between users and items is generated, which is denoted as $\tilde{R} = (\tilde{r}_{ij})_{M \times N}$. The proportion of removing is denoted as α and the influence of α on the results is later shown in section “Experiment and studies”. Adjacent matrix A of network is constructed similar to Ref. [24]. In this paper, adjacency matrix is given as follows:

$$A = \begin{bmatrix} \mathbf{0} & \tilde{R}_{M \times N} \\ (\tilde{R}^T)_{N \times M} & \mathbf{0} \end{bmatrix}.$$

Here, A is an $N + M$ dimension square matrix. $\mathbf{0}$ is zero matrix.

Evolutionary clustering algorithm

Wu et al. [27] have presented generalized Kuramoto model to identify the community structure for positive networks. The model is defined as follows:

$$\begin{aligned} \dot{\theta}_i = & \omega_i + \frac{K_p}{N} \sum_{j=1}^N a_{ij} \sin(\theta_j - \theta_i) \\ & + \frac{K_n}{N} \sum_{j=1}^N (1 - a_{ij}) \sin(\theta_j - \theta_i). \end{aligned} \quad (2)$$

Here, θ_i is the state of the i -th node, ω_i denotes the initial state of node i , and it is randomly generated from the interval $[0, 2\pi]$. N refers to the number of nodes in the network. If the i -th node is connected with the j -th node, $a_{ij} = 1$, otherwise, $a_{ij} = 0$. $K_p > 0$ is to make the states of two connected oscillators in the network synchronize and $K_n < 0$ is to make the states of two unconnected oscillators in the network evolve far away.

Inspired by this, the model is used in heterogeneous network, and the user is divided into different communities by the relationship between the user and the item. The evolutionary clustering rule is designed as: if the user is rating the item with higher score, they are regarded as in the same group.

In this paper, the dynamic evolutionary clustering model is defined as:

$$\begin{aligned} \dot{\theta}_i = & \omega_i + \frac{K_p}{N + M} \sum_{j \in NL_i} \sin(\theta_j - \theta_i) \\ & + \frac{K_n}{N + M} \sum_{j \notin NL_i} \sin(\theta_j - \theta_i). \end{aligned} \quad (3)$$

Here,

$$NL_i = \{j | A_{ij} < \delta\},$$

where $N + M$ refers to the number of users and items in the network. The coupling parameters $K_p > 0$, $K_n < 0$ are to make the nodes with higher scores in the network evolve together and to make the nodes with lower scores in the network evolve far away, respectively. δ denotes the critical value of the high and low scores, the size is the median of the nonzero element in the matrix A . With the random generation of the initial values of the nodes, the iteration begins. Ref. [27] has verified the convergence of the nodes evolution. After a certain number of iterations, the states of nodes would be stable. Nodes with state values in nearby

Table 1 Similarity indices

Common neighbors (CN)	$S_{uv} = \Gamma(u) \cap \Gamma(v) $
Hub prompted index (HPI)	$S_{uv} = \frac{ \Gamma(u) \cap \Gamma(v) }{\min\{k(u), k(v)\}}$
Hub depressed index (HDI)	$S_{uv} = \frac{ \Gamma(u) \cap \Gamma(v) }{\max\{k(u), k(v)\}}$
Salton index (SI)	$S_{uv} = \frac{ \Gamma(u) \cap \Gamma(v) }{\sqrt{k(u) \times k(v)}}$
Jaccard index (JI)	$S_{uv} = \frac{ \Gamma(u) \cap \Gamma(v) }{ \Gamma(u) \cup \Gamma(v) }$
Cosine (COS)	$S_{uv} = \frac{\sum_{i=1}^k r_{ui} r_{vi}}{\sqrt{\sum_{i=1}^k r_{ui}^2} \sqrt{\sum_{i=1}^k r_{vi}^2}}$
Pearson (COR)	$S_{uv} = \frac{\sum_{i \in \Gamma(uv)} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \Gamma(uv)} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in \Gamma(uv)} (r_{vi} - \bar{r}_v)^2}}$
Adjusted cosine (ADJ)	$S_{uv} = \frac{\sum_{i \in \Gamma(uv)} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \Gamma(u)} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in \Gamma(v)} (r_{vi} - \bar{r}_v)^2}}$

range ($|\theta_i - \theta_j| < \varepsilon$) would get closer and be divided into the same group. At the same time, nodes with large different state values would be divided into the different groups. Higher ε means more nodes are divided into the same cluster and lower ε means fewer nodes are divided into the same cluster. ε is to confirm which nodes would be divided into the same cluster.

