Hybrid recommender system with core users selection

Chenxia Jin (jinchenxia2005@126.com)
Hebei University of Science and Technology https://orcid.org/0000-0002-5741-9473
Jusheng Mi
Hebei Normal University
Fachao Li
Hebei University of Science and Technology
Jiahuan Zhang
Hebei Normal University

Research Article

Keywords: Hybrid Recommender system, Matrix factorization, Collaborative filtering, Genetic algorithms

Posted Date: April 27th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1552154/v1

Hybrid recommender system with core users selection

Chenxia Jin[*,a,b], Jusheng Mi[a], Fachao Li[b], Jiahuan Zhang[a]

[a]School of Mathematical Sciences, Hebei Normal University, Shijiazhuang, China;

[b]School of Economics and Management, Hebei University of Science and Technology, Shijiazhuang, China;

[*]jinchenxia2005@126.com, mijsh@263.net, lifachao@tsinghua.org.cn

Corresponding author: jinchenxia2005@126.com

Tel. 86-031181668359; ORCID: 0000-0002-5741-9473

Abstract Recommender system plays an increasingly important role in identifying the individual's preference and accordingly makes a personalized recommendation. Matrix factorization is currently the most popular model-based collaborative filtering (CF) method that achieves high recommendation accuracy. However, similarity computation hinders the development of CF-based recommendation systems. Preference obtained only depends on the explicit rating without considering the implicit content feature, which is the root cause of preference bias. In this paper, the content feature of items described by fuzzy sets is integrated into the similarity computation, which helps to improve the accuracy of user preference modelling. The importance of a user is then defined according to preferences, which serves as a baseline standards of the core users selection. Furthermore, core users based matrix factorization model (CU-FHR) is established, then genetic algorithm is used to predict the missing rating on items. Finally, MovieLens is used to test the performance of our proposed method. Experiments show CU-FHR achieves better accuracy in prediction compared with the other recommendation methods.

Keywords Hybrid Recommender system Matrix factorization Collaborative filtering Genetic algorithms

1. Introduction

The internet/information technology leads to information overload, then results in difficulty of user finding relevant information. Therefore it brings a big challenge for users to facilitate decision-making. Recommender systems (RSs) are the very tools to address the information overload problem through predicting user's preferences on a given item based on assorted types of information. The basic idea is that similar users like similar items. Recommender systems have been widely applied in many web-based domains such as e-commerce, e-government and e-learning and so on [1].

One of the most popular methods for RSs is Collaborative Filtering (CF), which recommends item i to user u if the users similar to u have rated i high or i is similar to the items rated high by u. This method is widely recognized as a prominent recommendation technique if the collected historical rating data are available. However, the users' prior ratings on items are very sparse in practice which occurs because most users tend to only rate a small fraction of the items they prefer. In the case of rating information scarcity, CF often suffers from data sparsity problems that are particularly severe and challenging [2-5]. Unfortunately, CF may lead to poor recommendation because the prediction accuracy of the calculated unseen preferences is considerably low [6]. This limitation will restrict the applications of CF in RSs to a certain degree. To address this issue in CF, various researches have been done to improve the performance of recommendations in sparse situations. Some of them filled the miss value to resolve the sparsity problem. Among them, default voting value and imputation are two favored methods to overcome the drawbacks of CF [2,7]. But the filling results do not consider the variance of the ratings. Either the values are often away from the reality [2] or they distort the data distribution [7]. According to famous Pareto's principle, just a few key ideas are critical to decision-making. Similarly, in a recommender system, a few key users will have a great effect on recommendation quality. Motivated by this idea, we emphasized the role of core users in recommendation. Based on this assumption, this paper produces the clusters of users according to the similarity using KNN [8], then from which the core user are extracted to form a small clique. Accordingly, the partial ratings matrix rather than the original one can be used as a basic recommendation foundation. This process, at a maximal degree, can possibly reduce the sparsity and lower the computation complexity in recommendation generation. Here, we should pay more attention to two major issues. 1) Pearson’s correlation and Cosine-based formula are often used to calculate the similarity. As we know, the two methods are dependent on the historical data on the items that have co-rated by both users under consideration. This limitation may lead to a fake similarity. For example, the similarity between two users is relatively low when there exist very fewer same items in the rating list; and even worse, the similarity cannot be computed if no same items exist. Thus, this paper improves the accuracy of Cosine’s similarity based on the content features of similar items. That is, if two users are inclined to like the items with similar characteristics, such as the genres, director, protagonist of a movie, then the two users technically have a higher similarity. 2) After the execution of KNN, each user is surrounded by fixed number of neighbors based on a given threshold. According to the frequency of each user emerging in all the groups, the presetting number of core users with the highest occurrence can be determined.

CF can be commonly divided into two groups: memory-based CF and model-based CF [1]. Memory-based CF can either be user-based and item-based methods. User-based CF method generates recommendations according to the preferences of neighbor users (the most similar users), while item-based method recommends items, the most similar items to the past preferences, to a specific user. Model-based CF method is able to make recommendations through creating some models, such as matrix factorization [9], probabilistic model [10] and Bayesian classifier [11]. Matrix factorization (MF) is probably the most successful realization and has been widely applied in many commercial fields [2]. In matrix factorization, the original user-item rating matrix is decomposed into two low-dimension matrices: user-feature matrix and item-feature matrix, in which the users and items are related to their latent features [12]. As a matter of fact, MF-based recommendation task is to predict the missing values in the rating matrix through the production of the factorized user-specific matrix and item-specific matrix. It can be mathematically described as a convex optimization problem [13]. The common techniques to find the optimal solutions include stochastic gradient descent [13, 14], artificial immune system approach [15] and genetic algorithm [3, 16, 17] and so on. At present, genetic algorithm has been successfully applied to MF for further recommendation due to the better global optimization capability, wider adaptive ability and easier operability [18]. It searches the optimal solutions through repeatedly executing the selection, crossover and mutation operation. For example, Navgaran [16] gave the MF-based recommendation by combining with genetic algorithm considering the whole users and items. Kilani [17] improved the work of Navgaran [16], the matrix factorization is done on small parts of the two low-dimension matrices, that are related to the active user. All these efforts can speed the recommendation process and lead to a high quality recommendation. However, the work starts from a specific user, only considering his (her) own neighbors, and emphasizes the local features of a recommender system without taking the integrity into account. To make up this deficiency, we establish a core users-based matrix factorization model, reflecting the overall feature of a recommender system and reduce the sparsity.

Generally speaking, the items often contain some uncertain features/descriptions. Therefore, fuzzy techniques are suitable to model the uncertainties in items. For example, movies are often multi-genres and multi-actors, and the attributes values represented by fuzzy set can be more accurate [19]. Movies can be labeled as all kinds of genres. Such as, a movie can be stated as romantic, action, scary and animation and so on. Taking a mixture of genre content into consideration can improve the accuracy of content-based recommendation [20]. Thus the membership degree of a movie to each genre is a pre-step for comprehensively integrating all content information. This paper uses the determination method in [20]. Accordingly, we adopt fuzzy linguistic variables to represent the items features to fulfill the recommendation under uncertain environment.

The contributions of this paper can be summarized as follows:

1) An improved similarity computation method is proposed. It can deal with the fake similarity problem by considering the genres similarity of the both rated items, in which fuzzy sets are adopted to capture the uncertainty in rating and classification. So it can improve the accuracy of similarity in recommender system.

