Heliyon 9 (2023) e18183

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Adaptive genetic algorithm for user preference discovery in multi-criteria recommender systems

Mohammed Wasid [a,*], Rashid Ali [a,b], Sana Shahab [c]

[a] Interdisciplinary Centre for Artificial Intelligence, Aligarh Muslim University, Aligarh, India
[b] Department of Computer Engineering, Aligarh Muslim University, Aligarh, India
[c] Department of Business Administration, College of Business Administration, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

ARTICLE INFO

Keywords:
Collaborative filtering
Normalized rating count
Multi-criteria decision making
Genetic algorithms
Recommender systems

ABSTRACT

A Multi-Criteria Recommender System (MCRS) represents users' preferences on several factors of products and utilizes these preferences while making product recommendations. In recent studies, MCRS has demonstrated the potential of applying Multi-Criteria Decision Making methods to make effective recommendations in several application domains. However, eliciting actual user preferences is still a major challenge in MCRS since we have many criteria for each product. Therefore, this paper proposes a three-phase adaptive genetic algorithm-based approach to discover user preferences in MCRS. Initially, we build a model by assigning weights to multi-criteria features and then learn the preferences on each criteria during similarity computation among users through a genetic algorithm. This allows us to know the actual preference of the user on each criteria and find other like-minded users for decision making. Finally, products are recommended after making predictions. The comparative results demonstrate that the proposed genetic algorithm based approach outperforms both multi-criteria and single criteria based recommender systems on the Yahoo! Movies dataset based on various evaluation measures.

1. Introduction

The World Wide Web has become a source of abundant information, accessed by a huge number of users in the form of text, web pages, pictures, videos, and more. The rise of e-commerce and social network sites has further increased the volume of information available over the web, enabling users to easily access, create, share, and upload diverse information. Numerous e-commerce websites sell millions of products, making it challenging for users to choose the desired one among so many options. Due to the vast amount of information, finding what we desire that most closely matches our preferences has become a significant challenge, referred to as information overload [1]. Recent developments in information filtering methods have made it possible to make a vast amount of information available to Internet users. These filtering methods are used as a tool to tame information overload and sift through the vast amount of available information to find relevant content. Recommender systems (RS) can address the information overload problem and be used as a decision-making tool by suggesting the most suitable and valuable products to users [2]. For example, when we search for a product on Amazon.com, it recommends suitable products so that we can decide which product to buy. Thus, RSs process and convert the vast available data into user-oriented information to make expert-level decisions [3]. With intense growth in interest, RSs have been deployed on hundreds of different sites, serving millions of users. Hence, they enable sellers to increase their

* Corresponding author.
E-mail address: erwasid@gmail.com (M. Wasid).

https://doi.org/10.1016/j.heliyon.2023.e18183
Received 20 December 2022; Received in revised form 10 July 2023; Accepted 11 July 2023
Available online 12 July 2023

revenue by selling a large number of products and satisfy users who want to buy desired products quickly. Fig. 1 shows a general architecture depicting the interaction between a user and RS.

RSs can be categorized into two types based on the rating space it operates in: two-dimensional and multidimensional. A two-dimensional or single-criteria RS (SCRS) is a traditional RS that operates on the user-item rating space. These SCRS typically experience a number of issues, including sparsity and cold startness. On the other hand, when additional user-item attributes or features are added to this rating space, multidimensionality becomes an issue. These additional features are recognized as 'side information' and systems that support such features are known as multidimensional RSs because they don't have a fixed number of dimensions, unlike traditional RSs. Multidimensionality is one of the severe problems in RSs [4].

A two-dimensional RS can be extended into a Multi-Criteria Recommender System (MCRS) by appending various criteria ratings to the overall rating space [5]. MCRS utilizes item attributes to make recommendations for users according to their multi-criteria decision making. By identifying user preferences based on various aspects of an item, MCRS can increase user satisfaction. However, there are numerous instances in real-world scenarios where users behave differently based on various attributes [2,6]. For example, service, cleanliness, location, and food can all be criteria for a restaurant, whereas a movie's criteria can include visual, story, direction, and acting. Thus, some users may choose a recommendation based on the quality of the food instead of cleanliness of a particular restaurant [7].

In this direction, user interests should be learned and prioritized to reflect their actual preferences. Getting the exact user preference is challenging as each user has different preferences for various item features. Therefore, researchers are exploring ways to combine RSs with other emerging areas to identify actual user preferences for accurate decision-making [8,9]. The use of Genetic Algorithms (GA) to solve optimization issues is a new trend in many application domains [10,11]. GA has previously been used to find the weights for user-item related features that signify their priority within the user profile [6,12]. In MCRS, GA has been used as a weighted aggregator after learning the weights of each criteria for every user [2,13]. Similarly, heuristic approaches also aggregate the user's preferences based on various criteria. Aggregating criteria ratings is a crucial step, and if not performed properly, it may lower the value of having multiple factors of an item [14]. Therefore, to develop a truly personalized MCRS, these criteria need to be efficiently incorporated and their individual weights must be learned and adjusted for actual user preference discovery in MCRS.

In this paper, we propose an adaptive GA-based MCRS to discover actual preferences of the users. To avoid the issue of multidimensionality, we partitioned the entire database into single-criteria and multi-criteria ratings. We processed both types of ratings separately using various similarity measures and a normalized rating count approach to determine user preferences at single-criteria and multi-criteria levels. Finally, these preferences are combined to obtain the overall preference of each user. GA is employed to adapt the criteria weights to capture user priorities for each criteria.

The following is the list of main contributions of this paper:

- An efficient framework is developed that enables multidimensional RSs, handles multidimensionality, and enhances the predictive and recommendation performance of MCRS.
- A normalized rating count method is utilized for incorporating the multi-criteria ratings into SCRS without multidimensionality issues.
- A GA-NRC-CRS approach to MCRS is developed using GA.

The paper is organized as follows. The background and related research for this paper are presented in Section 2 and the proposed GA-based approach is presented in Section 3. Section 4 presents the experiments and results. Finally, Section 5 concludes the paper.

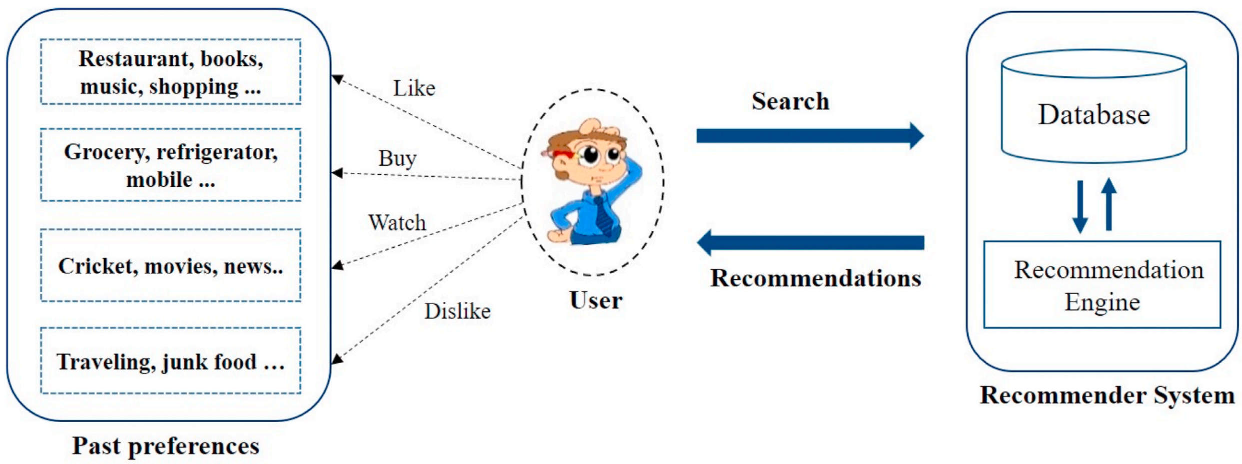

Fig. 1. General architecture of a recommender system.

2. Background and related work

This section provides an overview of the background and related works relevant to the work done in this paper.

2.1. Background

2.1.1. Recommender systems

RSs have emerged as a web personalization tool that assists online users by alleviating the problem of information overload. An effective RS always ensures that it captures the actual preferences of users and recommends only those things that the user actually wants. One of the primary concerns in developing a successful RS is the correct choice of a relevant recommendation technique [15]. This choice determines the methodology used by the RS to predict ratings for users on new items. Collaborative Filtering (CF) is a recommendation technique that is most commonly used in the development of RSs. CF is relies on the idea that user preferences are stable and that 'people who have previously agreed tend to do so again in the future' [3]. Therefore, the core concept of this technique is to provide recommendations depending on the joint preferences of users who share similar interests [16]. The source for finding users with similar interests is their common historical preferences and similar choices. Generally, the CF technique is classified into two categories known as memory-based and model-based techniques [3]. The memory-based technique deals with the numerical ratings of the experienced items, while the model-based technique creates a model using machine learning techniques to make personalized recommendations.

