APPLICATION OF SOFT COMPUTING

Hybrid recommender system with core users selection

Chenxia Jin1,2 • Jusheng Mi1 • Fachao Li2 • Jiahuan Zhang1

Accepted: 23 July 2022 / Published online: 25 August 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

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 modeling. 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 advancement of internet/technology brings an exponential expansion of web contents, often confusing the users to finding information that most meet their needs. Therefore, information overload is a big challenge for users to make correct decision. Recommender system (RS) is an efficient tool to address information overload problem (Selvi and Sivasankar 2019). It can assist users by providing them with selection of appropriate items based on their preferences. Its basic idea is that similar users like similar items. Recently, RSs have been widely applied in many web-based domains such as e-commerce, e-government and e-learning (Jelodar et al. 2020; Lu et al. 2015; Su and Khoshgoftaar 2009).

Collaborative filtering (CF) is one of the most consolidated methods in RSs, 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 accepted if the rating data of the items 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. Unfortunately, the prediction accuracy of the calculated unseen preferences is considerably low (Martín-Vicente et al. 2014), which will result in poor recommendation. Obviously, CF often suffers from data sparsity problems that are particularly severe and challenging (Bobadilla et al. 2011; Bobadilla et al. 2013; Mao et al. 2017; Su and Khoshgoftaar 2009). To address this issue, a large amount of research work has been done to improve the recommendation performance. 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 (Ntoutsi et al. 2012; Su and Khoshgoftaar 2009). But the filling results do not consider the variance of the ratings.
Either the values are often away from the reality (Su and Khoshgoftaar 2009) or they distort the data distribution (Ntoutsi et al. 2012). Of course, there are still some concerns on implicit feedback based on users’ activities (Najafabadi et al. 2019). However, these studies cannot consider the role of highly influential users of the social circle which is very crucial to modeling the user’s profiles (Choudhary et al. 2020). According to famous Pareto’s principle, just a few key ideas are critical to decision-making. Similarly, in a recommender system, a few highly influential users will have a great effect on recommendation quality. Therefore, how to identify the most important users is the main objective of improving recommendation accuracy. Motivated by this idea, we incorporate the core users to generate recommendation in this study. To achieve this, the clusters of users are firstly produced according to the similarity using KNN (Ali et al. 2020), then from which the core user is 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. However, there are two major issues should be paid to more attention. 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 a 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 is usually categorized into two groups: memory-based and model-based methods (Lu et al. 2015). Memory-based CF can either be user-based or item-based methods. User-based methods generate recommendations according to the preferences of neighbor users (the most similar users), while item-based methods recommend the most similar items according to the past preferences of a user. Model-based methods can make recommendations through creating some models, such as matrix factorization (Corso and Romani 2019), probabilistic model (Guan et al. 2012) and Bayesian classifier (Liu et al. 2015). Among them, matrix factorization (MF) is more preferred and has been widely applied in many commercial fields (Su and Khoshgoftaar 2009). 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 (Lei et al. 2020). 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 (Candès and Recht 2009). The common techniques to find the optimal solutions include stochastic gradient descent (Candès and Recht 2009; Koren et al. 2009), artificial immune system approach (Duma et al. 2018) and genetic algorithm (Bobadilla et al. 2011; Kilani et al. 2018; Navgaran et al. 2013) 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 (Holland 1975). It searches the optimal solutions through repeatedly executing the selection, crossover and mutation operation. For example, Navgaran (2013) gave the MF-based recommendation by combining with genetic algorithm considering the whole users and items. Kilani (2018) improved the work of Navgaran (2013), 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 up the recommendation process and produce 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 alleviate the influence of data sparsity.

Generally speaking, the gleaned features or descriptions of items often contain some uncertainties. For example, movies are often described by multi-genres and multi-actors. That is, movies can be labeled as romantic, action, scary and animation and so on. If taking all the genres content into recommender system, the accuracy of content-based recommendation will be improved (Zenebea and Norciob 2009). Besides, it assumed that each item can be appropriately represented as a fuzzy set by which the similarity relationship for the set of items can be developed (Yera and Martínez 2017). In the case of this framework, the membership degree of a movie to each genre is a pre-step for comprehensively integrating all content information. Accordingly, we use a fuzzy membership determination method of Zenebea and Norciob (2009) for the proposed recommender system.
characterized by a membership function

\[ \mu_A(x) : x \in X \rightarrow [0, 1] \]

where \( \mu_A(x) \) 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. Specialy, 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

\[ A = \{(x, \mu_A(x)) / x \in X, \mu_A(x) \in [0, 1] \} \]

This study extends traditional CF and the framework of Kilani (Kilani et al. 2018). 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 both rated items, in which fuzzy sets are adopted to capture the uncertainty in rating. 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 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 Sect. 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 Sect. 5. The conclusion and further study are presented in Sect. 6.