Similarity Indices

Local similarity indices

Most of the existing similarity indices are classified into two categories: local and global similarity indices. The similarity indices [28] are defined in Table 1, where $\Gamma(u)$ denotes the set of neighbors of u , $\Gamma(uv)$ denotes the set of common neighbors of u and v , $k(u)$ is the degree of user u , r_{ui} denotes user u rating of item i , $\bar{r}_u$ is historical average score of u .

User correlation indices

In real life, people have different expectations for the products, those who have lower expectations tend to give a high lever of satisfaction feedback, while the others seem like to be much pickier. For instance, user A who has the lower expectations always gives 4 or 5 stars for the product feedback; user B who belongs to the other side only gives 4 stars feedback which could be possibly the high score he or she marked in the history. So when both of them make their comments like 4 stars on product, it actually means different, dose not it? So in this paper, we evaluate the user's preference when they do the feedback.

On the other hand, different feedbacks from different users can be a good evaluation about the product, for example, user

A gives 2 stars while user B gives 5 stars on a same product. Why they originally buy this product? The reason maybe is that they both like its appearance. But why they mark feedback are quite different? The reason maybe is practicability, price, even after-sales service. Thereby, it is more difficult to figure out why exactly users give different evaluations about same product. So, the paper takes the general evaluation into account; meanwhile, the feedback needs to focus on the same product.

Based on the above reasons, the calculation method of user correlation is given as follows :

$$SIM_{uv} = \frac{|B| + |C|}{\sqrt{|N(u)||N(v)|}}. \quad (4)$$

Here, $B = \{i | r_{ui} \geq \beta \text{ and } r_{vi} \geq \gamma\}$, $C = \{i | r_{ui} < \beta \text{ and } r_{vi} < \gamma\}$. $N(u)$ denotes items set of user u has evaluated in original score matrix R . $N(v)$ denotes items set of user v has evaluated in original score matrix R . β and γ are two parameters; in our experiment they represent historical average rating scores, respectively. It can be found that the numerator of SIM_{uv} considers potential information of users and the denominator is about degrees of users, which can obtain more similar information between users u and v .

In our algorithm, we do not adopt the normalization in similarity indices. Similarity calculation is applied to differentiate the influences of neighbors.

The UCEC&CF Algorithm Flow

In this paper, Collaborative Filtering Recommendation Algorithm Based on User Correlation and Evolutionary Clustering (denoted as UCEC&CF) algorithm is proposed. The procedure of algorithm is given as Fig. 1.

Step 1 Generate the adjacency matrix. Input score matrix of users and items, normalize the matrix by (1), remove the items with fewer scores, and generate the adjacency matrix as A .

Step 2 Generate the initial states of nodes. States of nodes are randomly generated from the interval $[0, 2\pi]$.

Step 3 Update states of nodes. Given two values for coupling strength parameters K_p and K_n and iteration threshold t , states are updated by (2) and the states are stable at some certain values. In all the simulations of this paper, $K_p = 10$, $K_n = -0.01$, $t = 100$.

Step 4 Obtain the group structure. Nodes would be divided into a group if the state values meets the condition $|\theta_i - \theta_j| < \varepsilon$. In all the simulations of this paper, $\varepsilon = 0.001$.

Step 5 Calculate the user correlation matrix. User correlation matrix SIM is calculated by (3) in its group based on the original score matrix.

Fig. 1 The UCEC&CF Algorithm Flow

Step 6 Predict the ratings. For every rating of items to be tested from target users, the calculation method is according to formula (4).

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in U} SIM_{uv}(r_{vi} - \bar{r}_v)}{\sum_{v \in U} SIM_{uv}}. \quad (5)$$

Here, r_{ui} denotes the rating of item j from user i , $\bar{r}_u$ is historical average score of u . SIM_{uv} denotes the relationship between user u and v . U is neighbor set which contains the most similar k users with user u in same group according to our user correlation index.

