An efficient framework to combat multidimensionality in multi-criteria recommender systems

An Efficient Framework to Combat Multidimensionality in Multi-Criteria Recommender Systems

Khalid Anwar[1], Mohammed Wasid[2,*], Aasim Zafar[3], Arshad Iqbal[4]

[1,2] School of Computer Science Engineering & Technology, Bennett University, Greater Noida -201310, India
[3]Department of Computer Science, Aligarh Muslim University, Aligarh - 202002, India
[4]K.A. Nizami Centre for Quranic Studies, Aligarh Muslim University, Aligarh - 202002, India

(*Corresponding Author: Mohammed Wasid (*erwasid@gmail.com*))

Abstract. Recommender Systems (RS) are widely adopted software tools for easing information overload and generating personalized recommendations to the users. Collaborative filtering (CF) is one of the extensively implemented recommendation techniques that provides recommendations based on the preferences of other like-minded users. Traditional RS generally works on a single numerical rating on items provided by the users. Recent research suggests that the incorporation of multi-criteria ratings into classical RS has greatly improved the utility of recommendations. However, incorporation of multi-criteria ratings into classical systems is still a matter of concern as it causes the multidimensionality issue. In this paper, we propose an efficient framework to reduce the multidimensionality in multi-criteria recommender systems (MCRS) using two different multi-criteria aggregation methods. Unlike the traditional aggregation methods, our utilized aggregation methods combine the multiple ratings by preserving the importance of individual criteria ratings. Moreover, in order to identify the most accurate neighborhood set for each user, we combine the separately calculated overall and aggregated score based similarities together to obtain the total similarity. Extensive experiments performed on benchmark dataset indicate that our proposed approaches outperformed the existing multi-criteria recommendation approaches on various evaluation measures.

Keywords: Multi-Criteria Recommender System, Collaborative Filtering, Rating Aggregation, OWA, TOPSIS

1. Introduction

In an era when information and communication technologies are all over, the amount of information on the web is growing exponentially. This exponential growth of information over the internet has eased the accessibility of digital resources and created the problem of information overload. To put this into perspective, the internet held over 4.7 billion web pages and over 500 hours of video content are uploaded to YouTube every minute. Additionally, the number of daily tweets on Twitter exceeded 500 million and approximately 500,000 new blogs are published daily. This deluge of data has made it increasingly challenging for individuals to sift through and discern accurate, relevant and valuable information from the vast sea of noise, posing a significant challenge to information seekers and decision-makers. Recommender Systems (RS) are personalization tools that are used to filter out huge amount of information, consecutively make personalized recommendations and reduce the problem of information overload [1]. RS have replicated the human experts in various real-life decision-making. They are being used in e-commerce to know the sentiments and preferences of customers and recommend products and services as per their needs. In academia, RS are designed to recommend institutes, courses, books etc. [2]. RS are also utilized in diverse areas, like entertainment, sports, news, hotels, etc. to fulfill the needs of users and make profits for the organizations. Traditional RS generate recommendations based on three main filtering techniques, namely Content-based (CB), collaborative filtering (CF) and hybrid filtering (HF). Among them, CF is one of the widely implemented recommendation techniques which provides recommendations to a user based on the preferences of other like-minded users [3]. For example, at Netflix, if a user frequently watches action movies with specific actors, the system will recommend other action movies featuring those actors or with similar attributes. It is further categorized into memory-based and model-based CF techniques [4]. A model-based technique is developed using a modelling approach such as, machine learning or statistical learning techniques. Memory-based technique directly uses historical ratings and is further identified as item-based and user-based CF techniques [5]. The item-based technique calculates the similarities among the items while the user-based technique calculates the similarities among the users to identify similar items (or users) [6].

On the other hand, the CB approach considers the past browsing and purchasing behaviour of the user. This technique finds the description or content of the item to be recommended, compare them with the content of the items preferred in the past and then recommend it if both are highly similar [7]. For example, at Amazon, if a user buys a camera and other users who bought the same camera also bought a specific lens, Amazon may recommend that lens to the user. However, CB and CF methods have been extensively used techniques but they face certain limitations while making appropriate recommendations. Literature has suggested that these limitations of individual techniques can be overcome if combined together, known as hybrid technique [8]. For example, Spotify analyzes the audio characteristics of songs (content-based) and considers user listening history and preferences (collaborative filtering). If a user frequently listens to rock music, the system may recommend rock songs with attributes similar to their listening history while also considering songs liked by users with similar tastes. There are various ways through which both techniques can be combined together effectively [9]. Usually, collaborative filtering based recommender system takes the single overall rating of an item, finds similar items based on the overall ratings and generates recommendations [10]. But there are always multiple features of an item that can be used to judge it [11]. For example, a hotel can be rated on features like cleanliness, quality of food and quality of service. So, recommending items based on just one overall rating doesn't show enough about the actual choice of a user. Therefore, multi-criteria recommender systems (MCRS) are designed to incorporate ratings of multiple components of an item [12]. Literature shows that the incorporation of multi criteria (MC) rating into traditional RS improves the performance and accuracy of these systems [13]. However, MCRS are more effective than the traditional RS but they suffer from the multidimensionality issue, thus making the system more complex. The curse of dimensionality is one of the severe issues in MCRS.