2 Related works

2.1 Fuzzy sets techniques

Fuzzy set theory, firstly proposed by Zadeh (1965), 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

\[ \mu_A(x) : x \in X \rightarrow [0, 1] \]

where \( \mu_A(x) \) denotes a fuzzy set “liked”, then \( \mu_A(x) \in [0, 1] \) denotes the degree of \( x \) being liked by users. Specialy, 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

\[ A = \{(x, \mu_A(x)) / x \in X, \mu_A(x) \in [0, 1] \} \]

variable, whose values are words and sentences, can be a more realistic option to represent imprecise assessments (Zadeh 1975; Zhang et al. 2013). The linguistic variable with qualitative description is more in line with the human thinking patterns. At the same time, the fact the linguistic variable is 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 (Loepp et al. 2019; Zhang et al. 2013). Recommender systems use the interaction data between users and items to automatically predict and identify the user preferences to make recommendations. Generally, recommender methods are roughly categorized into four types: content-based (CB) (Anwaar et al. 2018), collaborative filtering-based (CF) (Su and Khoshgoftaar 2009), knowledge-based (KB) (Viktoratos et al. 2018), hybrid recommender systems (Ayub et al. 2020; Jelodar et al. 2020; Kilani et al. 2018). Among them, hybrid recommender systems 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 (Burke 2002; Kermany et al. 2017; Zhang et al. 2013). 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 et al. (2018) established a genetic algorithm-based hybrid recommendation system of neighborhood-based and matrix factorization-based approach. Zhang et al. (2013) combined user-based and item-based collaborative filtering methods with fuzzy set techniques and applied it to mobile product and service recommendation. Anwaar et al. (2018) 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 (2002) proposed a classification of hybrid recommender systems and listed seven basic hybridization mechanisms. Kermany et al. (2017) incorporated demographic information and an item-based ontological semantic filtering approach into fuzzy multi-criteria collaborative filtering for movie recommendation. Viktoratos et al. (2018) hybridized a collaborative system and a knowledge-based system to solve the cold start problem. Palomares et al. (2018) 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 (Aguilera et al. 2017).

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 \( X \in \mathbb{R}^{n \times m} \) (\( \mathbb{R}^{n \times m} \) represents the sets of matrix with \( n \) rows and \( m \) columns), then matrix factorization is to decompose \( X \) into two low-dimension matrices: the user feature matrix \( U \in \mathbb{R}^{n \times k} \) and the item feature matrix \( V \in \mathbb{R}^{k \times m} \). \( k(k \ll m, k \ll n) \) is the number of latent factors and can be an adjusted parameter in experiment setting. Formally, \( X \) is denoted approximately by \( X \approx UV \), in which the \( i \)th row of \( U \) and the \( j \)th column of \( V \) represents the \( i \)th user and the \( j \)th item, respectively. The goal of matrix factorization is to find \( U \), \( V \) such that \( X \approx X \).

In \( X \), \( x_{ij} \in \{1, 2, 3, 4, 5, \Delta\} \), “1–5” denotes the rating score of user \( i \) to item \( j \); \( \Delta \) denotes user \( i \) did not rate item \( j \). The recommendation task is to predict the value of \( \Delta \) in \( X \), by which the top items with the highest rating will be recommended to a specific user. Let \( \text{err}_{ij} = |x_{ij} - \hat{x}_{ij}| \) 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 Kiliani et al. (2018). In general, matrix factorization can be converted into an optimization problem as follows:

\[
\min l(X, U, V) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \text{err}_{ij} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} |x_{ij} - \hat{x}_{ij}|
\]

where \( \hat{x}_{ij} = \sum_{p=1}^{k} u_{ip} v_{pj} \) 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 (Schafer et al. 2007). Therefore, the objective function to measure the total loss with regularization penalty can be

\[
\min l(X, U, V) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} |x_{ij} - \hat{x}_{ij}| + \beta(||U||^2 + ||V||^2)
\]

where \( \beta \) 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 (Krasnoshchok et al. 2014; Zhang et al. 2019) and GAs (Bobadilla et al. 2011; Kilani et al. 2018; Navgaran et al. 2013).