2) Core users selection method for recommender system is proposed. The users are firstly clustered by KNN technique according to similar preferences, and then the core users are selected by the importance of users. This strategy can effectively help to reduce the computations complexity.

3) A new hybrid recommendation model with core users (CU-FHR) is proposed. CU-FHR merges content features of items into CF and is also a hybrid of matrix factorization and genetic algorithm. It can favorably alleviate the data sparsity problem occurred in CF. Therefore CU-FHR can improve the recommendation accuracy.

The rest of this paper is organized as follows. We review some work on fuzzy set theories, matrix factorization technique and related research background in Section 2. Section 3 gives the computation of user similarity considering both rating and content-based item similarity. Section 4 presents the framework of CU-FHR approach and expatiates the design steps. The experiments and results analysis are demonstrated in Section 5.The conclusion and further study are presented in Section 6.

2 Related works

2.1 Fuzzy sets techniques

Fuzzy set theory, firstly proposed by Zadeh in 1965 [21], has been applied to handle various uncertainties in many practical fields. A fuzzy set A in a domain space X is characterized by a membership function $\mu_A(x): x \in X \to [0,1]$. $\mu_A(x)$ can be termed as a membership degree of x belonging to A. In the contexts with different concept presentation, $\mu_A(x)$ has different meanings. For example, if X denotes the pool of movies in MovieLens, A denotes a fuzzy set "liked", then $\mu_A(x) \in [0,1]$ denotes the degree of x being liked by users. Specially, fuzzy set A defined in discrete domain X is represented by the set of pairs of the elements $x \in X$ and the corresponding membership degree $A = \{(x, \mu_A(x)) / x \in X, \mu_A(x) \in [0,1]\}$. Sometimes it is difficult for users to give an exact real value to express an opinion inclination. Linguistic variable, whose values are words and sentences, can be a more realistic option to represent imprecise assessments [22, 23]. The linguistic variable with qualitative description is more in line with the human thinking patterns. At the same time, the fact the linguistic variable are treated as fuzzy sets makes the computation more convenient.

2.2 Recommender systems

Recommender systems help users to discover quickly and easily the items, such as products and services, that may interest them [23, 24] . Recommender systems use the interaction data between users and items to automatically predict and identify the user preferences so as to make recommendations. Generally, recommender methods are roughly categorized into four types: content-based (CB) [25], collaborative filtering-based (CF) [2], knowledge-based (KB) [26], hybrid recommender systems [17, 27, 28]. Among them, hybrid recommenders have gained much attention in recent years. It is a combination of two or more of recommendation approaches to achieve better performance of a recommender system [23, 29, 30]. Some work only relied on the rating data, and some tried machine learning and data mining methods such as neural network, genetic algorithms. For example, Kilani [17] established a genetic algorithm-based hybrid recommendation system of neighborhood-based and matrix factorization-based approach. Zhang [23] combined user-based and item-based collaborative filtering methods with fuzzy set techniques and applied it to mobile product and service recommendation. Anwaar [25] introduced a hybrid framework on the basis of both CF and CB approaches exploiting the semantic of the contents as well as the user preferences to increase the performance of recommender systems. Burke [29] proposed a classification of hybrid recommender systems and listed seven basic hybridization mechanisms. Kermany [30] incorporated demographic information and an item-based ontological semantic filtering approach into fuzzy multi-criteria collaborative filtering for movie recommendation. Viktoratos [26] hybridized a collaborative system and a knowledge-based system to solve the cold start problem. Palomares [31] combined collaborative filtering techniques with fuzzy decision-making approaches by conflating preference information with user-profile data in the recommendation process. The aforementioned references significantly suggest that the combination of CF recommendation approach and other techniques may achieve good performance in specific domains recommendations. It also has been proven that a hybrid recommender system of CF and other methods is the most popular approach for recommender systems [32].

As a model-based CF, matrix factorization method has gained great popularity by mapping both users and items to a joint latent factor space of low-dimensionality, such that user-item interactions are modeled as inner products in that space. Suppose an original user-item rating matrix $X \in \mathbf{R}^{n\times m}$ ($\mathbf{R}^{n\times m}$ represents the sets of matrix with n rows and m columns), then matrix factorization is to decompose X into two low-dimensional matrices: the user feature matrix $U \in \mathbf{R}^{n\times k}$ and the item feature matrix $V \in \mathbf{R}^{k\times m}$. k ($k = m, k = n$) is the number of latent factors and can be an adjusted parameter in experiment setting. Formally, X is denoted approximately by $\hat{X} = UV$, in which the ith row of U and the jth column of V represents the ith user and the jth item, respectively. The goal of matrix factorization is to find U, V such that $\hat{X} \approx X$.

In X, $x_{ij} \in \{1,2,3,4,5,\Delta\}$, "1-5" denotes the rating score of user i to item j; Δ denotes user i did not rate item j. The recommendation task is to predict the value of Δ in X, by which the top items with the highest rating will be recommended to a specific user. Let $err_{ij} = | x_{ij} - \hat{x}_{ij} |$ denote the difference between the real value and the predicted value of user i to item j, called error. In this paper we use the loss function for all users given by Kilani [17]. In general, matrix factorization can be converted into an optimization problem as follows:

$$\min l(X,U,V) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m} err_{ij} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}| x_{ij} - \hat{x}_{ij} | = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}| x_{ij} - U_i V_j |, \tag{1}$$

Where, $\hat{x}_{ij} = \sum_{p=1}^{k} u_{ip} v_{pj}$, n and m are the number of users and items.

Due to the extreme sparsity of rating matrix X, matrix factorization often suffers from over fitting problem. Regularization penalty is usually added to reduce the influence of the problem [33]. Therefore, the objective function to measure the total loss with regularization penalty can be

$$\min l(X,U,V) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}| x_{ij} - U_i V_j | + \beta(\| U_i \|^2 + \| V_j \|^2), \tag{2}$$

Where β is a parameter used to control the extent of regularization, $\| \cdot \|$ is Frobenius norm. For the above optimization problem (2), there are many techniques to determine U_i, V_j such as stochastic gradient descent [34, 35] and GAs [3,16, 17].

3. Users similarity based on rating and fuzzy content feature

One of the most important issues what we should mainly concern in recommender systems is to compute the similarity between users and between items. Until now, there are many methods on similarity calculating, among which the Pearson correlation and Cosine vector are two popular methods that have gained wide applications. However, the above similarity-based method highly depends on the rating matrix, not considering the similarity degree in contents (features, genres, attributes etc.) of items, which may lead to a big difference between the predicted value and practical value. Therefore, similarity on the co-rated items can't be neglected in recommendation scenarios. If two users have rated or experienced some items with high similarity, then it is commonly considered that the two users have the same preferences. This section firstly introduces the fuzzy set theories to compute the content-based similarity between items, then gives the users similarity based on similar co-rated items.

3.1 Items similarity using fuzzy sets

To handle the non-uniqueness of item features and improve the credibility of similarity, this study integrates fuzzy sets techniques with Cosine-based method into the computation of users similarity.