Step 7 Top-N recommend. All the predict scores for the target user are sorted in descending order. Pick the first N items to generate a recommend list and give them to the target user.

Experiment and studies

All the algorithms are coded in Matlab 2013a; the configuration of the experimental platform is Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz, 64 GB memory. The operating system is Windows Server 2008 R2 Enterprise.

Dataset and evaluation metrics

We conducted experiments on movielens dataset (<http://grouplens.org/datasets/movi-elens/>). The dataset from Grouplens is the most popular dataset in the field of recommender systems. Our evaluation is restricted to 100 K and 1 M datasets; the former consists of 1682 movies, 943 users, and almost 100,000 known ratings in the scale from 1 to 5 and the latter contains 1,000,209 anonymous ratings of approxi-

Fig. 2 100 K dataset. Comparison results of DEHC-CF and UCEC&CF algorithms. **a** MAE comparing results, **b** RMSE comparing results

mately 3900 movies made by 6040 users. To reduce the error, we conducted fivefold cross-validation taking 80% of available ratings as train and the rest as test data.

In general, the evaluation metrics of recommended system are accuracy, but the accuracy of prediction and Top-N recommendation is different. In this paper, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are applied to verify the accuracy of prediction. The lower the MAE (or RMSE) is, the better the prediction ratings.

$$\text{MAE} = \frac{1}{n} \sum_{u=1}^n |r_{ui} - \hat{r}_{ui}|. \quad (6)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{u=1}^n |r_{ui} - \hat{r}_{ui}|^2}. \quad (7)$$

Here, n is the number of ratings to be predicted. r_{ui} and $\hat{r}_{ui}$ are the real and predicted scores, respectively.

We also evaluate our algorithm based on the precision of Top-N recommendation measured in terms of F in varying lengths of recommendation list. The higher the F is, the better the recommender system algorithm.

$$F = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}, \quad (8)$$

$$\text{recall} = \frac{\sum_u |R_u \cap T_u|}{\sum_u |T_u|}, \quad (9)$$

$$\text{precision} = \frac{\sum_u |R_u \cap T_u|}{\sum_u |R_u|}. \quad (10)$$

R_u denotes the set of users in the real high ratings (i.e., ≥ 3 in movielens dataset) and T_u denotes users set in predicted high ratings in the test set.

Table 2 Comparisons of DEHC-CF and UCEC&CF algorithms on F (%) with 100 K dataset

Algorithm	No. of recommendations					
	2	4	6	8	10	12
DEHC-CF	15.54	27.43	36.59	43.73	49.05	54.02
UCEC&CF	15.57	27.43	36.59	43.80	49.60	54.15

The bolditalic values are corresponding optimal values

Experimental results

In this subsection, we apply the multiview to test the accuracy of our algorithm UCEC&CF, and compare the accuracy of the prediction and Top-N recommendation with other algorithms.

1. Comparing results of processing score matrix. To test the impact of normalization and dimension reduction on the score information, we compare our algorithm with DEHC-CF [24]. In their paper, the authors established heterogeneous network, and generated the groups based on score information. Parameters in clustering principle for five cross-validation sets are $\delta_1 = 0.0070$, $\delta_2 = 0.0072$, $\delta_3 = 0.0071$, $\delta_4 = 0.0071$, $\delta_5 = 0.0069$, and $\alpha = 50\%$ in the following. The parameters of our clustering algorithm are $K_p=10$, $K_n = -0.01$.