The accuracy of traditional CF-based RSs has been improved by many researchers by incorporating multi-criteria ratings [13,17, 18]. MCRSs have been used in a variety of research domains such as tourism [18], service [7,17], movies [19,20], health systems [4], and so on. Although the traditional CF system has a straightforward method for measuring similarity, MCRSs lack an absolute method for combining criteria ratings when computing similarity. Some efforts have been made to incorporate these additional ratings into the traditional RS without dimensionality expansion. For instance, a multi-criteria CF technique was developed that clustered the users according to their preferences on each criteria [21]. Their proposed approach used criteria ratings as clustering parameters instead of an additional feature. They further extended their proposed approach by fuzzifying the multi-criteria features before clustering for better recommendations [5]. Similarly, Nilashi et al. [4] created clusters and applied a dimensionality reduction method using higher-order singular value decomposition to identify similar users and items within each cluster in MCRS.

2.1.2. Genetic algorithm

In the past, researchers have used several evolutionary techniques for developing effective RSs [9]. GA is a method of computational optimization that draws inspiration from genetics and natural selection. It is a kind of evolutionary algorithm that looks for the best solutions to a problem by looking at the population as a whole. The main components of a GA include [10]:

- **Initialization:** A set of initial solutions is randomly generated to form the initial population.
- **Fitness Function:** This is defined to evaluate the quality of each solution in the population. The fitness function guides the selection of the most promising solutions for further optimization.
- **Selection:** The selection process involves choosing the best individuals from the population based on their fitness scores. Various selection techniques can be used, such as roulette wheel selection or tournament selection.
- **Crossover:** This genetic operator is used to change the chromosomes from one generation to another. The process of crossover is similar to biological reproduction as it takes more than one parent solution and produces a child solution (new individual) from them. The application of the crossover operator depends on the representation used for the chromosome. Given real-valued parents $\alpha_{1=\{}\alpha_{11}, \alpha_{12}, ..., \alpha_{1n\}}$ and $\alpha_{2=\{}\alpha_{21}, \alpha_{22}, ..., \alpha_{2n\}}$ from the population, single-point crossover will generate two offspring as $\alpha_{1'=\{}\alpha_{11}, \alpha_{12}, ..., \alpha_{2n\}}$ and $\alpha_{2'=\{}\alpha_{21}, \alpha_{22}, ..., \alpha_{1n\}}$. The process of single-point crossover used in our work is graphically represented in Fig. 2.

•**Mutation:** It is responsible for ensuring that the population's genetic variety is preserved from one generation to the next. It resembles biological mutations in many ways. Mutation performs an arbitrary alteration of a single chromosome to produce a new individual. Alteration means that it modifies one or more gene values in the chromosome, as shown in Fig. 3. The solution may vary completely from the earlier solution. Hence, GA can achieve a better solution by using this operator.

•**Termination:** The algorithm terminates when a stopping criteria is met. This could be a maximum number of generations, reaching a satisfactory fitness level, or exceeding a computation time limit.

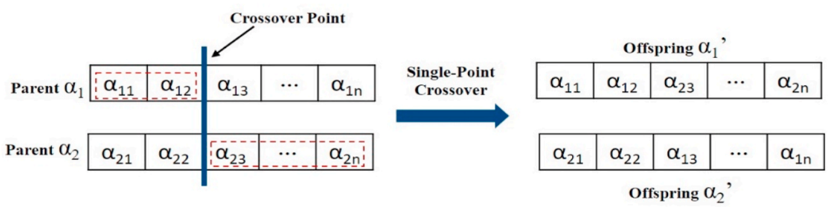

Fig. 2. Single-point crossover operator.

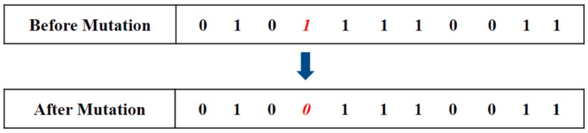

Fig. 3. Mutation operator in genetic algorithm.

By iteratively applying selection, crossover, and mutation operators, GA can efficiently explore a large search space and converge toward optimal solutions.

2.2. Related works

Multi-Criteria Collaborative Filtering (MCCF) is a type of neighborhood-based MCRS that incorporates traditional CF techniques in its recommendation process [33]. MCCF focuses on leveraging the similarities among users or items based on multiple criteria to make personalized recommendations. Similar to the CF technique, the MCRS has been classified as memory-based and model-based approaches [2]. The model-based approach involves developing a system that identifies a function that determines how single criteria and multi-criteria ratings relate to one another [14]. On the other hand, the memory-based approach expands traditional RS by identifying user similarities based on different criteria and aggregating these similarities using different methods. Calculating the weighted aggregation of multi-criteria ratings is one of the challenging tasks involved in producing recommendations for users. Therefore, different weight learning algorithms such as Genetic Programming (GP) [19], GA [20,22], particle swarm optimization [23], bacterial foraging optimization [24], etc. are used for learning the weights.

For instance, GA was used to capture the ideal weights for various features [22]. The proposed approach learned a set of weights of various features for each user, allowing the prediction of unseen item ratings based on specific criteria. Gupta and Kant [19] utilized the GP algorithm to aggregate criteria ratings after learning optimal weights for various criteria. Similarly, GA has also emerged as a popular optimization technique for MCRS. GA can be employed for feature weighting in MCRS to capture the importance of each criteria to know the exact preferences of the users [25]. The GA actually learns the relative preference of different features of different users and predicts the ratings of items based on these preferences. Hassan and Hamada [26] employed an adaptive GA that dynamically updates the crossover and mutation rates, modeling users' preferences for various criteria ratings. The adaptive GA determines a unique optimal weight vector for each user, which is used to make recommendations.

Another approach employed by Gupta and Kant [20] involved using an item's credibility score, which was aggregated by taking into account credibility scores on various criteria. In order to aggregate the credibility score, GA was used to capture the proper weights for each criteria for each user. Authors in Refs. [27,28] developed an aggregation function-based approach by employing an adaptive GA to effectively integrate the criteria ratings and improve the accuracy of the MCRS. The aggregation function was applied based on an adaptive GA and was developed to improve the MCRS extrapolation capabilities. Similarly, Hamada et al. [29] proposed adaptive GA and fuzzy logic-based recommendation models to integrate the multi-criteria ratings into the traditional RS. Results inferred that their developed models gave better results in comparison to traditional CF-based RS. In another study, Hassan and Hamada [30] developed three different recommendation models using three variants of GA to improve the accuracy of MCRS. The adaptive and multi-heuristic GA were designed by incorporating appropriate mathematical changes in the standard GA. Results showed that the GA-based approaches gave better performance and the multi-heuristic GA-based approach outperformed other approaches. On the other hand, GP was employed in MCRS to learn criteria weights [14,19]. Where GP was used to learn a function to transform user preferences into an aggregate of criteria ratings. It was employed to learn a function for transforming user preferences for aggregating criteria ratings. Gupta and Kant [11] conducted a comparative analysis of both GA and GP in MCRS and empirically showed that both GA and GP gave close or equivalent results in the same computational environment. GA was found to have an advantage over GP due to its smaller search space, indicating no significant advantages of using GP over GA in MCRS. Taking this into consideration, we have chosen to utilize GA as the basis for our proposed approach.

Upon reviewing the existing approaches, we have identified that many of these approaches rely on a function to establish the relationship between single criteria and multi-criteria ratings, or they employ GA to incorporate and aggregate various criteria ratings. While these methods may be effective in incorporating multi-criteria ratings, they often overlook the importance of maintaining the distinct preferences associated with each item. In addition, it is worth noting that most of these approaches primarily focused on testing their proposed methods using either Predictive accuracy measures (such as MAE, RMSE, and Coverage) or Recommendation accuracy measures (such as Precision, Recall, and F-measure). Therefore, a fine-tuned system needs to be developed that can address the challenge of multidimensionality while preserving these diverse preferences and evaluate it across multiple evaluation measures to provide a more comprehensive and robust assessment of its effectiveness. Our proposed adaptive GA approach aims to determine optimal weights for different criteria without aggregating them or establishing any relationship between ratings. By doing so, we ensure that the multiple preferences associated with each item are retained and considered in the recommendation process. Furthermore, we have taken a comprehensive approach by evaluating our proposed approach on both predictive and recommendation accuracy measures. This allows us to gain a deeper understanding of its capabilities in terms of accurately predicting user preferences and providing high-quality recommendations.