Let $I=\{I_1, I_2, \cdots, I_m\}$ be an items set, $A=\{A_1, A_2, \cdots, A_t\}$ be an features set of items, then for an item $I_j\,(j=1,2,\cdots,m)$, it can take multiple values $A_1, A_2, \cdots, A_L\,(L \le t)$. The multi-valued features in an item can be represented by a fuzzy set. The membership function of item I_j to value $A_k\,(k=1,2,\cdots,L)$, denoted by $\mu_{A_k}(I_j)$, can be interpreted as the degree of similarity of I_j to a prototype A_k of the item. We use the Gaussian-like method in [21] to determine the value of $\mu_{A_k}(I_j)$.

$$\mu_{A_k}(I_j) = p_k \Big/ 2^{\sqrt{\alpha |L_j| (p_k - 1)}} \quad , \tag{3}$$

Where $|L_j| = L$ is the number of features values of A associated with item $I_j\,(j=1,2,\cdots,m)$, $p_k\,(1 \le p_k \le L)$ is the rank position of value A_k, $\alpha > 1$ is a parameter that needs to be determined as a threshold to control the difference between consecutive values of A in $I_j\,(j=1,2,\cdots,m)$.

For an given item I_j characterized by a series of feature values $A_k\,(k=1,2,\cdots,L)$, we can get a vector as follows,

$$\{(A_1, \mu_{A_1}(I_j)), (A_2, \mu_{A_2}(I_j)), \cdots, (A_L, \mu_{A_L}(I_j))\} \quad . \tag{4}$$

For items I_j and I_i with the representations as equation (4), a cosine vector-based similarity measure between I_j and I_i is defined as

$$s(I_j, I_i) = \frac{\sum_k \mu_{A_k}(I_j)\mu_{A_k}(I_i)}{\sqrt{\sum_k (\mu_{A_k}(I_j))^2}\sqrt{\sum_k (\mu_{A_k}(I_i))^2}} \quad , \tag{5}$$

Remark 1 Compared with the Cosine vector-based similarity measure, Pearson correlation formula processes the variables using mean value method, which can help reduce the influence of the numerical difference of individual variables on the overall similarity. However, the advantage can only be shown in the case of real number rating. For example, the rating value ranges from 1 to 5. In this paper, we mainly use the fuzzy membership degree $\mu_{A_k}(I_j)$ of item I_j to value A_k, $k=1,2,\cdots,L$ as a basic factor to compute the similarity. As we all know, the value of $\mu_{A_k}(I_j)$ lies within the interval $[0,1]$ and a large proportion of the values are 0 (Please see Tables 1 and 4). Under this situation, the advantages of the Cosine vector-based similarity can be explained from two aspects: 1) The overall similarity value is not very low; 2) The computation is easy and the complexity is low.

Therefore, we will use the Cosine vector-based similarity in formula (5).

For two movies I_1 = {Castle in the Sky: Action/Adventure/Animation/Fantasy/Romance/ Family}, I_2 = {Copycat 1995: Crime/Mystery/Thriller/Drama}, the similarity of items to each feature $A_k\,(k=1,2,\cdots,10)$ with $\alpha=1.2$ can be computed as equation (5). The results are shown in Table 1.

Table 1 Fuzzy representation of item similarity in feature spaces

	Action	Adventure	Animation	Fantasy	Romance	Family	Crime	Mystery	Thriller	Drama
$A(I_1)$	1	0.3114	0.2162	0.1596	0.1212	0.0938	0	0	0	0
$A(I_2)$	0	0	0	0	0	0	1	0.4380	0.3503	0.2882
$s(I_1, I_2)$	0									

3.2 Users similarity based on rating and similar items

In this paper, the Cosine-based method is selected for measuring the similarities between the two users u_1 and u_2. In many real cases, two users possibly have very fewer co-rated items, thus, the Cosine-based similarity is unlikely to reflect the actual preference between two users. Especially, if the two users have no co-rated items, the similarity can not be computed through Cosine-based method. However, if two users have rated different items which are very similar in content and also given similar rating scores, then the two users have similar tastes. Under the guidance of the above principle, we therefore give the following similarity measure based on the equation (5):

$$s(u_1,u_2)=\begin{cases} ws_1(u_1,u_2)+(1-w)s_2(u_1,u_2), & S_{u_1}\cup S_{u_2}\neq\phi \\ 0, & S_{u_1}\cup S_{u_2}=\phi \end{cases}, \tag{6}$$

In equation (6), we give the following detailed descriptions:

1) S_{u_1} and S_{u_2} are the set of items that u_1 and u_2 have rated, respectively;

2) $s_1(u_1,u_2)=\dfrac{\sum_{I_k\in S_{u_1u_2}} r_{u_1,I_k} r_{u_2,I_k}}{\sqrt{\sum_{I_k\in S_{u_1u_2}}(r_{u_1,I_k})^2}\sqrt{\sum_{I_k\in S_{u_1u_2}}(r_{u_2,I_k})^2}}$, $S_{u_1u_2}=S_{u_1}\cap S_{u_2}$ is the set of items of both u_1 and u_2 have rated;

3) $s_2(u_1,u_2)=\dfrac{\sum_{(I_k,I'_k)\in S} r_{u_1,I_k} r_{u_2,I'_k}\, s(I_k,I'_k)}{\gamma * \sqrt{\sum_{(I_k,I'_k)\in S}(r_{u_1,I_k})^2}\sqrt{\sum_{(I_k,I'_k)\in S}(r_{u_2,I'_k})^2}}$, $r_{u_1,I_k}, r_{u_2,I'_k}$ are the rating score of user u_1 on item I_k, user u_2 on item I'_k, respectively, $S=S_{u_1}\cup S_{u_2}-S_{u_1u_2}$;

4) $s(I_k,I'_k)$ is the fuzzy set theoretic content-based similarity between I_k and I'_k;

5) $\gamma=|S_{u_1}-S_{u_1u_2}|*|S_{u_2}-S_{u_1u_2}|$ is a adjusting parameter which maps the value of the second part into interval [0, 1], $|\cdot|$ represents the number of elements in a matrix; Note: $s_2(u_1,u_2)=0$ for $\gamma=0$;

6) $w=|S_{u_1u_2}|/|S_{u_1}\cup S_{u_2}|$ denotes the weight which connotes the contribution to the similarity, with more attention on the importance of co-rating items with higher w.

Remark 2 From the equation (6), the first part $s_1(u_1,u_2)$ reflects the explicit similarity feature based on the co-rated item, while the second part $s_2(u_1,u_2)$ denotes the implicit similarity inferred by different items rated. $s(u_1, u_2)$ takes into account not only explicit rating of users on both experienced items, but also the implicit links according to the contents of items, hence it makes the value more practical and be more line with the principle "the higher similarity the rated items are with, the more similar the two users are". The similarity is used to cluster the users aiming to find the core users in recommendation systems.

To better illustrate our method, in the following we will outline a small-scale example as shown in Tables 2-3. The corresponding results are shown as Tables 4-6.