### 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 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 cannot 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 \( I = \{I_1, I_2, \ldots, I_m\} \) be an items set, \( A = \{A_1, A_2, \ldots, A_L\} \) be an features sets of item, then for an item \( I_j (j = 1, 2, \ldots, m) \), it can take multiple values \( A_1, A_2, \ldots, A_L(L \leq t) \). The multi-valued features in an item can be represented by a fuzzy set. The membership function of item \( I_j \) to value \( A_k (k = 1, 2, \ldots, L) \), 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 proposed by Zenebea and Norciob (Zenebea et al. 2009) to determine the value of \( \mu_{A_k}(I_j) \).

\[
\mu_{A_k}(I_j) = \frac{p_k}{2\sqrt{\pi}L_k(p_k-1)}
\]

where \( |L_k| = L \) is the number of features values of \( A \) associated with item \( I_j (j = 1, 2, \ldots, m) \), \( p_k (1 \leq p_k \leq L) \) is the rank position of value \( A_k \), \( x > 1 \) is a parameter that needs to be determined as a threshold to control the difference between consecutive values of \( A \) in \( I_j (j = 1, 2, \ldots, m) \).
For an given item $I_j$ characterized by a series of feature values $A_k (k = 1, 2, \ldots, L)$, we can get a vector as follows,

$$\{ (A_1, \mu_{A_1}(I_j)), (A_2, \mu_{A_2}(I_j)), \ldots, (A_L, \mu_{A_L}(I_j)) \} \tag{4}$$

For items $I_j$ and $I_l$ with the representations as Eq. (4), a cosine vector-based similarity measure between $I_j$ and $I_l$ is defined as

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

**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 $A_k, k = 1, 2, \ldots, L$ 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 $I_1 = \{\text{Castle in the Sky: Action/Adventure/Animation/Fantasy/Romance/Family} \}$, $I_2 = \{\text{Copycat 1995: Crime/Mystery/Thriller/Drama} \}$, we can compute the membership degree of each movie to each feature $A_k (k = 1, 2, \ldots, 10)$ with $\alpha = 1.2$ by formula (3) and the similarity of two movies by formula (5). The results are presented in Table 1.

### 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 cannot be computed through Cosine-based method. However, if two users have rated different items which are very similar in content and 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 Eq. (5):

$$s(u_1, u_2) = \begin{cases} \frac{w_{S_1}(u_1, u_2) + (1 - w)_{S_2}(u_1, u_2)}{2}, & S_{u_1} \cup S_{u_2} \neq \phi \\ 0, & S_{u_1} \cup S_{u_2} = \phi \end{cases} \tag{6}$$

In Eq. (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. $s_1(u_1, u_2) = \frac{\sum_{I_k \in S_{u_1} \cap S_{u_2}} r_{u_1, k} r_{u_2, k}}{\sqrt{\sum_{I_k \in S_{u_1}} (r_{u_1, k})^2} \sqrt{\sum_{I_k \in S_{u_2}} (r_{u_2, k})^2}}$ , $\ S_{u_1 \cap u_2}$ is the set of items of both $u_1$ and $u_2$ have rated.

3. $s_2(u_1, u_2) = \frac{\sum_{(I_k, I_k') \in S} r_{u_1, k} r_{u_2, k'} s(I_k, I_k')}{\gamma \sqrt{\sum_{(I_k, I_k') \in S} (r_{u_1, k})^2} \sqrt{\sum_{(I_k, I_k') \in S} (r_{u_2, k'})^2}}$ , $r_{u_1, k}, r_{u_2, k'}$ are the rating score of user $u_1$ on item $I_k$, user $u_2$ on item $I_k'$, respectively, $S = S_{u_1} \cup S_{u_2} - S_{u_1 \cap u_2}$.