3. Genetic algorithm for user preference discovery

In this section, we discuss the proposed GA based MCRS approach for user preference discovery. Fig. 4 presents an abstract view of the major components of the proposed approach. In the below subsections, we discuss each component of the proposed approach in detail.

3.1. User preference elicitation at single-criteria level

This step involves finding user preferences using single-criteria ratings extracted from historical user ratings. These single-criteria ratings are used to implement a classical CF approach. Once user preferences on various items are found, the next step is to calculate the similarity among the user rating preferences. There are various similarity measures available for computing the similarity among users [16,31]. Following are some of the popular similarity measures that we have used in this research.

- **Pearson Correlation (PC)** is a correlation based most extensively implemented similarity measure utilized to determine user similarity. Equation (1) is used to compute the PC similarity S between two user u and user v.

$$S_{uv}^{pc} = \frac{\sum_{i \in A_{uv}} \left(r_{u,i} - \bar{r}_u\right)\left(r_{v,i} - \bar{r}_v\right)}{\sqrt{\sum_{i \in A_{uv}} \left(r_{u,i} - \bar{r}_u\right)^2} \sqrt{\sum_{i \in A_{uv}} \left(r_{v,i} - \bar{r}_v\right)^2}} \tag{1}$$

where $r_{u,i}$ is the numerical rating at item i provided by user u and $\bar{r}_u$ denotes the mean of ratings of user u. A_{uv} is the collection of items that user u and user v both co-rated.

- **Cosine-based Similarity (CS)** calculates the cosine angle between two users to assess the similarity among them, CS is computed using equation (2).

$$S_{uv}^{cs} = \frac{\sum_{i \in A_{uv}} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i \in A_{uv}} r_{u,i}^2} \sqrt{\sum_{i \in A_{uv}} r_{v,i}^2}}, \tag{2}$$

•**Extended Jaccard Coefficient (JC)** is an extended method of conventional Jaccard coefficient. The JC similarity S between two user u and user v is computed similar to the method of cosine similarity, as shown in equation (3).

$$S_{uv}^{jc} = \frac{\sum_{i \in A_{uv}} r_{u,i} \cdot r_{v,i}}{\sum_{i \in A_{uv}} r_{u,i}^2 + \sum_{i \in A_{uv}} r_{v,i}^2 - \sum_{i \in A_{uv}} r_{u,i} \cdot r_{v,i}}, \tag{3}$$

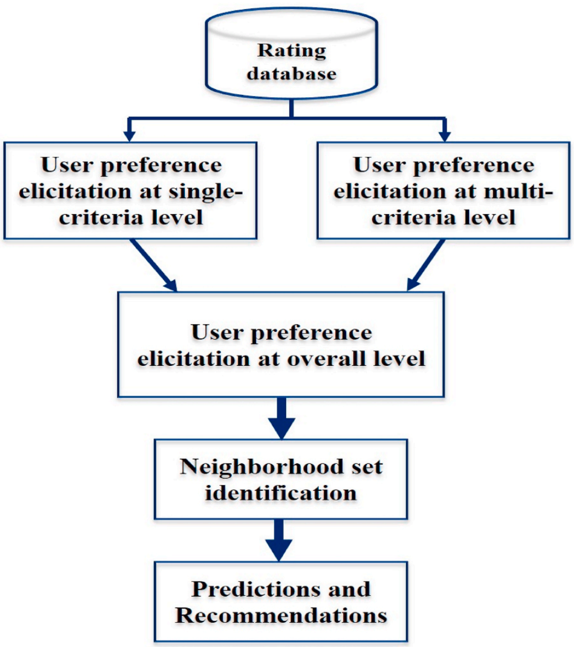

Fig. 4. An abstract view of the major components of the proposed approach.

- **Modified Mahalanobis Distance (MMD)** is the extended version of conventional Mahalanobis distance [21]. This measure determines the inverse of the variance-covariance matrix for database users. The MMD distance D between user u and user v is computed using equation (4).

$$D_{uv}^{mmd} = \frac{\sum_{i \varepsilon A_{uv}} \sqrt{\left(\frac{r_{u,i}-\overline{u}}{\sigma_u}\right)^2 + \left[\left\{\left(\frac{r_{v,i}-\overline{v}}{\sigma_v}\right) - \rho_{uv}\left(\frac{r_{u,i}-\overline{u}}{\sigma_u}\right)\right\}\frac{1}{\sqrt{1-\rho_{uv}{}^2}}\right]^2}}{|A_{uv}|}, \tag{4}$$

where σ_u^2 and σ_v^2 depict the variances of the values of user u and user v, respectively. $\rho_{uv}\sigma_u\sigma_v$ depict the covariance between users.

- **OS measure** gives better performance than some of the classical similarity measures [31]. The left part of equation (5) considers the number of co-rated items between users while the right part uses the rating information among the users.

$$S_{uv}^{os} = \exp\left(-\frac{N - |I_u \cap I_v|}{N}\right) \cdot \frac{\sum_{i \in I} \exp\left(-\frac{|r_{u,i}-r_{v,i}|}{\max\left(r_{u,i}-r_{v,i}\right)}\right)}{|I_u \cap I_v|} \tag{5}$$

where N is the cardinality of items, I_u and I_v are the sets of items co-rated by user u and v, $\max\{r_{u,i}, r_{v,i}\}$ is the maximum rating in $r_{u,i}$ and $r_{v,i}$.

- **Common rating weight similarity measure (CRS)** is developed by combining similarity measure with a weighting method [23], as shown in equation (6). This measure gives weights to the similarity obtained among users based on the size of the co-ratings among them.

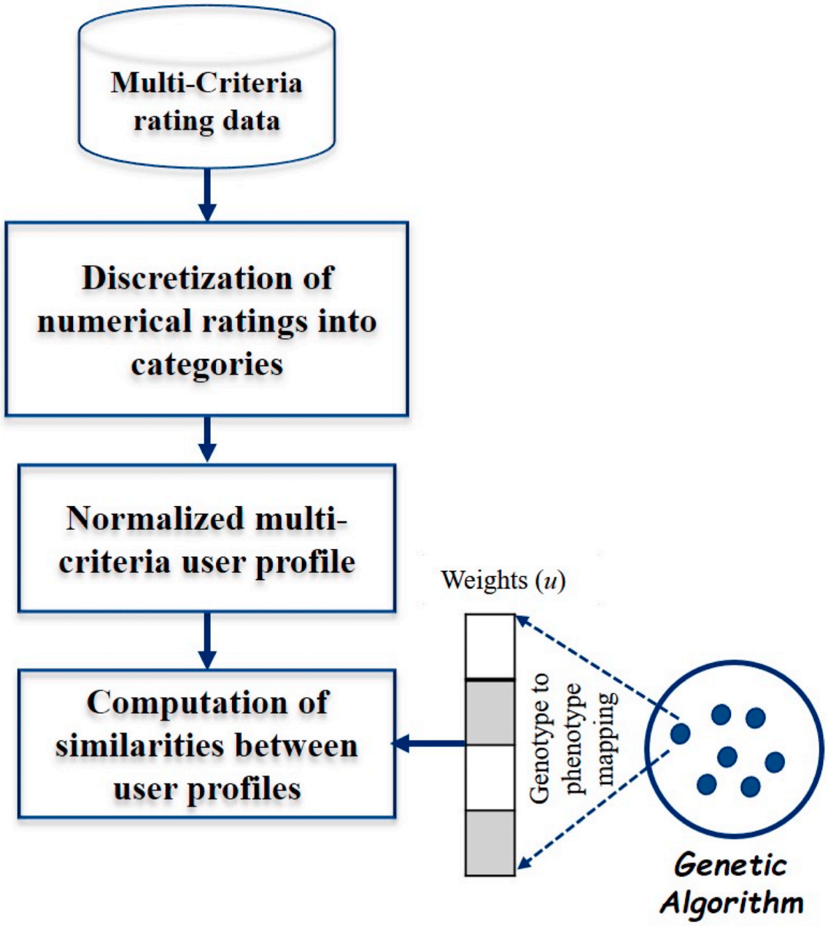

Fig. 5. Steps of user preference elicitation at multi-criteria level.

$$S_{uv}^{crs} = \left(\frac{|I_u \cap I_v|}{(|I_u| + |I_v|) - (|I_u \cap I_v|)} \cdot \frac{1}{1 + \exp\left(- \frac{|I_u \cap I_v|}{df} \right)} \right) \tag{6}$$

where *df* is calculated based on the level of sparsity present in the database.