Table 2 The information on movies

Movie ID	Title	Genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy
6	Heat (1995)	Action|Crime|Thriller
7	Sabrina (1995)	Comedy|Romance

Table 3 The ratings of users on movies

user	I_1	I_2	I_3	I_4	I_5	I_6	I_7
u_1		3					
u_2			4				
u_3	4						
u_4						3	
u_5		3					
u_6	5		3				5

Table 4 The membership degree of movie to genres

	Adventure	Animation	Children	Comedy	Fantasy	Romance	Drama	Action	Crime	Thriller
I_1	1	0.3662	0.2718	0.2113	0.1676	0	0	0	0	0
I_2	1	0	0.5369	0	0.4671	0	0	0	0	0
I_3	0	0	0	1	0	0.6834	0	0	0	0
I_4	0	0	0	1	0	0.4671	0.5369	0	0	0
I_5	0	0	0	1	0	0	0	0	0	0
I_6	0	0	0	0	0	0	0	1	0.5369	0.4671
I_7	0	0	0	1	0	0.6834	0	0	0	0

Table 5 The similarity degree between movies

	I_1	I_2	I_3	I_4	I_5	I_6	I_7
I_1	1	0.8814	0.1542	0.1521	0.1867	0	0.1542
I_2	0.8814	1	0	0	0	0	0
I_3	0.1542	0	1	0.8874	0.8256	0	1
I_4	0.1521	0	0.8874	1	0.8148	0	0.8874
I_5	0.1867	0	0.8256	0.8148	1	0	0.8256
I_6	0	0	0	0	0	1	0
I_7	0.1542	0	1	0.8874	0.8256	0	1

Table 6 The similarity degree between users

	u_1	u_2	u_3	u_4	u_5	u_6
u_1	1	0	0.8814	0	1	0.1912
u_2	0	1	0.1542	0	0	0.3333
u_3	0.8814	0.1542	1	0	0.8814	0.3333
u_4	0	0	0	1	0	0
u_5	1	0	0.8814	0	1	0.1912
u_6	0.1912	0.3333	0.3333	0	0.1912	1

4 The proposed method (CU-FHR)

To alleviate the data sparsity problems and improve the recommendation accuracy, this paper develops an approach which integrates fuzzy content-based method (the similarity to the label attribute of each item), genetic algorithm with matrix factorization. It first employs the user similarity-based KNN cluster method to produce K core users to form a dense user-item rating matrix, and based on this matrix, genetic algorithm-based matrix factorization is applied to generate recommendations. This approach takes advantages of dimensionality reduction on two sides, the size of original rating matrix and latent features obtained by matrix factorization, which can deal with the sparsity problems and meanwhile reduce the computation complexity.

This section introduces the CU-FHR method in 3 steps, as shown in Figure 1.

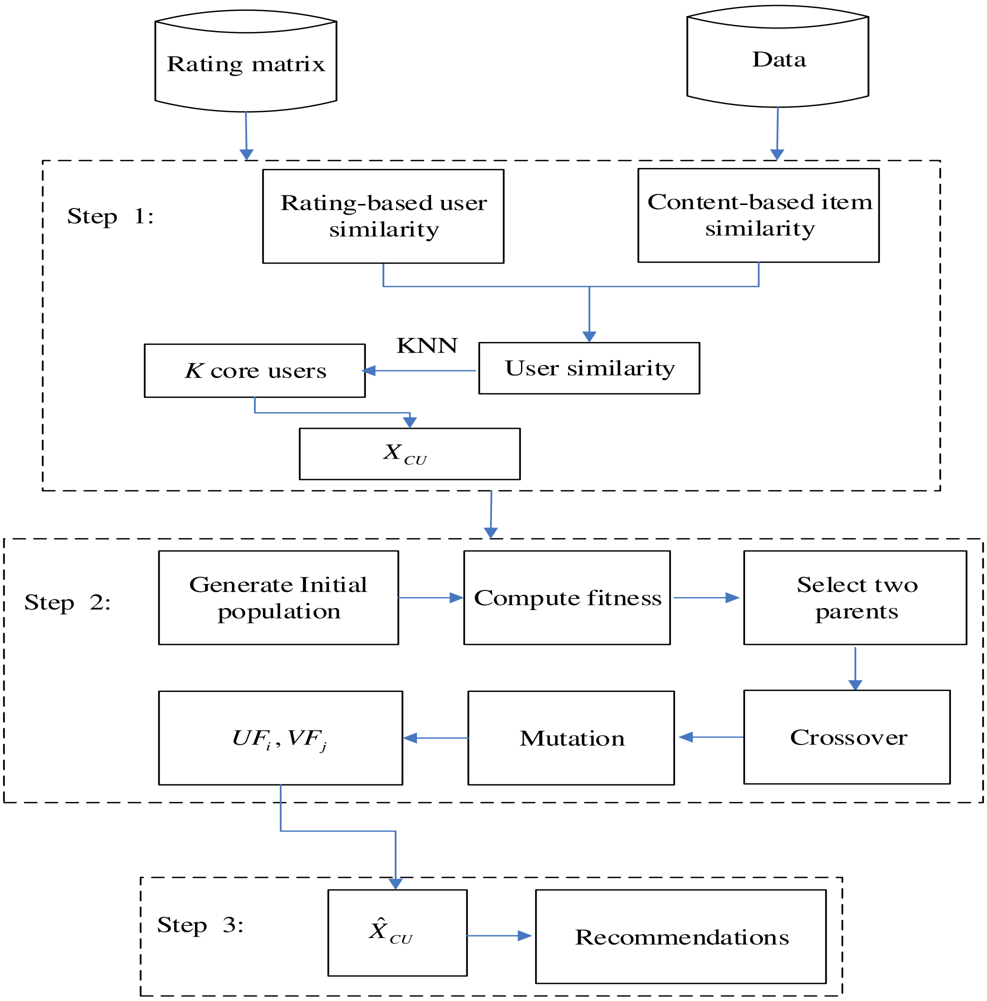

Figure 1 Procedure of CU-FHR method

Step 1: Find K core users based on KNN

In the work of Kilani [17], only the neighbor users of the active user were considered in recommendation, which will face an unsolvable difficulty. The neighbor users can't be found when this user is a new one to the system. To avoid this limitation, this paper will select a certain number of core users from the whole, abandoning the idea of an orientated particular user. Core users (CUs) are the set of users who have similarity with the most users. If a user in recommendation system is similar to the majority of other users, we assume that he (she) is similar to the newer. Based on the assumption, we can use the core users to generate recommendation which can handle the cold start problem.

Considering the fact that each member contributes differently to the recommendation according to their importance degree, we only utilize partial users' rating instead of all individual members' rating on item for the missing value prediction. Hence, we introduce an index to evaluate the importance of each member and propose a member importance-based model to identify core users. The core user-item rating matrix is then used as the input for matrix factorization for further recommendation.

1) The K_1 neighbor users to the each user will firstly be selected by KNN based on similarity,

where K_1 is a parameter. Suppose there are n users $U = u_1, u_2, \cdots, u_n$ and m items $I = I_1, I_2, \cdots, I_m$ in a certain recommendation system, then n sets of neighbor users will be available. Let NEI^i denote the neighbor users of user u_i, for each $u \in U$, we give the following two equations:

$$S_u^i = \begin{cases} 1 & u \in NEI^i \\ 0 & otherwise \end{cases}, \tag{7}$$

$$P_u^i = \begin{cases} r^i & u \in NEI^i \\ \infty & otherwise \end{cases}, \tag{8}$$

Where, r^i is the kth order of u in NEI^i.

2) We define MC_u as the importance of user u,

$$MC_u = \sum_{i=1}^{n} S_u^i \sum_{i=1}^{n} (1 / P_u^i), \tag{9}$$

3) Rank the users in descending order according to MC_u, then select the first K users as the core users, where K is a parameter which can impact the recommendation accuracy. It is noteworthy that if the active user who needs recommendation does not appear on the list of CUs, then the active user can be added into the CUs pool. Algorithm 1 presents how to find the CUs in detail.

Algorithm 1. Find K core users.

Input:

$X \in \mathbf{R}^{n \times m}$, original rating matrix;

$A = \{A_1, A_2, \cdots, A_t\}$, feature set of items;

K_1, the number of neighbor user for each user;

K, the number of core users;

Output:

CUs, arrays of core users;

X_{CU}, the corresponding rating matrix of CUs;

1: Compute the item similarity as in equation (5);

2: Compute the user similarity as in equation (6);

3: Obtain K_1 neighbors of each user according to KNN;

4: Compute the user importance as in equation (9);

5: Obtain K core users;

6: **return** CUs and X_{CU} ($X_{\mathrm{CU}} \in \mathbf{R}^{K \times m}$ or $X_{\mathrm{CU}} \in \mathbf{R}^{(K+1) \times m}$).