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

5. $\gamma = |S_{u_1} - S_{u_2}| * |S_{u_2} - S_{u_1}|$ 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. $w = |S_{u_1 \cap u_2}| / |S_{u_1} \cup S_{u_2}|$ 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 Eq. (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. $s(u_1, u_2)$ considers not

| Table 1 Fuzzy representation of item similarity in feature spaces |
|---------------------------------|
| Action | Adventure | Animation | Fantasy | Romance | Family | Crime | Mystery | Thriller | Drama |
|-------|-----------|-----------|---------|---------|--------|-------|---------|----------|-------|
| $\mu_{A_1}(I_1)$ | 1 | 0.3114 | 0.2162 | 0.1596 | 0.1212 | 0.0938 | 0 | 0 | 0 |
| $\mu_{A_2}(I_2)$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.4380 | 0.3503 |
| $s(I_1, I_2)$ | 0 | | | | | | | | |
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, the more similar the two users”. The similarity is used to cluster the users aiming to find the core users in recommendation systems.

In the following, we will use a small-scale example to better illustrate our method. Table 2 gives the detailed information of seven movies including the titles and the corresponding genres. Table 3 gives the ratings of six users to the seven movies. Then, according to the above discussions, the membership degree of each movie to the genres, the similarity between two movies and the similarity between two users are presented in Tables 4, 5 and 6, respectively.

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, 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.

Following will introduce the CU-FHR method in detail. We first give the concrete implementation steps of CU-FHR through Fig. 1.

**Step 1: Find \( K \) core users based on KNN**

In the work of Kilani et al. (2018), only the neighbor users of the active user were considered in recommendation, which will face an unsolvable difficulty. The neighbor users cannot 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.

\( \text{(1) The } K_1 \text{ neighbor users to the each user will firstly be selected by KNN based on similarity, where } K_1 \text{ is a parameter. Suppose there are } n \text{ users } \{u_1, u_2, \ldots, u_n\} \text{ and } m \text{ items } \{I_1, I_2, \ldots, I_m\} \text{ in a certain recommendation system, then } n \text{ sets of neighbor users will be available. Let } \text{NEI}^u_i \text{ denote the neighbor users of user } u_i \text{, for each } u \in U, \text{ we give the following two equations:} \)

Table 2 The information on movies

| Movie ID | Title                | Genres                                         |
|----------|----------------------|-----------------------------------------------|
| \( I_1 \) | Toy Story (1995)     | Adventure|Animation|Children|Comedy|Fantasy                                  |
| \( I_2 \) | Jumanji (1995)       | Adventure|Children|Fantasy                                  |
| \( I_3 \) | Grumpier Old Men (1995) | Comedy|Romance                                  |
| \( I_4 \) | Waiting to Exhale (1995) | Comedy|Drama|Romance                                  |
| \( I_5 \) | Father of the Bride Part II (1995) | Comedy                                  |
| \( I_6 \) | Heat (1995)          | Action|Crime|Thriller                                  |
| \( I_7 \) | 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         |           |           |           |           |           |
| \( u_3 \) |           |           | 3         |           |           |           |           |
| \( u_4 \) |           |           |           | 3         |           |           |           |
| \( u_5 \) |           |           |           |           | 3         |           |           |
| \( u_6 \) |           |           |           |           |           | 5         | 5         |
The membership degree of movie to genres

| Adventure | Animation | Children | Comedy | Fantasy | Romance | Drama | Action | Crime | Thriller |
|-----------|-----------|----------|--------|---------|---------|-------|--------|-------|----------|
| $\mu_{Ak}(I_1)$ | 1 | 0.3662 | 0.2718 | 0.2113 | 0.1676 | 0 | 0 | 0 | 0 |
| $\mu_{Ak}(I_2)$ | 1 | 0 | 0.5369 | 0 | 0.4671 | 0 | 0 | 0 | 0 |
| $\mu_{Ak}(I_3)$ | 0 | 0 | 0 | 1 | 0 | 0.6834 | 0 | 0 | 0 |
| $\mu_{Ak}(I_4)$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $\mu_{Ak}(I_5)$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.5369 | 0.4671 |
| $\mu_{Ak}(I_6)$ | 0 | 0 | 0 | 0 | 1 | 0 | 0.6834 | 0 | 0 | 0 |

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:
- $X \in \mathbb{R}^{n \times m}$, original rating matrix;
- $A = \{A_1, A_2, \ldots, A_k\}$, 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 by equation (5);
2: Compute the user similarity by equation (6);
3: Obtain $K_1$ neighbors of each user according to KNN;
4: Compute the user importance by equation (9);
5: Obtain $K$ core users;
6: Return CUs and $X_{CU}$ ($X_{CU} \in \mathbb{R}^{K \times m}$ or $X_{CU} \in \mathbb{R}^{(K+1) \times m}$).