After computing the similarity among the users, the neighborhood set for an active user can be formed by selecting the top-*k* most similar users who have higher similarity than a certain threshold. Lastly, these neighbors are used to make predictions and recommendations to the active user. These are the three phases for implementing traditional CF using single-criteria ratings. Such CF-based RS approaches are termed based on the similarity measure they have used (for example, SCPC is a Pearson Correlation-based SCRS, SCCS is a Cosine similarity-based SCRS, etc.).

In our proposed approach, the similarity obtained at the single-criteria level is fused with the similarity obtained at the multi-criteria level to find the user preferences at the overall level.

3.2. User preference elicitation at multi-criteria level

In this step, we use multi-criteria ratings for identifying user preferences at multi-criteria level. The flow for elicitation of the user preferences through multi-criteria ratings is illustrated in Fig. 5.

First of all, discretization is applied in order to convert the multi-criteria ratings into different categories. For discretizing each criteria rating from 1-bad to 13-excellent, five different groups are created [*1→ bad(1,2,3), 2→ average(4,5), 3→ good(6,7,8), 4→ verygood(9,10), 5→ excellent(11,12,13)*], such that, the average of a group [*2; 4.5; 7; 9.5; 12*] have same (two and half) step size. A sample of resulting ratings of user u_1 is shown in Table 1.

Now that the categorical values for every single category in the multi-criteria rating matrix have been determined, the following equation (7) is used to create a normalized multi-criteria user profile.

$$nrc(cr, at) = \frac{\sum_{p \in at}^{|n|} f_{cr,p}}{|n|} \tag{7}$$

where attribute *at* of criteria *cr* is used to calculate normalized rating count *nrc(cr,at)*. The normalization is done by dividing the repeat count $f_{cr,p}$ of attribute *at* in criteria *cr* with total number of items *n*. This process is repeated for all five groups for *c* criteria to get the normalized profile of each user, as shown in Table 2.

The similarity among these normalized ratings of users is computed using the normalized rating count measure, as shown in equation (8).

$$S_{uv}^{nrc} = \frac{c}{c + \sqrt{\sum_{cr=1}^{c} \sum_{f=1}^{5} \left(u_{cr,f} - v_{cr,f} \right)^2}} \tag{8}$$

where *c* is the cardinality of criteria and $u_{cr,f}$ depicts the feature *f* of criteria *cr* for user *u*.

In real-life scenario, it is quite possible that different criteria may be given different weights which we call as criteria weights. By imposing weights to criteria in normalized rating count measure, learning algorithms can be used to find these weights. For these approaches, the updated equation, after appending weights, can be rewritten as a weighted normalized rating count (WNRC) measure as shown in the following equation (9).

$$S_{uv}^{wnrc} = \frac{c}{c + \sqrt{\sum_{cr=1}^{c} w_{cr} \times \sum_{f=1}^{5} \left(u_{cr,f} - v_{cr,f} \right)^2}} \tag{9}$$

where w_{cr} is the weight for the cr^{th} criteria. These weights need to be identified and fine-tuned to capture the priorities of each user in order to develop an effective recommendation system. We have employed GA that adapts the criteria weights based on the similarities computed among the users. We discuss the exact procedure of learning weights in the following subsection.

Table 1
Discretization of multi-criteria ratings.

u_1	Cr_1	Cr_2	Cr_3	.	Cr_c
I_1	bad	good	average	.	excellent
I_2	good	average	average	.	good
…	…	…	…		…
I_m	bad	very good	very good	.	excellent

Table 2
Normalized multi-criteria rating count profile of *m* users.

	Cr1					Cr2					Cr3	Cr_c				
	bad	*average*	*good*	*very good*	*excellent*	*bad*	*average*	*good*	*very good*	*excellent*		*bad*	*average*	*good*	*very good*	*excellent*
$\boldsymbol{u_1}$	0.14	0.04	0.14	0.33	0.33	0.10	0.02	0.20	0.18	0.47	…	0.08	0.04	0.29	0.16	0.41
$\boldsymbol{u_2}$	0.10	0.03	0.03	0.17	0.64	0.10	0.03	0.035	0.21	0.60		0.10	0.03	0.03	0.21	0.60
			.					.			.			.		
$\boldsymbol{u_m}$	0.07	0.07	0.18	0.37	0.29	0.07	0.03	0.33	0.18	0.37		0.07	0.03	0.18	0.29	0.40

3.3. Multi-criteria preference elicitation using genetic algorithm

The complete process of preference elicitation and weight learning using GA is presented in Algorithm 1.

Algorithm 1 Learning weights for multi-criteria ratings using genetic algorithm.

Step 1: [*Initialization*]
Create initial population (*Section 3.3.1*).
Step 2: [*Fitness Assessment*]
Calculate fitness scores for each individual using fitness function (*Section 3.3.2*).
Step 3: [*Update population*]
Create new population using crossover and mutation operators of GA.
Step 4: [*Termination*]
a. Iteratively process weight learning until termination condition is reached.
b. if termination condition met, then the process terminates and returns a set of optimal criteria weights, otherwise go to step 2.

3.3.1. Initial feature weights for genetic algorithm

GA adapts the weights of the criteria to determine the user priorities for various criteria. The criteria weights of user u are presented as a set of weights, $weight(u) = (w_1, w_2, \ldots, w_n)$, where n shows the number of criteria. The genotype of w_i is a real-valued number. When a feature's weight is set to 0, that criteria is ignored. This enables criteria selection to be user-adaptive according to their preference. In order to begin (initialize) the GA, there are four steps needed to accomplish. Fig. 6 shows all four steps for generating the initial weights for GA.

By taking an example of c criteria, the working of initializing GA by using initial random values in each step is demonstrated through Table 3.

3.3.2. Fitness function

Designing a suitable fitness function for GA applications can be typical, especially in the case of the application discussed in this work. This is because each set of weights in the GA population must be used for similarity computation, requiring a new run of the system on the entire set of data to determine the fitness of each newly acquired set of weights [12]. For the active user, a good set of weights should generate a good neighborhood set of users, resulting in good recommendations. Consider treating the issue as a supervised learning task when designing the fitness function in order to address this. This entails running the system, calculating the predicted ratings for each item in the training ratings set to establish the fitness value for the learned set of weights, and then randomly dividing the entire active user's ratings into training and testing rating sets.

Therefore, an individual's fitness can be assessed by the absolute difference between the actual value and predicted values for all items present in the training rating set. The fitness value in GA based approach can be calculated on the basis of following equation

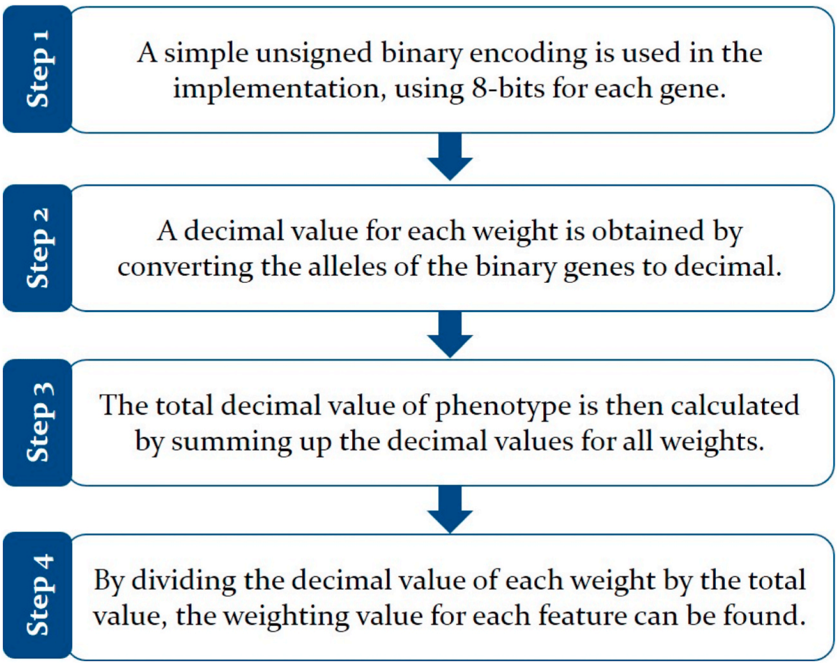

Fig. 6. Steps for generating initial weights for genetic algorithm.

Table 3
Example for generating initial weights for genetic algorithm.

Step no.	Criteria Cr_1	Criteria Cr_2	Criteria Cr_3	…	Criteria Cr_c	Remarks
1	01011001	10010000	00000101		01100100	Random binary numbers
2	89	144	5	.	100	Let the sum is 338
4	0.2633	0.4260	0.0148		0.2959	Sum is equal to 1

(10).

$$fit = \frac{1}{t}\sum_{i=1}^{t}|r_i - p_i|, \tag{10}$$

where t represents the number of items in training set, r_i represents the actual value and the predicted value of item i is represented by p_i.