Step 2: Factorize the matrix X_{CU} based on genetic algorithm.

This step aims to predict the missing value in X_{CU} through matrix factorization based on genetic algorithm. The goal is to minimize the regularized differences between the original ratings in X_{CU} and the predicted ratings that is the product of user feature matrix and item feature matrix. Hence, the objective can be for an active user u,

$$\min l(X,U,V)=\begin{cases}\dfrac{1}{K}\sum_{i=1}^{K}\dfrac{1}{m}\sum_{j=1}^{m}|x_{ij}-\sum_{p=1}^{k}u_{ip}v_{pm}|+\beta(\|U_{Kk}\|^2+\|V_{km}\|^2), & u\in \mathrm{CU}_s\\ \dfrac{1}{K+1}\sum_{i=1}^{K+1}\dfrac{1}{m}\sum_{j=1}^{m}|x_{ij}-\sum_{p=1}^{k}u_{ip}v_{pm}|+\beta(\|U_{(K+1)k}\|^2+\|V_{km}\|^2), & u\notin \mathrm{CU}_s\end{cases}, \quad (10)$$

Where k is the number of latent features.

In designing genetic algorithm, real encode is employed to represent the individual, which consists of two parts in this paper. Without loss of generality, an individual is formally defined as what described in [17] for each $u\in \mathrm{CUs}$. That is,

$$individual=UF*VF=\begin{pmatrix}u_{11} & u_{12} & \cdots & u_{1k}\\ u_{21} & u_{22} & \cdots & u_{2k}\\ \vdots & \vdots & \vdots & \vdots\\ u_{K1} & u_{K2} & \cdots & u_{Kk}\end{pmatrix}\begin{pmatrix}v_{11} & v_{12} & \cdots & v_{1m}\\ \vdots & \vdots & \vdots & \vdots\\ v_{k1} & v_{k2} & \cdots & v_{km}\end{pmatrix}$$

The implementation of genetic algorithm includes four steps as follows. Table 7 shows the notations of some parameters used in genetic algorithm.

Table 7 Some notations

Symbols	meaning
Initialpop	initial population
chromosome	population after iteration
Fit(s)	the fitness of an individual s
Popsize	the size of the population
Croprob	crossover probability
Mutprob	mutation probability
k	the number of latent features
Maxiterations	the maximum iteration
$UF_i\in\mathbf{R}^{K\times k}$	user feature population
$VF_j\in\mathbf{R}^{k\times m}$	item feature population

1) Initializes the population. This step is to create an initial population with *Popsize* individuals, which is demonstrated by Algorithm 2. At first, similar to probabilistic matrix factorization, the elements in UF and VF are randomly generated according to $N(0,1)$ (Normal distribution with mean value 0 and variance 1), which can be realized by a function *Random*(0-1) that returns a value between 0 and 1. For the purpose of presentation, Algorithm 2 has assumed $X_{\mathrm{CU}}\in\mathbf{R}^{K\times m}$. The case for $X_{\mathrm{CU}}\in\mathbf{R}^{(K+1)\times m}$ is handled by the same way and has, therefore, been omitted in this step.

Algorithm 2. Initializes the population randomly.

Input: k, K, *Popsize*;

Output: *Initialpop*, UF, VF;

While $size<Popsize$ **do**

For each individual in the population **do**

For each $u_a(a=1,2,\cdots,K)$ in UF **do**

For each element $u_{ab}(a=1,2,\cdots,K;b=1,2,\cdots,k)$ in UF, **do**

$u_{ab}=Random(0-1)$;

End

End

For each $v_c (c = 1, 2, \cdots, k)$ in VF **do**

For each element $v_{cd} (c = 1, 2, \cdots, k; d = 1, 2, \cdots, m)$ in VF, **do**

$v_{cd} = Random(0-1)$;

End

End

End

End While

2) Compute the fitness. The fitness of each member in population is closely related to the objective of this member. The fitness function is defined in the following

$$fitness(s) = 1/\min l(X, U, V), \tag{11}$$

3) Select two parents. Roulette wheel parent selection method based on the fitness is used to choose two parents, leading to some individuals having high fitness be picked out with a big probability.

4) Crossover operator. This paper uses two-point crossover operator to produce two children. That is, the parts of the parents between these two crossover points, which are randomly generated, are swapped to produce two children. The detail is shown in the following. Suppose the two parent individuals are parent 1 and parent 2 as follows, the two crossover points randomly generated are $\overset{\downarrow}{5}$ *and* $\overset{\downarrow}{3}$ in parent 1, the corresponding $\overset{\downarrow}{4}$ *and* $\overset{\downarrow}{2}$ in parent 2, then we can get child 1 and child 2 after crossover.

$$parent\ 1 = \begin{pmatrix} 1 & 2 & 3 & \cdots & 5 \\ 4 & \overset{\downarrow}{5} & 4 & \cdots & 2 \\ 2 & 1 & 2 & \cdots & 3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 4 & 2 & 5 & \cdots & 1 \\ 5 & 3 & 2 & \cdots & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & \cdots & 5 \\ 4 & 5 & 4 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 5 & \overset{\downarrow}{3} & 2 & \cdots & 4 \end{pmatrix},$$

$$parent\ 2 = \begin{pmatrix} 5 & 2 & 4 & \cdots & 5 \\ 3 & \overset{\downarrow}{4} & 2 & \cdots & 2 \\ 3 & 5 & 3 & \cdots & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 3 & 4 & 5 & \cdots & 5 \\ 3 & 2 & 2 & \cdots & 4 \end{pmatrix} \begin{pmatrix} 3 & 4 & 3 & \cdots & 2 \\ 5 & 5 & 4 & \cdots & 3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 2 & \overset{\downarrow}{2} & 2 & \cdots & 4 \end{pmatrix}$$

$$child\,1=\begin{pmatrix}1&2&3&\cdots&5\\4&5&2&\cdots&2\\3&5&3&\cdots&2\\\vdots&\vdots&\vdots&\vdots&\vdots\\3&4&5&\cdots&5\\3&2&2&\cdots&4\end{pmatrix}\begin{pmatrix}3&4&3&\cdots&2\\5&5&4&\cdots&3\\\vdots&\vdots&\vdots&\vdots&\vdots\\2&3&2&\cdots&4\end{pmatrix},\quad child\,2=\begin{pmatrix}5&2&4&\cdots&5\\3&4&4&\cdots&2\\2&1&2&\cdots&3\\\vdots&\vdots&\vdots&\vdots&\vdots\\4&2&5&\cdots&1\\5&3&2&\cdots&3\end{pmatrix}\begin{pmatrix}1&2&3&\cdots&5\\4&5&4&\cdots&1\\\vdots&\vdots&\vdots&\vdots&\vdots\\5&2&2&\cdots&4\end{pmatrix}$$

5) Mutation operator. The goal is to introduce diversity. A random point in a child is selected with a given mutation probability, and the value is changed randomly into a value based on *Random*(0-1).

Remark 3 After mutation for each iteration, the elite strategy is used in the iteration of genetic algorithm. Therefore, the fitness of the population "*chromosome*" are totally computed for present generation, and the best individual with the highest fitness will replace the worst one in the last generation unconditionally.