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 $u$,

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 was described by Kilani et al. (2018) for each \( u \in \text{CU}_s \). That is,

\[
\begin{align*}
\text{individual} &= UF \ast VF \\
&= \begin{pmatrix}
    u_{11} & u_{12} & \cdots & u_{1k} \\
    u_{21} & u_{22} & \cdots & u_{2k} \\
    \vdots & \vdots & \ddots & \vdots \\
    u_{K1} & u_{K2} & \cdots & u_{Kk}
\end{pmatrix} \begin{pmatrix}
    v_{11} & v_{12} & \cdots & v_{1m} \\
    v_{21} & v_{22} & \cdots & v_{2m} \\
    \vdots & \vdots & \ddots & \vdots \\
    v_{km} & v_{km} & \cdots & v_{km}
\end{pmatrix}
\end{align*}
\]

The implementation of genetic algorithm includes five steps stated as follows. Firstly, we will give the notations of some parameters used in genetic algorithm as shown in Table 7.

![Fig. 1 Procedures of CU-FHR method](image)
(1) **Initializes the population** This step is to create an initial population with \( P_{\text{size}} \) 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 \( \text{Random}(0-1) \) that returns a value between 0 and 1. For the purpose of presentation, Algorithm 2 has assumed \( X_{\text{CU}} \in \mathbb{R}^{K \times m} \). The case for \( X_{\text{CU}} \in \mathbb{R}^{(K+1) \times m} \) is handled by the same way and has, therefore, been omitted in this step.

Algorithm 2. Initializes the population randomly.

**Input:** \( k, K, P_{\text{size}}; \)

**Output:** \( \text{Initial pop, } UF, VF; \)

**While** size \(< P_{\text{size}} \) 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**

\[
 u_{ab} = \text{Random}(0-1);
\]

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**

\[
 v_{cd} = \text{Random}(0-1);
\]

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

\[
 \text{fitness}(s) = 1/ \min \ l(X,U,V)
\]

(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 \( \tilde{5} \) and \( \tilde{3} \) in parent 1, the corresponding \( 4 \) and \( 2 \) in parent 2, then we can get child 1 and child 2 after crossover.
Algorithm 3. GA in CU-FHR

Input: $k, \text{Popsize, } X_{CU}, \text{Initialpop, Croprob, Mutprob, Maxiterations}$;

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

$chromosome = Initialpop$;

While $generation < Maxiterations$ do

  $Fit(chromosome);$  
  $S = \text{max}(Fit(chromosome));$  
  $chro = Select(Fit(chromosome));$  
  $New1 = \text{crossover}(chro);$$New2 = \text{mutation}(New1);$  
  $S^* = \text{min}(Fit(New2));$$New3 = (New2 - S^*, S);$  
  $chro = New3;$  
  $generation = generation + 1;$

End While

(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” is 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,
\( \hat{X} = UF \ast VF \), which is crucial to recommendation.

**Step 3: Generate recommendation** According to the predicted ratings \( \hat{X} = UF \ast VF \) 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. MovieLens is a common and open dataset that is often used for recommender system research, which can be downloaded from [http://grouplens.org/datasets/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 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 \( \min I(X, U, V) \). But it is worth noting that the remaining 20% of the movies vary with different user, therefore, we make experiments for each user.

The baseline methods contain NLM (Kilani et al. 2018), NRS (Navgaran et al. 2013), 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),
   
   \[
   \text{MAE} = \frac{1}{|Y|} \sum_{i,j : X_{ij} \in Y} |\hat{X}_{ij} - X_{ij}| \\
   \text{RMSE} = \sqrt{\frac{1}{|Y|} \sum_{i,j : X_{ij} \in Y} (\hat{X}_{ij} - X_{ij})^2} 
   \]

   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,
5.2 Results

In our experiments, the selected number of core users is basically consistent with that of neighbors obtained according to the similarity of Kilani et al. (2018).

First, we will explain the influence of $k$ and $K$ on the four evaluation metrics, i.e., MAE, RMSE, Precision and Recall. Although different $k$ and $K$ will affect the values, we find it is more stable when $k = 8$ and $K = 77$ after running all the experiments. In the following, we will present the visual changing results of the four evaluation metrics with $k$ and $K$ through Figs. 2, 3, 4, 5 and 6.