3.4. User preference elicitation at overall level

This step combines the similarities calculated from the single criteria and multi-criteria levels. The purpose of this combination is to discover user preferences at the system level and identify the most accurate neighborhood set. Following equation (11) combines the 'single criteria similarity using CRS measure' and 'multi-criteria similarity using WNRC measure' to find the overall similarity between user u and user v.

$$S_{uv}^{overall} = S_{uv}^{crs} \cdot S_{uv}^{wnrc} \tag{11}$$

The most similar users also known as 'neighbors', can be identified using the above equation (11). These neighbors are used for predicting the unseen items of the active user in the prediction and recommendation phase of the CF based recommendation process.

3.5. Predictions & recommendations

The Resnick's prediction formula (also known as weighted sum) is the most widely used aggregation function and is calculated using equation (12). The benefit of this formula is that it compensates the rating scale variations in case of users varying rating scale. This enables the predicted ratings to fall within the mean rating of a given active user [6].

$$p_{u,i} = \overline{r}_u + nf\sum_{v\in k} s_{uv} \times \left(r_{v,i} - \overline{r}_v\right), \tag{12}$$

where s_{uv} is the similarity among the users u and v, k represents the users who rated item i, and normalization factor nf is measured using the following equation (13).

$$nf = \frac{1}{\sum_{v\in k}|s_{uv}|}, \tag{13}$$

4. Experiments and results analysis

This section discusses the experiments and results in every aspect. We discuss dataset selected for the experiments, evaluation measures, experiments and results analysis in detail.

4.1. Experimental settings

Yahoo! Movies dataset [32] is used for conducting the experiments. The dataset contains ratings on a scale from 1 to 13, which correspond to the lowest and highest preferences of the users. To pre-process the dataset with substantial ratings, we recorded those users who gave ratings to at least 20 movies and discarded movies with zero ratings. After this refinement, the pre-processed dataset contains 484 users, 945 movies, and 19050 ratings. Each movie has one single criteria and four multi-criteria ratings. We normalize the single criteria ratings on the numerical scale of 1–5 using the rating groups formed during the discretization process of multi-criteria ratings. The experiments are performed by varying different training-testing set split ratios from 60% to 90% for each user. The training data is used for model building while testing data for evaluating the model. The experiments are also performed with different neighborhood set sizes such that k varies from 10 to 70 on the scale of 10. The relevance of an item for accuracy, Precision, Recall, and F-measure are identified using a threshold value set at 3. Rating value below this threshold is treated as not relevant or disliked item.

In order to develop our GA based approach, we applied a simple GA and evolved the weights through a simple unsigned binary encoding by appending 8-bits for each criteria. The GA begins with random genotypes. The criteria weight for each decimal value falls in the range of 0–255 (referred to as the lower bound-upper bound). The final vector consists of 32-bit binary digits (i.e., 8 × 4). The initial criteria weights are determined by dividing each criteria's weighting value by the total of all decimal values for all criteria weights. The population size is selected as 10 in all the experiments, while algorithm iterates for maximum 30 generations. Every time one generation passes into the next, 10 new people are produced. The good individuals from these 10 recently produced individuals take the place of the bad individuals from the previously fixed-size population. The process terminates in choosing the run with the best weight out of the 10 runs for each active user. By using single point crossover to the eight parents who were chosen at random, eight new offspring are generated. Rest two individuals are used during mutation operation. GA learns weights for each criteria in the guidance of training ratings of the user. The evolving set of weights obtained during the intermediate steps is used to obtain the predicted ratings for every item. Equation (10) is used to calculate the fitness score of each individual. Finally, after termination, the evolved set of weights is added to each criteria and used to predict ratings for all items in the test set. The computational complexity of the GA based approach is $O(PGRn\hat{}2)$, where P is the population size, G is the number of generations, R is the number of runs, n is the size of the training data. The GA parameter values used in the experiments are given in Table 4.

We develop a GA based RS through weighted normalized rating count and common rating weight similarity (GA-NRC-CRS). The normalized rating count and common rating weight similarity are used to develop MCRS (NRC-CRS) prior to employing the GA. We compare our proposed GA-NRC-CRS approach with the GA and MCRS based on credibility score (MCRS-CS) [20] and multicriteria collaborative filtering based on fuzzy naive Bayesian classifier (MCCF-FNB) [33] approaches. Moreover, we also compare our proposed approach with two state-of-the-art MCRS namely Clustering with Fuzzy Side Information RS (Clust-FSI) [5] and Clustering with Side Information RS (Clust-SI) [21]. Besides, the comparison includes the single criteria approaches SCCRS [23], SCOS [31], SCMMD [21], SCJC, SCPC, and SCCS. The level of dataset density can be used to determine the density factor value in the CRS method, and the density of the Yahoo! Movies dataset is around 1%. Therefore, after performing multiple experiments, we select 4 as the value of the density factor *df*. Moreover, we use six different evaluation measures namely, MAE, RMSE, Coverage, Precision, F-measure, and Accuracy to evaluate the performance of the system [1]. The coverage metric is used to describe the proportion of items for which the RS can make predictions [6]. It is possible that the RS won't be able to predict every item. Larger value of Coverage corresponds to the system performing better. The following equation (14) of Coverage calculates the percentage of items across all users for which a prediction was required and one was given by the system.

$$Coverage = \frac{\sum_{i=1}^{N} p_i}{\sum_{i=1}^{N} n_i} \tag{14}$$

where p_i is the total number of items predicted for the active user and n_i is the total ratings in the test set.

4.2. Results

The experimental results are presented in three stages in this section. Firstly, we discuss the results obtained on fixed neighborhood size and split for better clarity. Secondly, we examine the results in detail by varying neighborhood sizes and splits. Thirdly, we present a comparative analysis of proposed approach with existing MCCF approaches.

4.2.1. Results on fixed neighborhood size and split

For each active user, out of all runs, the run with the best results is selected for GA-NRC-CRS and compared against the results from different comparing approaches. Results presented in Table 5, depict the relative performances of various recommendation approaches on the basis of six evaluation measures. The results are obtained when neighborhood size is kept at 30 and training-test split ratio at 70-30%. Results summarized in this table show that GA based GA-NRC-CRS approach outperforms both single criteria and MCRS on all evaluation measures.

The above presented results show that the MAE and RMSE values for GA-NRC-CRS approach are lower than those for all baseline approaches. The higher prediction values for GA-based systems show that a good set of matching users has been discovered, which improves the performance of the system. It is clear that the MAE and RMSE of the GA-NRC-CRS approach are much better than the state-of-the-art single criteria (SCOS, SCMMD) and multi-criteria (Clust-FSI and Clust-SI) approaches.

Our proposed RS has a higher Coverage than the results obtained using multi-criteria and single-criteria approaches. The system can predict more items than other approaches, as demonstrated by the higher Coverage value. As can be seen, the single-criteria SCCRS approach has a higher Coverage than the multi-criteria Clust-FSI and Clust-SI approaches. The CRS is able to achieve this improvement in predictive accuracy based measures (MAE, RMSE, and Coverage) only. In terms of recommendation accuracy measures (Precision, F-measure, and Accuracy), its performance is found lower than other single criteria approaches. This signifies that CRS is not performing well in terms of recommendation accuracy measures. On the other hand, the results show that GA-NRC-CRS approach has

Table 4
Parameter values used for genetic algorithm.

Parameter	Value	Description
Population size	10	Each generation has these many individuals in the population.
Termination threshold	0.025	This threshold is used to stop the algorithm when the fitness of best individual is below the given value.
Maximum iterations during each run	30	If the stopping condition does not meet then the algorithm iterates till 30 iterations and the best solution for the last iteration is used as the final solution.
Number of runs	10	The system runs for 10 times for each active user.

Table 5
Comparison of the performance of various approaches on fixed neighborhood size and split ratio.

RS type	Approach	MAE	RMSE	Coverage	Precision	F-measure	Accuracy
Multi-criteria	GA-NRC-CRS	**0.791**	**0.9976**	**0.9503**	**0.93**	**0.9298**	**0.909**
	NRC-CRS	0.8074	1.0115	0.9339	0.9204	0.8917	0.8435
	Clust-FSI	0.8675	1.0735	0.8901	0.9109	0.8839	0.8302
	Clust-SI	0.8822	1.0839	0.8817	0.9077	0.8871	0.8349
Single-criteria	SCCRS	0.8194	1.025	0.9335	0.9184	0.8862	0.8373
	SCOS	0.8373	1.0567	0.9073	0.9052	0.8893	0.8381
	SCMMD	0.8871	1.0904	0.7852	0.9139	0.887	0.8337
	SCJC	0.9459	1.1664	0.7137	0.9188	0.8808	0.8199
	SCPC	0.9586	1.1772	0.6974	0.916	0.8768	0.8123
	SCCS	0.9694	1.1923	0.6967	0.914	0.8765	0.8157

significant enhancement in its performance in terms of recommendation accuracy measures also. This signifies that the incorporation of GA and multi-criteria ratings have enhanced the performance of the traditional SCRS approach.