Algorithm 3 gives the process of GA in CU-FHR.

After finishing all the iterations, we choose the optimal individual (UF, VF) to finish the matrix completion, $\hat{X} = UF * VF$, which is crucial to recommendation.

Algorithm 3. GA in CU-FHR

Input: k, *Popsize*, X_{CU}, *Initialpop*, *Croprob*, *Mutprob*, *Maxiterations*;

Output: UF, VF, $\hat{X} = UF * VF$;

$generation = 0$;

Chromosome= *Initialpop*;

While $generation < Maxiterations$ **do**

Fit(*Chromosome*);

S=max(*Fit*(*Chromosome*));

Chromosome1=*Select*(*Fit*(*Chromosome*));

New1=*crossover*(*Chromosome1*);

New2=*mutation*(*New1*);

S^*=min(*Fit*(*New2*));

New3=(*New2*- S^*, *S*);

Chromosome= *New3*;

$generation = generation + 1$;

End While

Step 3: Generate recommendation

According to the predicted ratings $\hat{X} = UF * VF$ obtained in Step 2, the unrated items for the active user are ranked. Then the top-*N* items can be selected and the recommendations can be generated accordingly.

5 Experiments and analysis

This section presents the experimental results and the related analysis through a data set

MovieLens (http://grouplens.org/datasets/movielens/). We first introduce the data sets and experiment setting, followed by the empirical experiments results and parameter analysis.

5.1 Data sets and experimental setting

This section will test the prediction accuracy of the proposed CU-FHR. MovieLens dataset is a common and well-known dataset used for RS research. It consists of 100,000 ratings from 943 users on 1682 movies, where the rating scale is from 1 to 5. The larger the rating, the higher the liking degree of users to the movie. Each user has rated at least 20 movies and all movies have been rated at least once. During the experiments, first, we randomly use 80% of movies and the corresponding ratings to compute the user similarity to obtain *K* core users in Step 1; second, we use the ratings extracted from the *K* core user as training data for matrix factorization of CU-FHR and all unrated ratings are to be predicted, then the remaining 20% of movies based on Step 1 are used as testing data to compare all the actual rates with the predicted ratings through $\min l(X,U,V)$. But it is worthy noting that the remaining 20% of the movies vary with different user, therefore, we make experiments for each user.

The baseline methods contain Kilani's NLM [17], Navgaran's NRS [16], CU-FHR, CU-FR (CU-FHR without MF), and CU-HR (CU-FHR without fuzzy content similarity). NLM and NRS are the MF recommendation methods based on genetic algorithm. NLM is an improvement of NRS. NLM hybridizes CF and MF models. NLM has been proven to be more effective than NRS. CU-FHR, an expansion of NLM, combines fuzzy theories and neighbor-based core user selection with MF. Besides, some ablation experiments are further conducted including CU-FR (CU-FHR without MF), CU-HR (CU-FHR without fuzzy content similarity).

We run all the experiments for 50 times for each user and then take the average value. The fixed parameters are listed in the following: *Popsize*=50, *Croprob*=1, *Mutprob*=0.001, *Maxiterations=100*.

We will use two groups of evaluation metrics to test CU-FHR.

1) Mean absolute error (MAE) and root mean square error (RMSE),

$$\text{MAE} = \sum_{i,j,X_{ij} \in Y} |\hat{X}_{ij} - X_{ij}| / |Y| \ , \tag{12}$$

$$\text{RMSE} = \sqrt{\sum_{i,j,X_{ij} \in Y} (\hat{X}_{ij} - X_{ij})^2 / |Y|}, \tag{13}$$

where $\hat{X}_{ij}$ and X_{ij} are the predicted and true ratings, *Y* is the test set, and |*Y*| is the number of the test set. The smaller the value, the better the performance of recommendation.

2) Precision and Recall,

$$\text{Precision} = \sum_i |R_i \cap T_i| / \sum_i |R_i|, \tag{14}$$

$$\text{Recall} = \sum_i |R_i \cap T_i| / \sum_i |T_i|, \tag{15}$$

Where R_i is the recommendation lists of user i in training set, T_i is the recommendation lists in testing set.

5.2 Results

In our experiments, the selected number of core users basically is consistent with that of neighbors obtained according to the similarity in [17].

First, we will explain the influence of k and K on the four evaluation metrics. After run all the experiments, we find it is more stable when k=8 and K=77. The visual results are shown in Figures 2-6.

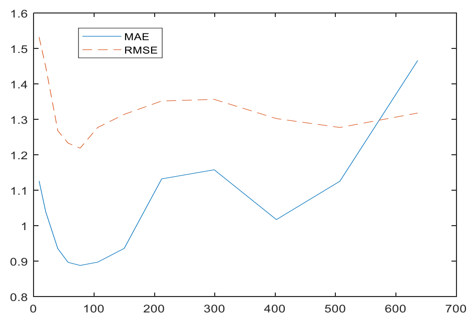

Figure 2 The results of MAE, RSME with different K when k=8

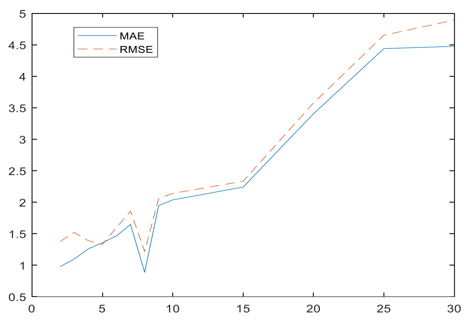

Figure 3 The results of MAE, RSME with different k when K=77

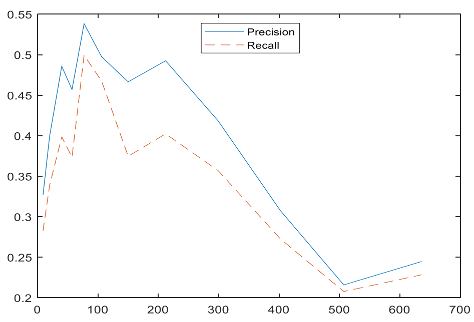

Figure 4 The results of Precision, Recall with different K when k=8

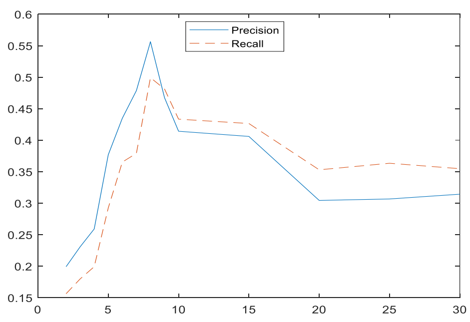

Figure 5 The results of Precision, Recall with different k when K=77

Figure 2 presents the variation of MAE and RMSE with K when $k=8$. In this figure, MAE and RMSE are basically smallest for K=77. During the 50 runs of experiments, although the value fluctuate with K, they always keep the smallest around K=77. These indicates the recommendation results are closely related to neighbor users with greater similarity degree which considers the movie's intrinsic attributes. Figure 4 presents the result of precision and recall with K when $k=8$. It is consistent with the result of Figure 2. Therefore, our method CU-FHR appropriately reflects the recommendation notion.

Figure 3 (Figure 5) presents the variation of MAE and RMSE (precision and recall) with k when K=77. In this figure, MAE and RMSE (precision and recall) reach the smallest (greatest) around k=8, then increase (decrease) as k increases from 8 to 25. This phenomenon tells us the number of latent features contributes to the recommendation accuracy, but it does not show the proportional relation. This indicates that redundant features not only increase the computation, but can't bring better effect. The fact shows that the decision-makers should focus on the main features of users and movie when making recommendation, which is accordance with decision-making preference.