Figure 2 presents the results of MAE and RMSE with $K$ when fixing $k = 8$. In this figure, MAE and RMSE are basically smallest for $K = 77$. After that, there is a slightly decreases for MAE as $K$ increases from 299 to 402 and for RMSE as $K$ increases from 299 to 507, and nearly increases for other $K$. During the 50 runs of experiments, although the value fluctuates with $K$, they always keep the smallest around $K = 77$. These indicate the recommendation results are closely related to neighbor users with greater similarity degree which considers the movie’s intrinsic attributes. Figure 4 presents the results of precision and recall with $K$ when fixing $k = 8$. The precision and recall reach the maximum when $K = 77$, after that the values almost decrease with $K$ increases and then increase.

Fig. 3 The results of MAE, RMSE with different $k$ when $K = 77$

Fig. 4 The results of precision, recall with different $K$ when $k = 8$

Fig. 5 The results of precision, recall with different $k$ when $K = 77$

Fig. 6 The results of MAE, RMSE with $\beta$

\[
\text{Precision} = \frac{\sum |R_i \cap T_i|}{\sum |R_i|}
\]

\[
\text{Recall} = \frac{\sum |R_i \cap T_i|}{\sum |T_i|}
\]

where $R_i$ is the recommendation lists of user $i$ in training set, $T_i$ is the recommendation lists in testing set.
after $K = 507$. This phenomenon indicates being greedy in user neighbors may possibly improve the recommendation quality, however it will increase the computation complexity. The results are consistent with that of Fig. 2. Therefore, our method CU-FHR appropriately reflects the recommendation notion and can be applicable.

Figure 3 (Fig. 5) presents the variation of MAE and RMSE (precision and recall) with $k$ when fixing $K = 77$. In this figure, MAE and RMSE (precision and recall) reach the smallest (largest) 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 cannot produce better effect. The fact shows that the decision-makers should focus on the main features of users and movies when making recommendation, which is accordance with decision-making preference.

Second, in model (10), $\beta$ 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 cannot effectively suppress Gaussian noise. $\beta$ 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 $\beta$ on MAE and RMSE, we make some experiments for fixed $k = 8$, $K = 77$. The results are presented in Fig. 6.

Figure 6 presents the results of MAE and RMSE on the regularization tradeoff parameter $\beta$. From the figure, we see that the results were not influenced greatly when $\beta$ varied from 0 to 0.9, but the results are better when $\beta \in (0, 0.1]$ which is consistent with the discussions in [16].

Moreover, we conduct some ablation experiments of CU-FHR. First, CU-FR is a neighbor-based CF method, we discuss the variation of precision and recall with different $K$ shown in Fig. 7. We can see that the precision and recall increase with the number of neighbors (core users) which accords with the basic decision-making notion of CF. Second, for CU-HR, we discuss the variation of MAE and RMSE, precision and recall with different $K$ when fixing
In this paper, we present a new hybrid recommendation approach, called CU-FHR. This method improves matrix factorization model by selecting the core users and is applicable to movie recommendation. Unlike previous CF methods, CU-FHR innovatively incorporates the content feature of items described by fuzzy sets into the user similarity computation. The improved similarity can have a positive influence on the rating because it can describe the ambiguity of the prototype of movies. So it serves as a baseline standards of computing importance of a user for the core users selection. One advantage of CU-FHR is that it uses core users rather than all the users to establish a matrix factorization model. This will greatly reduce the computation complexity. Furthermore, the experimental results from a comparison with two other recommendation methods show that CU-FHR achieved better prediction accuracy. It 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 is an effective and feasible items recommendation method with wide applications in big data era.

In future, more research 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).

Funding The authors have not disclosed any funding.

Data availability Enquiries about data availability should be directed to the authors.