It can be observed from Fig. 2 that our method is slightly lower than DEHC-CF in some cases. Especially, when the number of neighbors is 60, our experiment is more effective. Mean value of MAE of DEHC-CF is 0.7491 and mean value of RMSE is 0.9572. Mean values of MAE and RMSE of UCEC&CF are 0.7482 and 0.9560, respectively. It can be shown that score matrix processing is meaningful to improve recommender precision. Table 2 shows the F measurements along with different number of recommendations. For all

Fig. 3 100 K dataset. Comparison results of different clustering methods. **a** MAE metrics, **b** RMSE metrics

Table 3 Comparison results of DTD, DPSO, Kmeans and UCEC&CF algorithm on F (%) with 100 K dataset

Algorithm	No. of recommendations					
	2	4	6	8	10	12
DTD	15.21	26.91	35.95	43.09	48.84	53.49
DPSO	14.48	25.80	34.63	41.55	47.20	51.82
Kmeans	15.50	27.39	36.58	43.83	49.65	54.30
UCEC&CF	15.57	27.43	36.59	43.80	49.60	54.15

The bolditalic values are corresponding optimal values

algorithms, F improves with increasing length of recommendation list. Our algorithm performs better and recommends more satisfactory items for user when the number of recommendations becomes larger.

2. Comparing results of different clustering methods. To test whether the clustering method of our algorithm is efficient, three methods are compared. They are DTD in Ref. [19], DPSO in Ref. [29] and Kmeans. In their papers, iteration threshold of DTD is 0.1. The parameters of DPSO are as follows: particle swarm size is 100, the number of interactions is 4, learning factor is 1.494, max and min inertia weights are 0.9 and 0.4, respectively. The cluster number of Kmeans is given three cases and individuals in network are divided into 3 clusters, 6 clusters and 10 clusters. The test results are shown in Fig. 3 and Table 3. All comparing results do not apply similarity indices normalization.

Figure 3 presents MAE and RMSE values along different setting of neighborhood size. It can be observed that our presented method is more effective than the other clustering algorithms. The result of DPSO is that each group contains only one user; so, the prediction accuracy is independent of the number of neighbors. Mean values of MAE of

UCEC&CF, DTD, DPSO and Kmeans are 0.7482, 0.7882, 0.8362 and 0.7557, respectively. Furthermore, mean values of RMSE for UCEC&CF, DTD, DPSO and Kmeans are 0.9560, 1.0070, 1.0437 and 0.9668. Besides, our proposed method is more efficient than Kmeans algorithm, as our method does not require the predefined number of clusters.

Table 3 shows the F values for all the algorithms. UCEC&CF is more efficient than DTD and DPSO in view of F values in all cases. In terms of top- N recommendation accuracy, our algorithm slightly inferior to Kmeans algorithm with increasing length of recommendation list. F values of DPSO are worst.

3. Comparing results of different similar indices. As follows, our user correlation results are compared with other eight similarity indices (see Table 1), and the results are shown in Fig. 4 and Table 4. All comparing results do not apply similarity indices normalization.

It can be seen that our method carries the best performance from Fig. 4. Some compared similarity indices only consider the degree of the nodes and others are computed from users' ratings to same items. Our algorithm obtains higher recommendation results, and the reason is that our presented user correlation considers degree of nodes and score information.

To further check the performance of the proposed algorithm, Table 4 gives recommended accuracy between algorithms in different number of recommendations. It can be seen that our method gains better performance than the others.

4. Comparing results of different α . In the algorithm, part of the data is removed to reduce the impact of sparseness. α is the proportion of removal items. Experimenting with

Fig. 4 100 K dataset. Comparing results of UCEC&CF and existing similarity indices. **a** MAE metrics, **b** RMSE metrics

Table 4 The comparison results between UCEC&CF and existing similarity indices on $F(\%)$ with 100 K dataset

Algorithm	No. of recommendations					
	2	4	6	8	10	12
CN	15.41	27.24	36.41	43.50	49.39	53.88
HPI	15.46	27.26	36.35	43.56	49.37	54.03
HDI	15.46	27.37	36.47	43.58	49.40	54.07
SI	15.50	27.36	36.51	43.69	49.40	54.13
JI	15.46	27.41	36.57	43.64	49.46	54.03
COS	15.43	27.36	36.41	43.54	49.41	54.03
COR	15.41	27.28	36.44	43.67	49.40	54.12
ADJ	15.48	27.27	36.49	43.64	49.48	54.13
UCEC&CF	15.57	27.43	36.59	43.80	49.60	54.15

The bolditalic values are corresponding optimal values

Fig. 5 100 K dataset. Influence results of α on the recommendation performance

100 K dataset, the results are shown in Fig. 5; parameter α has slight influence on the results. In the simulations of this

part, the number of neighbors is 30 and the number of recommendations is 10. Range of MAE is [0.7432, 0.7457], range of RMSE is [0.9506, 0.9534] and range of F is [0.4948, 0.4964].