Second, in model (10), β is a regularization parameter whose value will directly affect the accuracy of recommendation. If the value is too large, we will lose some potential information in recommendation, else the value is too small, the model can't effectively suppressing Gaussian noise. β can help prevent over fitting and improve the prediction accuracy of model (10) and further make a balance between them. To analyze the influence of β on MAE and RMSE, we make some experiments for fixed k=8, K=77. The results are presented as Figure 6.

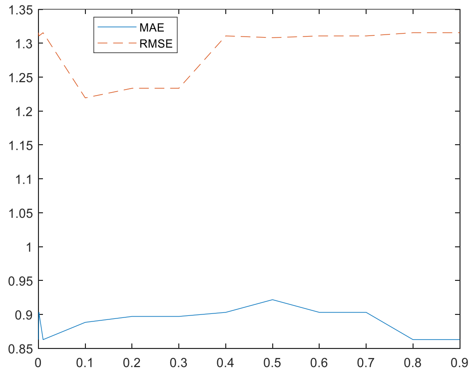

Figure 6 The results of MAE, RMSE with β

Figure 6 presents the variation of MAE and RMSE with the regularization tradeoff parameter β. From the figure, we see that the results were not influenced greatly when β varied from 0 to 0.9, but the results are better when $\beta \in (0, 0.1]$.

Moreover, we conduct some ablation experiments of CU-FHR. The relative results are further presented in Figures 7-9. Figure 7 gives the variation of precision and recall for CU-FR with different K. CU-FR is a neighbor-based CF method, the precision (recall) will increase with the number of neighbors (core users). Figure 8 (Figure 9) gives the variation of MAE and RMSE (precision and recall) for CU-HR with k=8. CU-HR is almost same as the process of CU-FHR, the results have similar inclination to CU-FHR, only presenting slightly different in concrete value.

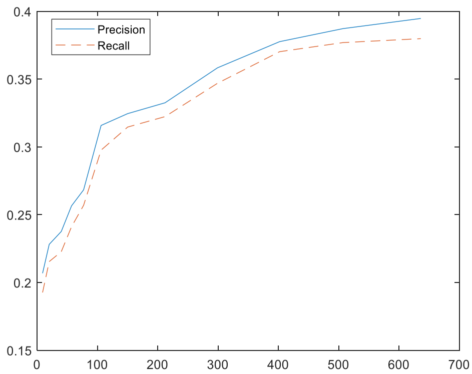

Figure 7 The results of Precision, Recall for CU-FR with different K

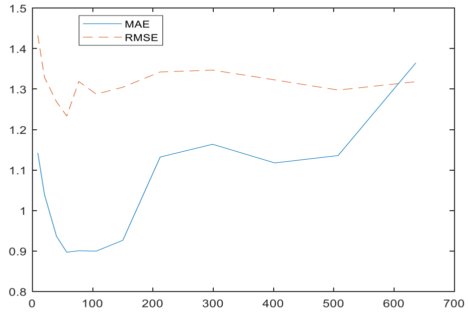

Figure 8 The results of MAE, RMSE for CU-HR with different K when k=8

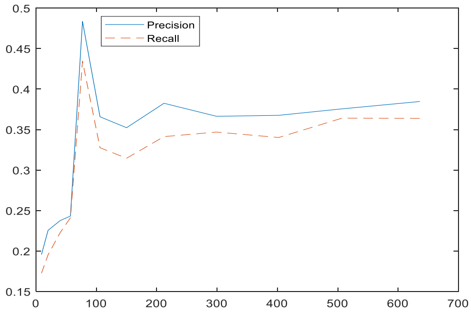

Figure 9 The results of Precision, Recall for CU-HR with different K when k=8

Finally, we compare our CU-FHR with Kilani's [17] (NLM) and Navgaran's [16] (NRS). All the parameters setting is listed: *Popsize*=50, *Croprob*=1, *Mutprob*=0.001, *Maxiterations*=100, *K*=77, *k*=8 for all the experiments; the regularization parameter $\beta = 0.1$ for CU-FHR and CU-HR; the parameter $\alpha = 0.0002, \beta = 0.02$ in updating the value of the selected mutation point for NRS [16]. By making experiments, the results are shown as Table 8.

Table 8 The comparison results

	CU-FHR	NLM	NRS	CU-FR	CU-HR
MAE	0.8883	0.9034	----	----	0.8970
RMSE	1.2193	1.3146	----	----	1.2335
Precision	0.5376	0.4925	0.1992	0.3947	0.4836
Recall	0.4927	0.4023	0.1563	0.3798	0.4345

Table 8 shows that our method CU-FHR is slightly better than NLM and NRS in spite of MAE, RMSE, Precision and Recall. It is almost 1.62%, 7.25% less than NLM in MAE and RMSE respectively; 9.16%, 22.47% more than NLM in Precision and Recall respectively. Furthermore, it is nearly 169.88%, 215.23% more than NRS in Precision and Recall respectively. Besides, from the ablation experiments, we know that the Precision and Recall are a bit less than that of CU-FHR, and the MAE and RMSE are a bit more than that of CU-FHR, which shows hybrid recommendation methods are more effective and demonstrates better performance.

6 Conclusions

We have proposed a new hybrid recommendation approach, called CU-FHR, which combines matrix factorization model and genetic algorithm with fuzzy set techniques for movie recommendation. In this study, we have incorporated the content feature of items described by fuzzy sets into the similarity computation. The improved similarity can have a positive influence on the rating and classification because it can describe the ambiguity of the prototype of movies. So it serves as a baseline standards of the importance of a user for selecting the core users. CU-FHR has used core users rather than all the users to establish a matrix factorization model, which is solved through genetic algorithm to predict the rating on items. Furthermore, we have used MovieLens to test the performance. Experiments results show that CU-FHR is almost 1.62%, 7.25% less than NLM in MAE and RMSE respectively; and 9.16%, 22.47% more than NLM in Precision and Recall respectively. In addition, it is nearly 169.88%, 215.23% more than NRS in Precision and Recall respectively. Therefore, CU-FHR achieved better prediction accuracy compared with other recommendation methods.

In the future, more researches are needed to improve the recommendation diversity to satisfy different personalization requirements. Furthermore, there still exists some interesting issues to be discussed. For example, the extreme rating in application systems can be used to select core users. This research direction may possibly enhance the diversity and satisfaction degree of personalized recommendations.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (72101082, 62076088), the Natural Science Foundation of Hebei Province (F2021208011).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicts of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Lu J, Wu D, Mao M, Wang W, Zhang G (2015) Recommender system application developments: A survey, Decision Support Systems, 74:12-32. https://doi.org/10.1016/j.dss. 2015.03. 008.

[2] Su X, Khoshgoftaar T M (2009) A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence. https://doi.org/10.1155/2009/421425.

[3] Bobadilla J, Ortega F, Hernando A, Alcalá J (2011) Improving collaborative filtering recommender system results and performance using genetic algorithms, Knowledge-Based Systems, 24:1310-1316. https://doi.org/10.1016/j.knosys.2011.06.005.

[4] Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey, Knowledge-Based Systems, 46: 109-132. https://doi.org/10.1016/j.knosys.2013.03.012.