4.2.2. *Results on varying neighborhood size and splits*

The results presented in the previous section show the performance of GA based MCRS for only on top-30 neighborhood size and fixed train-test split ratio at 70-30%. In this section, we discuss the performance of various recommendation approaches through varying neighborhood sizes and splits. The value of k for Top-k neighborhood size varies from 10 to 70, while the training-testing is done by varying training size from 60% to 90% on the scale of 10.

We have conducted extensive experiments using seven different neighborhood set sizes, four different train-test split ratio, and six evaluation measures for ten different RSs. Therefore, total 168 different results are obtained for a single approach. Table 6 shows the results of the proposed GA based MCRS obtained by varying neighborhood set sizes and train-test split ratio.

In the same way, results are obtained for the other nine approaches by varying neighborhood set size and split ratio. For better understanding, we compute the maximum, minimum, and average values of the different evaluation measures for all ten different recommendation approaches. These results are presented in Table 7.

Above presented results show that the proposed GA-NRC-CRS approach has lowest maximum, minimum, and average in terms of MAE and RMSE evaluation measures compared to all comparative approaches. While highest maximum, minimum, and average values in terms of Precision, F-measure, and accuracy evaluation measures. It is interesting to see that the SCCRS and SCPC single criteria approaches show minimum RMSE and maximum precision, respectively. It infers that these approaches show better RMSE and Precision in some of the cases. But the actual performance can only be analyzed through their average result values. Therefore, for better understanding, the average results for MAE, RMSE, Coverage, Precision, F-measure, and Accuracy evaluation measures presented in Table 7 are graphically represented in Figs. 7–12, respectively.

Table 6
Results of the GA based multi-criteria approach on varying neighborhood set size and train-test split ratio.

Evaluation Measure	Split ratio	Neighborhood Set Size						
		Top-10	Top-20	Top-30	Top-40	Top-50	Top-60	Top-70
MAE	60–40	0.8108	0.8054	0.7913	0.79	0.7919	0.7871	0.7827
	70–30	0.815	0.8093	0.791	0.7931	0.7784	0.7802	0.7779
	80–20	0.8201	0.81	0.8088	0.7997	0.7997	0.7953	0.7901
	90–10	0.8622	0.8498	0.8366	0.8237	0.8155	0.8202	0.8111
RMSE	60–40	1.047	1.0289	1.0029	1.0073	1.0022	1.0001	0.9903
	70–30	1.009	0.9998	0.9976	0.9906	0.9804	0.9709	0.9744
	80–20	0.9945	0.9961	0.99	0.9798	0.9705	0.9708	0.9711
	90–10	0.985	0.9871	0.9675	0.9623	0.959	0.9587	0.9587
Coverage	60–40	0.8944	0.909	0.9372	0.9631	0.9789	0.9765	0.9849
	70–30	0.8996	0.943	0.9503	0.9621	0.9787	0.9821	0.9892
	80–20	0.9189	0.9235	0.95	0.9695	0.979	0.9848	0.9847
	90–10	0.9121	0.9256	0.9207	0.9514	0.9705	0.9723	0.9832
Precision	60–40	0.9349	0.9301	0.9282	0.9208	0.9237	0.9291	0.9203
	70–30	0.9302	0.9319	0.93	0.9289	0.9208	0.9291	0.9247
	80–20	0.931	0.9232	0.9305	0.9316	0.9307	0.9299	0.9364
	90–10	0.9394	0.9388	0.9369	0.9367	0.9356	0.9302	0.9311
F-measure	60–40	0.9189	0.9148	0.9207	0.9311	0.93	0.9366	0.9319
	70–30	0.9127	0.9184	0.9298	0.9309	0.9341	0.9336	0.9349
	80–20	0.9259	0.9251	0.9304	0.9347	0.9388	0.9391	0.939
	90–10	0.9329	0.93	0.9362	0.937	0.9392	0.9357	0.9499
Accuracy	60–40	0.9122	0.9143	0.9109	0.9097	0.9122	0.9122	0.9134
	70–30	0.9092	0.9088	0.909	0.9187	0.9191	0.9145	0.9144
	80–20	0.9133	0.9134	0.917	0.9199	0.921	0.9203	0.9207
	90–10	0.9089	0.9035	0.9025	0.9087	0.9101	0.9121	0.9162

Table 7
Comparison of the performance of various approaches on varying neighborhood size and split ratio using different evaluation measures.

Evaluation Measure	Parameter	GA-NRC-CRS	NRC-CRS	Clust-FSI	Clust-SI	SCCRS	SCOS	SCMMD	SCJC	SCPC	SCCS
MAE	Maximum	**0.8622**	0.8726	0.9581	0.9291	0.8963	0.9355	0.945	1.0181	1.0241	1.0487
	Minimum	**0.7779**	0.7796	0.8362	0.8314	0.7847	0.8024	0.8463	0.8898	0.8892	0.9057
	Average	**0.8052**	0.8144	0.8808	0.8784	0.8209	0.8444	0.8919	0.9397	0.9422	0.9608
RMSE	Maximum	**1.047**	1.0718	1.1424	1.1289	1.0899	1.1665	1.1545	1.22	1.2235	1.2531
	Minimum	0.9587	0.9608	1.008	1.0129	**0.9562**	0.9641	1.0013	1.0685	1.0602	1.0885
	Average	**0.9875**	0.9981	1.0622	1.0571	1.0042	1.0389	1.0678	1.127	1.1323	1.1523
Coverage	Maximum	**0.9892**	0.9765	0.956	0.9523	0.9758	0.969	0.943	0.8941	0.872	0.8619
	Minimum	**0.8944**	0.7722	0.6059	0.6157	0.7663	0.7313	0.4674	0.4182	0.4021	0.4064
	Average	**0.9534**	0.9205	0.8584	0.8579	0.9182	0.9016	0.7828	0.7148	0.6962	0.6911
Precision	Maximum	0.9394	0.9456	0.95	0.9473	0.9448	0.9236	0.9528	0.9563	**0.9682**	0.967
	Minimum	**0.9203**	0.9153	0.9109	0.9074	0.9132	0.8949	0.9088	0.9109	0.9078	0.9063
	Average	**0.9301**	0.9232	0.9202	0.9158	0.9228	0.909	0.9193	0.9241	0.9253	0.9219
F-measure	Maximum	**0.9499**	0.9209	0.9283	0.9313	0.9171	0.9146	0.9405	0.9361	0.9454	0.9476
	Minimum	**0.9127**	0.8848	0.8838	0.884	0.886	0.8799	0.8824	0.8729	0.8708	0.8731
	Average	**0.931**	0.8982	0.8956	0.8948	0.8983	0.8986	0.8987	0.8905	0.8911	0.8893
Accuracy	Maximum	**0.921**	0.8772	0.8935	0.8911	0.8726	0.8698	0.9051	0.9043	0.9166	0.9166
	Minimum	**0.9025**	0.834	0.8253	0.8311	0.8347	0.8221	0.825	0.8085	0.8106	0.8101
	Average	**0.9130**	0.8502	0.8444	0.8437	0.8504	0.8473	0.8483	0.8363	0.8373	0.8334

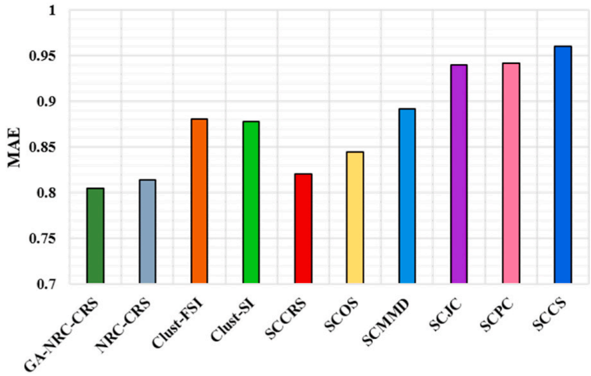

Fig. 7. Results of the comparison of different approaches on the basis of average MAE.