5. The comparison results between several existing recommendation algorithms. Collaborative filtering and matrix decomposition are representative of existing recommendation algorithms. To test the effectiveness of our overall algorithm, our UCEC&CF algorithm is compared with DEHC-CF and DTNM [23] on two datasets: MovieLens100K, MovieLens1M. DEHC-CF is a novel heterogeneous evolutionary clustering algorithm based on user collaborative filtering, the coupling strength $K1 = 20$ and $K2 = -20$. DTNM is a novel fuzzy double trace norm minimization method for recommender systems, the values of rank are integer between 5 and 15, and it has little effect on the result. DTNM is compared with rank = 5. The comparing results on different data sets are illustrated in Fig. 6 and Table 5.

Figure 6a, b gives the MAE and RMSE values, respectively, for 100 K dataset. MAE and RMSE values for the 1 M dataset are given in Fig. 6c, d. From overall view, the improvement in MAE and RMSE is more pronounced for the 1 M dataset than the 100 K one as the former has higher explicit rating data sparsity. The result of DTNM is independent of the number of neighbors, which is because that it is a rank-dependent matrix decomposition algorithm. As expected, our UCEC&CF algorithm performs the best amongst all compared algorithms, which means ours work gains better performance than DEHC-CF and DTNM. Our work broadly includes: building heterogeneous networks, setting cluster principles, defining user correlation, etc.

Table 5 gives a description of UCEC&CF, DEHC-CF and DTNM when items are recommended to users. The

Fig. 6 MAE and RMSE of different methods on two datasets

Table 5 Comparison results of different algorithms on F (%)

MovieLens	Algorithm	No. of recommendations					
		2	4	6	8	10	12
100 K	DEHC-CF	15.34	27.16	36.18	43.39	49.09	53.76
	DTNM	15.46	27.29	36.38	43.58	49.32	54.01
	UCEC&CF	15.57	27.43	36.59	43.80	49.60	54.15
1 M	DEHC-CF	12.69	23.53	32.17	39.12	44.79	49.53
	DTNM	12.55	23.19	31.68	38.49	44.04	48.71
	UCEC&CF	12.82	23.74	32.42	39.40	45.11	49.89

The bolditalic values are corresponding optimal values

experiment results demonstrate that our method gains better performance than the others compared algorithms with different datasets.

In a word, in section “Experiment and Studies”, two real datasets with different size are considered to verify the efficiency of our proposed algorithm UCEC&CF. Five diversity experiments are tested and UCEC&CF gains excellent recommendation results.

Conclusion

A new CF model based on user correlation and evolutionary clustering is proposed in this paper. The performance of our proposed method is verified by comparative experiments from different aspects. On the one hand, only normalized and reduced dimension score matrix is applied to generate clustering principle. An improved dynamic evolutionary model is constructed to imitate the constantly changing states values in the network. Similar users are divided into the same group. Besides, user correlation is proposed to measure the distance between users by combining the users satisfaction and the potential score information. In each group, a novel user correlation measurement is adopted to find the highest similar neighbors with target user. Finally, user-based collaborative filtering is adopted in each group. Extensive experiments demonstrate our proposed method gains better recommendation performance than the other compared algorithms. But the work is still much room for improvement, we want to join some other information of network into the design for further performance enhancement.