[5] Mao M, Lu J, Zhang G, Zhang J (2017) Multirelational Social Recommendations via Multigraph Ranking, IEEE Transactions on Cybernetics, 47(12): 4049-4061. https://doi.org/10.1109/TCYB. 2016. 2595620.

[6] Martín-Vicente M I, Gil-Solla A, Ramos-Cabrer M, Pazos-Arias José J, Blanco-Fernández Y, López-Nores M (2014) A semantic approach to improve neighborhood formation in collaborative recommender systems, Expert Systems with Applications, 41(17):7776-7788. https://doi.org/ 10.1016/ j.eswa.2014.06.038.

[7] Ntoutsi E, Stefanidis K, Norvag K, Kriegel H P (2012) Fast group recommendations by applying user clustering, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-642

-34002-4_10.

[8] Ali M, Jung L T, Abdel-Aty A H, Abubakar M Y, Elhoseny M, Ali I (2020) Semantic-k-NN algorithm: An enhanced version of traditional k-NN algorithm, Expert Systems With Applications, 15: 1113374. https://doi.org/10.1016/j.eswa.2020.113374.

[9] Del Corso G M, Romani F (2019) Adaptive nonnegative matrix factorization and measure comparisons for recommender systems, Applied Mathematics and Computation, 354: 164-179. https:// doi.org/10.1016/j.amc.2019.01.047

[10] Guan N, Tao D, Luo Z, Yuan B (2012) Online nonnegative matrix factorization with robust stochastic approximation, IEEE Transactions on Neural Networks and Learning Systems, 23(7) : 1087-1099. https://doi.org/10.1109/TNNLS.2012.2197827.

[11] Liu B, Xiong H, Spiros P, Fu Y, Yao Z (2015) A general geographical probabilistic factor model for pointof interest recommendation, IEEE Transactions on Knowledge and Data Engineering, 27(5): 1167-1179. https://doi.org/10.1109/TKDE.2014.2362525.

[12] Lei C, Dai H, Yu Z, Li R (2020) A service recommendation algorithm with the transfer learning based matrix factorization to improve cloud security, Information Sciences, 513: 98-111. https:// doi.org/10.1016/j.ins.2019.10.004.

[13] Candès E J, Recht B (2009) Exact matrix completion via convex optimization, Foundations of Computational Mathematics, 9: 717-772. DOI 10.1007/s10208-009-9045-5.

[14] Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems, Computer, 42 (8): 30-37.

[15] Duma M, Twala B (2018) Optimising latent features using artificial immune system in collaborative filtering for recommender systems, Applied Soft Computing, 71: 183-198. https://doi.org/10.1016/ j.asoc.2018.07.001.

[16] Navgaran D Z, Moradi P, Akhlaghian F (2013) Evolutionary based matrix factorization method for collaborative filtering systems, 2013 21st Iranian Conference on Electrical Engineering, ICEE 2013. https://doi.org/10.1109/IranianCEE.2013.6599844.

[17] Kilani Y, Otoom A F, Alsarhan A, Almaayah M (2018) A genetic algorithms-based hybrid recommender system of matrix factorization and neighborhood-based techniques, Journal of Computational Science, 28:78-93. https://doi.org/10.1016/j.jocs.2018.08.007.

[18] Holland J, (1975) Adaptation in Nature and Artificial Systems. University of Michigan: USA

[19] Yera R, MartÍnez L (2017) Fuzzy Tools in Recommender Systems: A Survey, International Journal of Computational Intelligence Systems, 10: 776-803. https://doi.org/10.2991/ijcis.2017. 10.1.52.

[20] Zenebea A, Norciob A F (2009) Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems, Fuzzy Sets and Systems, 160: 76-94. https://doi.org/10.1016/j.fss.2008.03.017.

[21] Zadeh L A (1965) Fuzzy sets, Information and Control, 8(3): 338-353. https://doi.org/10.1016/ S0019-9958 (65) 90241-X.

[22] Zadeh L A (1975) The concept of a linguistic variable and its application to approximate reasoning-I, Information Sciences, 8 (3) :199-249. https://doi.org/10.1016/0020-0255(75)90036-5.

[23] Zhang Z, Lin H, Liu K, Wu D, Zhang G, Lu J (2013) A hybrid fuzzy-based personalized

recommender system for telecom products/services, Information Sciences, 235:117-129. https://doi. org/ 10.1016/j.ins. 2013.01.025.

[24] Loepp B, Donkers T, Kleemann T, Ziegler J (2019) Interactive recommending with Tag-Enhanced Matrix Factorization (TagMF), International Journal of Human-Computer Studies, 121: 21-41. https://doi.org/10.1016/j.ijhcs.2018.05.002.

[25] Anwaar F, Iltaf N, Afzal H, Nawaz R (2018) HRS-CE: A hybrid framework to integrate content embeddings in recommender systems for cold start items, Journal of Computational Science, 29 : 9-18. https://doi.org/10.1016/j.jocs.2018.09.008.

[26] Viktoratos I, Tsadiras A, Bassiliades N (2018) Combining community-based knowledge with association rule mining to alleviate the cold start problem in context-aware recommender systems, Expert Systems with Applications, 101: 78-90. https://doi.org/10.1016/j.eswa.2018.01.044.

[27] Ayub M, Ali Ghazanfar M, Mehmood Z, Alyoubi K H, Alfakeeh A S (2020) Unifying user similarity and social trust to generate powerful recommendations for smart cities using collaborating filtering-based recommender systems. Soft Computing 24:11071-11094, https://doi. org / 10.1007/s00500-019 -04588 -x.

[28] Jelodar H, Wang Y, Xiao G, Rabbani M, Zhao R, Ayobi S, Hu P, Masood I (2020) Recommendation system based on semantic scholar mining and topic modeling on conference publications, Soft Computing https://doi.org/10.1007/s00500-020-05397-3.

[29] Burke R (2002) Hybrid recommender systems: survey and experiments, User Modeling and User-Adapted Interaction, 12 (4) : 331-370. https://doi.org/10.1023/A:1021240730564.

[30] Kermany N R, Alizadeh S H (2017) A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques, Electronic Commerce Research and Applications, 21: 50-64. https://doi. org/10.1016/j.elerap.2016.12.005.

[31] Palomaresa I, Brownec F, Davis P (2018) Multi-view fuzzy information fusion in collaborative filtering recommender systems: Application to the urban resilience domain, Data & Knowledge Engineering, 113: 64-80. https://doi.org/10.1016/j.datak.2017.10.002.

[32] Aguilara J, Valdiviezo-Dıázb P, Riofrio G (2017) A general framework for intelligent recommender systems, Applied Computing and Informatics, 13: 147-160. https://doi.org/10.1016/j.aci.2016. 08.002.

[33] Schafer J, Frankkowski D, Herlocker J (2007) Collaborative filtering recommender systems, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 291-324. https://doi.org/10.1007/978-3-540- 72079-9_9.

[34] Krasnoshchok O, Lamo Y (2014) Extended content-boosted matrix factorization algorithm for recommender systems, International Conference on Knowledge-Based and Intelligent Information & Engineering Systems-Procedia Computer Science, 35: 417-426. https://doi.org/10.1016/j.procs. 2014.08. 122.

[35] Zhang Q, Lu J, Wu D, Zhang G (2019) A Cross-Domain Recommender System With Kernel-Induced Knowledge Transfer for Overlapping Entities, IEEE Transactions on Neural Networks and Learning Systems, 30(7): 1998-2012. https://doi.org/10.1109/TNNLS.2018. 2875144.