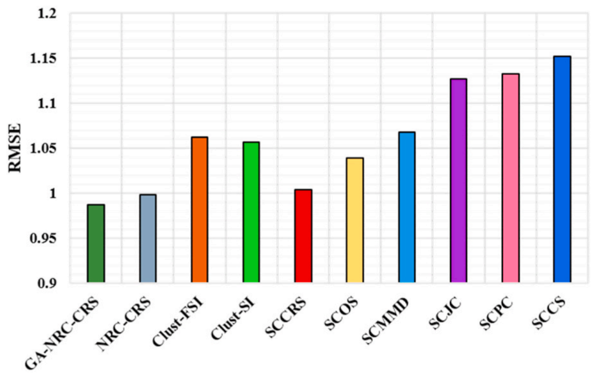

Fig. 8. Results of the comparison of different approaches on the basis of average RMSE.

From Figs. 7 and 8, it is evident that the GA-NRC-CRS approach has the lowest MAE and RMSE compared to other comparing approaches. It can be seen that some of the single criteria approaches outperform the Clust-FSI and Clust-SI multi-criteria approaches. This implies that only incorporating multi-criteria ratings does not improve the performance.

Fig. 9 shows the average Coverage of different neighborhood sets and splits. The average Coverage of GA based multi-criteria approach is higher than all other comparing approaches. We have seen from the results that Coverage value for GA-NRC-CRS approach is in the range of 89%–99% for all experimental settings whereas other approaches exhibit a sizable gap in their Coverage range.

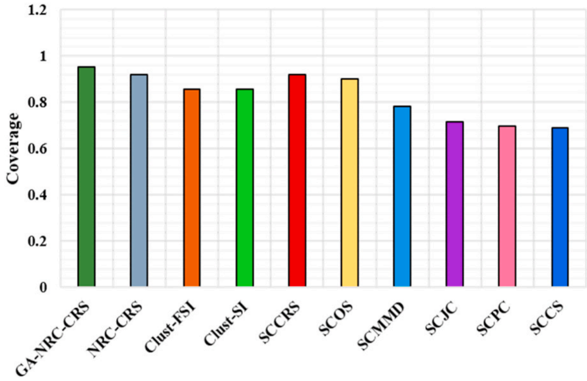

Fig. 9. Results of the comparison of different approaches on the basis of average Coverage.

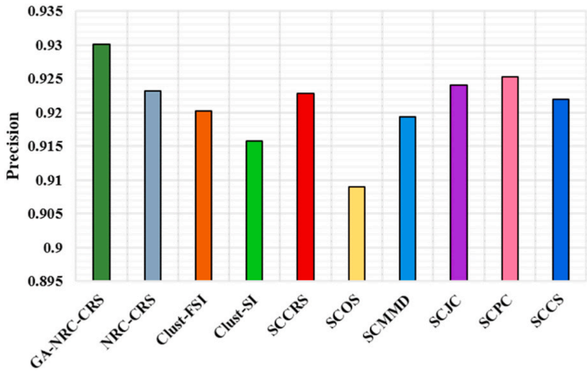

Fig. 10. Results of the comparison of different approaches on the basis of average Precision.

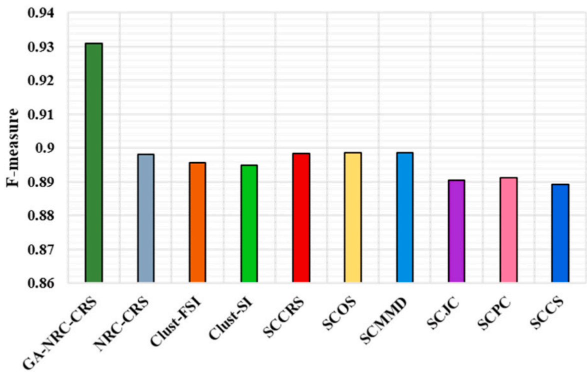

Fig. 11. Results of the comparison of different approaches on the basis of average F-measure.

Fig. 10 shows the average Precision of all the comparing approaches obtained on different neighborhood sets and for all splits. The aim of this comparison is to show the complete Precision analysis of our proposed approach with the remaining RSs. We can see that the Precision of our GA-NRC-CRS approach is superior to the rest of the approaches. The performance of NRC-CRS does not have any visible improvement after incorporating multi-criteria rating into SCCRS. This signifies that multi-criteria ratings are not making any improvement at the average precision level. The state-of-the-are SCOS approach gives the lowest performance in terms of average Precision.

Fig. 11 shows the average F-measure of all the comparing approaches obtained on different neighborhood sets and for all splits. It is interesting to see that the F-measure performance of all of the comparing approaches is almost similar except to the GA based approach. Our proposed GA-NRC-CRS approach achieved a significant F-measure performance improvement than the rest of the approaches. It can also be seen that the performance of SCCRS, SCOS, and SCMMD has almost equal performance in terms of the

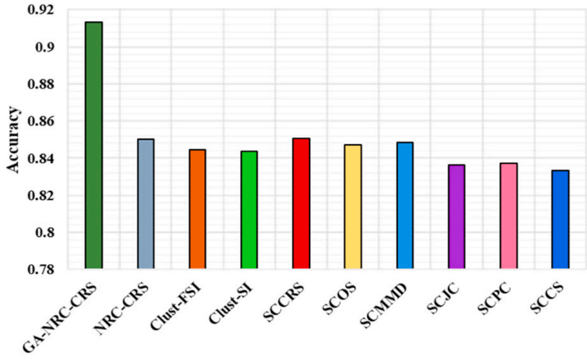

Fig. 12. Results of the comparison of different approaches on the basis of average Accuracy.

average F-measure.

Similarly, the average Accuracy of all the comparing approaches obtained on different neighborhood sets and for all splits is shown in Fig. 12. From this figure, we can find that our proposed GA based MCRS gives the highest Accuracy compared to the rest of the RSs.

The above presented results obtained on fixed and varying neighborhood size and split show that GA based approach (GA-NRC-CRS) always outperforms than the single criteria and multi-criteria based approaches. The use of multi-criteria ratings has improved the performance of single criteria SCCRS approach but failed to gain improvement in some of the experiments. Similarly, the CRS measure based SCCRS approach stands as the best single criteria approach but it also fails in terms of recommendation accuracy measures. This signifies that the use of GA is the only factor that captures the optimal preferences of the users and improves the overall effectiveness of MCRS.

4.2.3. Comparative analysis of proposed approach with existing MCCF approaches

It can be observed from above presented results that our proposed GA-NRC-CRS approach stands the best performing approach across all evaluation measures. In this subsection, we compare and discuss the results of our proposed GA-NRC-CRS approach, along with the existing GA based and neighborhood based MCCF approaches. The comparison aims to demonstrate the superior performance of our approach. Therefore, we compare our proposed GA-NRC-CRS approach with the GA and MCRS based on credibility score (MCRS-CS) [20] and multicriteria collaborative filtering based on fuzzy naive Bayesian classifier (MCCF-FNB) [33] approaches. The results are shown in Table 8 and graphically represented in Fig. 13. The evaluation metrics, including MAE, RMSE, coverage, precision, F-measure, and accuracy, are used for the analysis. The results are obtained when neighborhood size is kept at 30 and training-test split

Table 8
Comparison of the performance of proposed approach and existing MCCF approaches.

Approach	MAE	RMSE	Coverage	Precision	F-measure	Accuracy
GA-NRC-CRS	**0.791**	**0.9976**	0.9503	**0.93**	**0.9298**	**0.909**
MCRS-CS [20]	0.8624	1.0858	**0.9584**	0.9235	0.9101	0.83
MCCF-FNB [33]	0.8169	1.0423	0.9342	0.7445	0.8408	0.7326

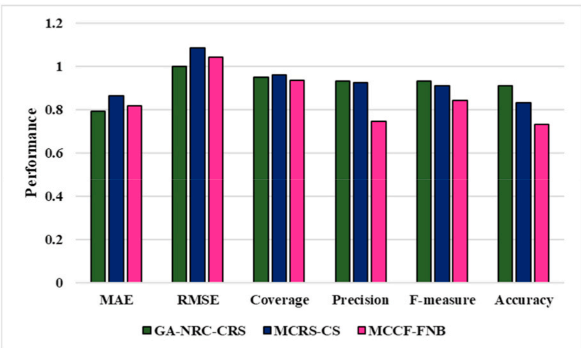

Fig. 13. Results of the comparison of proposed approach with existing MCCF approaches.

ratio at 70-30%.

From Table 8 and Fig. 13, it is evident that our proposed GA-NRC-CRS approach outperforms both the MCRS-CS and MCCF-FNB approaches in terms of MAE and RMSE. This indicates that our approach provides more accurate predictions compared to the other two approaches. On the other hand, the MCRS-CS approach achieves the highest coverage among the three approaches, implying that it can make predictions for a larger number of items. However, it does not perform well in terms of recommendation accuracy measures. When comparing the neighborhood-based MCCF-FNB approach with the MCRS-CS approach, the former exhibits better performance in terms of MAE and RMSE. However, it does not perform better than our proposed GA-NRC-CRS approach across all six performance measures.