Acknowledgements This research was supported through National Natural Science Foundation of China (No. 71561020, 61503203, 61702317, 61771297); Fundamental Research Funds for the Central Universities (No. GK201802013).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Chen LJ, Zhang ZK, Liu JH, Gao J, Zhou T (2017) A vertex similarity index for better personalized recommendation. *Phys. A* 466:607–615
- Singh R, Patra BK, Adhikari B (2015) A complex network approach for collaborative recommendation. *arXiv preprint arXiv:1510.00585*
- Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) GroupLens: an open architecture for collaborative filtering of netnews. In: *Proceedings of the ACM conference on Computer supported cooperative work*. ACM, pp 175–186
- Sarwar BM, Karypis G, Konstan JA, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. *Www* 1:285–295
- Leicht EA, Holme P, Newman ME (2006) Vertex similarity in networks. *Phys. Rev. E* 73(2):026120
- Salton G, McGill MJ (1983) *Introduction to modern information retrieval*. McGraw-Hill, New York
- Breese JS, Heckerman D, Kadie C (2013) Empirical analysis of predictive algorithms for collaborative filtering. *ACM* 7(7):43–52
- Shardanand U, Maes P (1995) Social information filtering: algorithms for automating word of mouth. In *Chi* 95:210–217
- Salton G (1989) *Automatic text processing: the transformation, analysis*. Addison-Wesley, Reading
- Pazzani MJ, Billsus D (2007) *Content-based recommendation systems*. Springer-Verlag, New York, pp 325–341
- Ren YB, Gong SJ (2009) A collaborative filtering recommendation algorithm based on SVD smoothing. In: *Third International Symposium on Intelligent Information Technology Application IEEE*, pp 530–532
- Ba Q, Li X, Bai Z (2013) Clustering collaborative filtering recommendation system based on SVD algorithm. In: *IEEE International Conference on Software Engineering and Service Science*, pp 963–967
- Zhang Y, Chen M, Huang D, Wu D, Li Y (2016) Idoctor: personalized and professionalized medical recommendations based on hybrid matrix factorization. *Future Gener Comput Syst* 66:30–35
- Sarwar BM, Karypis G, Konstan J (2002) Recommender systems for large-scale E-commerce: scalable. *Confer Comput Inform Technol* 1:291–324
- Haldar NAH, Khan FA, Ali A, Abbas H (2017) Arrhythmia classification using mahalanobis distance based improved fuzzy c-means clustering for mobile health monitoring systems. *Neurocomputing* 220:221–235
- Wang Z, Wen Q, Sun Y, Zhang H (2012) A Fault Detection Scheme Based on Self-Clustering Nodes Sets for Wireless Sensor Networks. In: *IEEE 12th International Conference on Computer and Information Technology*, pp 921–925
- Birtolo C, Ronca D, Aurilio G (2012) Trust-aware clustering collaborative filtering: identification of relevant items. Springer, Berlin Heidelberg, pp 374–384
- Huang C, Yin J (2010) Effective association clusters filtering to cold-start recommendations. In: *Seventh international conference on fuzzy systems and knowledge discovery, IEEE*, vol 5, pp 2461–2464
- Wu J, Jiao Y (2014) Clustering dynamics of complex discrete-time networks and its application in community detection. *Chaos* 24(3):268–4
- Wu J, Zhang L, Li Y, Jiao Y (2016) Partition signed social networks via clustering dynamics. *Phys A* 443:568–582
- Chen J, Wang H, Wang L, Liu W (2016) A dynamic evolutionary clustering perspective: community detection in signed networks by reconstructing neighbor sets. *Phys A* 447:482–492
- Liao CL, Lee SJ (2016) A clustering based approach to improving the efficiency of collaborative filtering recommendation. *Electr Commer Res Appl* 18:1–9
- Shang F, Liu Y, Cheng J, Yan D (2017) Fuzzy double trace norm minimization for recommendation systems. *IEEE Trans Fuzzy Syst* 99:1–1
- Chen J, Uljii, Wang H (2018) Collaborative rating prediction based on dynamic evolutionary heterogeneous clustering. *Swarm Evol Comput* 394–399
- Liji U, Chai Y, Chen J (2018) Improved personalized recommendation based on user attributes clustering and score matrix filling. *Comput Stand Interfaces* 57:59–67
- Ding J, Jiao L, Wu J (2015) Prediction of missing links based on multi-resolution community division. *Phys A* 417:76–85

27. Wu J, Jiao L, Jin C, Liu F, Gong M (2012) Overlapping community detection via network dynamics. *Phys Rev E* 85(2):016115
28. Lu L, Zhou T (2011) Link prediction in complex networks: a survey. *Phys A* 390(6):1150–1170
29. Pizzuti C (2008) GA-Net: a genetic algorithm for community detection in social networks. Springer-Verlag, New York, pp 1081–1090

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.