Declarations

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

Aguilara J, Valdiviezo-Dúzb P, Riofrio G (2017) A general framework for intelligent recommender systems. Appl Comput Inform 13:147–160. https://doi.org/10.1016/j.aci.2016.08.002

Ali M, Jung LT, Abdel-Aty AH, Abubakar MY, Elhoseny M, Ali I (2020) Semantic-k-NN algorithm: an enhanced version of traditional k-NN algorithm. Expert Syst Appl 15:1113374. https://doi.org/10.1016/j.eswa.2020.113374

Anwaar F, Iftaf N, Afzal H, Nawaz R (2018) HRS-CE: a hybrid framework to integrate content embeddings in recommender systems for cold start items. J Comput Sci 29:9–18. https://doi.org/10.1016/j.jocs.2018.09.008

Ayub M, Ali Ghazanfar M, Mehmood Z, Ayloubi KH, Alfakeeh AS (2020) Unifying user similarity and social trust to generate powerful recommendations for smart cities using collaborating filtering-based recommender systems. Soft Comput 24:11071–11094. https://doi.org/10.1007/s00500-019-04588-x

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

Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowl-Based Syst 46:109–132. https://doi.org/10.1016/j.knosys.2013.03.012

Burke R (2002) Hybrid recommender systems: survey and experiments. User Model User-Adap Inter 12(4):331–370. https://doi.org/10.1023/A:10212407030564

Candès EJ, Recht B (2009) Exact matrix completion via convex optimization. Found Comput Math 9:717–772. https://doi.org/10.1007/s10208-009-9045-5
Hybrid recommender system with core users selection

Choudhary N, Minz S, Bharadwaj KK (2020) Circle-based group recommendation in social networks. Soft Comput. https://doi.org/10.1007/s00500-020-05356-y

Del Corso GM, Romani F (2019) Adaptive nonnegative matrix factorization and measure comparisons for recommender systems. Appl Math Comput 354:164–179. https://doi.org/10.1016/j.amc.2019.01.047

Duma M, Twala B (2018) Optimising latent features using artificial immune system in collaborative filtering for recommender systems. Appl Soft Comput 71:183–198. https://doi.org/10.1016/j.asoc.2018.07.001

Guan N, Tao D, Luo Z, Yuan B (2012) Online nonnegative matrix factorization with robust stochastic approximation. IEEE Trans Neural Netw Learn Syst 23(7):1087–1099. https://doi.org/10.1109/TNNLS.2012.219782

Holland J (1975) Adaptation in nature and artificial systems. University of Michigan Press, Ann Arbor

Kermany NR, Alizadeh SH (2017) A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques. Electron Commer Res Appl 21:50–64. https://doi.org/10.1016/j.e-crap.2016.12.005

Kilani Y, Otoom AF, Alsaari A, Almajahey M (2018) A genetic algorithms-based hybrid recommender system of matrix factorization and neighborhood-based techniques. J Comput Sci 28:78–93. https://doi.org/10.1016/j.jocs.2018.08.007

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

Krasnoshchok O, Lamo Y (2014) Extended content-boosted matrix factorization algorithm for recommender systems. Int Conf Knowl-Based Intell Inf Eng Syst-Proced Comput Sci 35:417–426. https://doi.org/10.1016/j.procs.2014.08.122

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

Liu B, Xiong H, Spiros P, Fu Y, Yao Z (2015) A general geographical probabilistic factor model for pointof interest interest recommendation. IEEE Trans Knowl Data Eng 27(5):1167–1179. https://doi.org/10.1109/TKDE.2014.2362525

Löpp B, Donkers T, Kleemann T, Ziegler J (2019) Interactive recommending with tag-enhanced matrix factorization (TagMF). Int J Hum Comput Stud 121:21–41. https://doi.org/10.1016/j.ijhcs.2018.05.002

Lu J, Wu D, Mao M, Wang W, Zhang G (2015) Recommender system application developments: a survey. Decis Support Syst 74:12–32. https://doi.org/10.1016/j.dss.2015.03.008

Mao M, Lu J, Zhang G, Zhang J (2017) Multirelational social recommendations via multitag ranking. IEEE Trans Cybern 47(12):4049–4061. https://doi.org/10.1109/TCYB.2016.2595620

Martín-Vicente MI, Gil-Solla A, Ramis-Cabrér M, Pazos-Arias José J, Blanco-Fernández Y, López-Nores M (2014) A semantic approach to improve neighborhood formation in collaborative recommender systems. Expert Syst Appl 41(17):7776–7788. https://doi.org/10.1016/j.eswa.2014.06.038

Navgaran DZ, Moradi P, Akhlaghian F (2013) Evolutionary based matrix factorization method for collaborative filtering systems. In: 2013 21st Iranian conference on electrical engineering, ICEE 2013. Doi: https://doi.org/10.1109/IranianCEE.2013.6599844

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

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.