In summary, the above presented results confirm that the GA-NRC-CRS approach achieves the lowest MAE and RMSE values, signifying its superior accuracy. Although the MCRS-CS approach performs better in terms of coverage, it does not perform well in recommendation accuracy. The neighborhood-based MCCF-FNB approach performs better than the MCRS-CS approach in terms of MAE and RMSE, but it still underperforms when compared to our proposed GA-NRC-CRS approach across all evaluation metrics.

5. Conclusion and future work

This paper proposes an adaptive GA based approach for MCRS. Since an item is associated with multiple criteria features, all criteria contribute equally during similarity computation. This does not accurately reflect the situation in which users make decisions by giving higher preferences or weights to some specific features than others. These priorities or weights are subject to vary with time and the evolving choices of each user. We employed the GA to adapt these multi-criteria weights using a weighted normalized rating count measure. The multidimensionality issue was also addressed by assessing the single criteria and multi-criteria ratings separately. Experimental results showed that the GA-based MCRS approach outperformed various single criteria, multi-criteria, GA-based MCCF baseline approaches. The proposed approach can be applied to any RS with additional features. In future work, one can develop a context-aware RS using the proposed approach. Additionally, appropriate fuzzy sets can be designed to fuzzify criteria features to develop a fuzzy genetic based recommendation approach.

Author contribution statement

Mohammed Wasid: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Rashid Ali: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.

Sana Shahab: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Data availability statement

Data will be made available on request.

Funding

This research was supported by the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R259), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] E. Zangerle, C. Bauer, Evaluating recommender systems: survey and framework, ACM Comput. Surv. 55 (8) (2022) 1–38.
[2] D. Monti, G. Rizzo, M. Morisio, A systematic literature review of multicriteria recommender systems, Artif. Intell. Rev. 54 (2021) 427–468.
[3] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE transactions on knowledge and data engineering, IEEE 17 (6) (2005) 734–749.
[4] M. Nilashi, O. Ibrahim, H. Ahmadi, L. Shahmoradi, S. Samad, K. Bagherifard, A recommendation agent for health products recommendation using dimensionality reduction and prediction machine learning techniques, Journal of Soft Computing and Decision Support Systems 5 (3) (2018) 7–15.
[5] M. Wasid, R. Ali, Fuzzy side information clustering-based framework for effective recommendations, Comput. Inf. 38 (3) (2019) 597–620.
[6] M.Y.H. Al-Shamri, K.K. Bharadwaj, Fuzzy-genetic approach to recommender systems based on a novel hybrid user model, Expert systems with applications, Elsevier 35 (3) (2008) 1386–1399.
[7] L. Liu, N. Mehandjiev, D.L. Xu, Multi-criteria service recommendation based on user criteria preferences, in: Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, 2011, pp. 77–84.
[8] M. Kunaver, T. Pofizrl, Diversity in Recommender Systems: A Survey", Knowledge-Based Systems, vol. 123, Elsevier, 2017, pp. 154–162.
[9] M. Wasid, R. Ali, Use of Soft Computing Techniques for Recommender Systems: an Overview. Applications of Soft Computing for the Web, 2017, pp. 61–80.
[10] Seyedali Mirjalili, Seyedali Mirjalili, Genetic algorithm." Evolutionary Algorithms and Neural Networks: Theory and Applications, 2019, pp. 43–55.
[11] S. Gupta, V. Kant, A comparative analysis of genetic programming and genetic algorithm on multi-criteria recommender systems, in: Proceedings of the 5th International Conference on Communication and Electronics Systems, IEEE, 2020, pp. 1338–1343.

[12] S. Ujjin, P.J. Bentley, Using evolutionwy to learn user preferences", in recent advances in simulated evolution and learning, in: Kay Chen Tan (Ed.), Meng Hiot Lim, Xin Yao and Lipo Wang, World Scientific Publishing, 2004, pp. 20–40.
[13] S. Gupta, V. Kant, May. A review and classification of multi-criteria recommender systems, in: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, 2020, pp. 1156–1162.
[14] S. Gupta, V. Kant, A model-based approach to user preference discovery in multi-criteria recommender system using genetic programming, Concurrency Comput. Pract. Ex. 34 (11) (2022) e6899.
[15] K. Anwar, J. Siddiqui, S.S. Sohail, Machine learning-based book recommender system: a survey and new perspectives, Int. J. Intell. Inf. Database Syst. 13 (2–4) (2020) 231–248.
[16] M. Wasid, K. Anwar, October. An augmented similarity approach for improving collaborative filtering based recommender system, in: 2022 International Conference on Data Analytics for Business and Industry (ICDABI), IEEE, 2022, pp. 751–755.
[17] M. Nilashi, A. Ahani, M.D. Esfahani, E. Yadegaridehkordi, S. Samad, O. Ibrahim, N.M. Sharef, E. Akbari, Preference learning for ecofriendly hotels recommendation: a multi-criteria collaborative filtering approach, Journal of Cleaner Production, Elsevier 215 (2019) 767–783.
[18] M. Nilashi, O. bin Ibrahim, N. Ithnin, N.H. Sarmin, A multicriteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA: ANFIS, Electronic Commerce Research and Applications, Elsevier 14 (6) (2015) 542–562.
[19] S. Gupta, V. Kant, An aggregation approach to multi-criteria recommender system using genetic programming, Evolving Systems, Springer 11 (1) (2020) 29–44.
[20] S. Gupta, V. Kant, Credibility Score Based Multi-Criteria Recommender System", Knowledge-Based Systems, Elsevier, 2020, pp. 105–756.
[21] M. Wasid, R. Ali, Multi-criteria clustering-based recommendation using Mahalanobis distance, International Journal of Reasoning-based Intelligent Systems, Inderscience Publishers, Switzerland 12 (2) (2020) 96–105.
[22] C.S. Hwang, Genetic algorithms for feature weighting in multi-criteria recommender systems, in: Journal of Convergence Information Technology, 2010.
[23] M. Wasid, R. Ali, A frequency count approach to multi-criteria recommender system based on criteria weighting using particle swarm optimization, Appl. Soft Comput. 112 (2021), 107782.
[24] S. Geng, X. He, Y. Wang, H. Wang, B. Niu, K.M. Law, Multicriteria recommendation based on bacterial foraging optimization, Int. J. Intell. Syst. 37 (2) (2022) 1618–1645.
[25] C.S. Hwang, Genetic algorithms for feature weighting in multicriteria recommender systems, Journal of Convergence Information Technology 5 (8) (2010) 126–136.
[26] G. Kaur, S. Ratnoo, Adaptive genetic algorithm for feature weighting in in multi-criteria recommender systems, Pertanika Journal of Science & Technology 27 (1) (2019).
[27] H. Mohamed, L. Abdulsalam, H. Mohammed, September. Adaptive Genetic Algorithm for Improving Prediction Accuracy of a Multi-Criteria Recommender System. In 2018 IEEE 12th International Symposium on Embedded Multicore/Many-core Systems-On-Chip (MCSoC), IEEE, 2018, pp. 79–86.
[28] M. Hassan, M. Hamada, September. Improving Prediction Accuracy of Multi-Criteria Recommender Systems Using Adaptive Genetic Algorithms. In 2017 Intelligent Systems Conference (IntelliSys), IEEE, 2017, pp. 326–330.
[29] M. Hamada, A.L. Ometere, O.N. Bridget, M. Hassan, S.Y. Ilu, A fuzzy-based approach and adaptive genetic algorithm in multi-criteria recommender systems, Adv. Sci. Technol. Eng. Syst. J. 4 (2019) 449–457.
[30] M. Hassan, M. Hamada, Genetic algorithm approaches for improving prediction accuracy of multi-criteria recommender systems, Int. J. Comput. Intell. Syst. 11 (1) (2018) 146–162.
[31] A. Gazdar, L. Hidri, A new similarity measure for collaborative filtering based recommender systems, in: Knowledge-Based Systems vol. 188, Elsevier, 2020.
[32] K. Lakiotaki, N.F. Matsatsinis, A. Tsoukias, Multicriteria user modeling in recommender systems, IEEE Intelligent Systems, IEEE 26 (2) (2011) 64–76.
[33] Vibhor Kant, Tanisha Jhalani, Pragya Dwivedi, Enhanced multi-criteria recommender system based on fuzzy Bayesian approach, Multimed. Tool. Appl. 77 (2018) 12935–12953.