By going through the literature, we identified a few methods through which multidimensionality issue can be addressed. One possible approach is to combine the MC ratings into a single rating by using some aggregation methods such as, taking average, min, or max and make prediction based on this combined rating. This is the simplest approach of aggregation, but eventually degrades the essence of representing user preferences over multiple components of the items. Moreover, authors in [14] suggested two different approaches to keep the essence of multi-criteria ratings into recommendations. One approach calculates the similarities between the users on each individual rating. Then aggregate the individual similarities into aggregated similarity using simple average, min or max and predict the overall ratings based on the aggregated similarity. Another suggested approach divides the MC rating into different individual rating and then make prediction for individual rating. After that aggregation, the various predicted ratings into single overall rating using some machine learning or statistical model. Finally, the items are recommended based on the aggregated single rating. This approach is computationally complex and the selection of an appropriate aggregation model is also difficult [15]. We observe that some of these approaches are simple enough and thus fail to capture the essential properties of MC ratings or some of them are too complex in reality. Therefore, we identified that Ordered Weighted Averaging (OWA) and Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) are two well-known aggregation approaches that can also be used for aggregating the multi-criteria ratings into single rating by preserving the actual essence of individual criteria.

In this paper, we propose an effective framework to handle multi-dimensionality issue in multi-criteria recommender systems by incorporating the multi-criteria ratings through similarity fusion approach. The similarity fusion is done by combining the similarities obtained from overall ratings (single criteria) and multi-criteria ratings separately. Both similarities are obtained by following the traditional collaborative filtering approach. The only difference is that the multi-criteria similarity is obtained on the aggregated scores calculated through the OWA and TOPSIS aggregation methods. The combined similarity helps us in identifying the most accurate neighborhood set than the one obtained through the overall ratings only. The proposed approach is able to reduce the multidimensionality, complexity issues and improves the quality of recommendations. The major contribution of this paper is illustrated by the following points:

- An efficient Framework to handle the multidimensionality issue in MCRS is developed that improves the performance of RS by aggregating multi-criteria ratings.
- A new method to compute the total similarity among the users is developed that fuses the similarity calculated using two different ratings.
- The OWA and TOPSIS with objective weights are used to aggregate various ratings. The use of these two approaches handles the uncertainty, inconsistency, and vagueness during aggregation.
- The proposed MCRS approaches significantly improve the prediction accuracy of two existing most effective similarity measures.

Overall, addressing the curse of multidimensionality in recommender systems is crucial for designing truly personalized and effective recommendation engines. The implications of overcoming this challenge include improved recommendation accuracy, enhanced user engagement, increased personalization, discovery of niche content, reduced information overload, and business growth. Ultimately, it leads to a more satisfying and relevant user experience in various domains, such as e-commerce, content streaming, and more.

The remaining of this paper is organized as: Section 2 presents the related work. In Section 3, we discuss various components of the proposed MCRS approach. Section 4 discusses the dataset, experimental settings, and evaluation measures used in the experiments. The results of the experiments performed are presented and discussed in Section 5 followed by the conclusion in Section 6.

2. Literature Review

Multi-criteria recommender systems are the extension of classical single criteria recommender systems which allow users to represent their preferences over multiple aspects of an item [14]. Adomavicius et al. [16] categorized the MCRS into two main categories, namely model-based and memory-based. Model-based MCRS are designed with the assumption that individual criteria ratings have some association with the overall rating. So, in model-based, multi-criteria dataset is divided into individual criteria datasets. After dividing into single individual criteria, ratings for each criterion are predicted using some traditional recommendation algorithm. Then some statistical or machine learning or deep learning model is trained to predict the overall rating from the individual ratings [17].

Several researchers have proposed model-based MCRS, such as Kant et al. [18] used a Fuzzy Bayesian approach to handle uncertainties in user choices and aggregate multi-criteria ratings into an overall rating. However, the limitation of this method lies in its computational complexity, as fuzzy-based approaches tend to increase the processing time. Authors in [19] proposed gradient decent based back propagation and employed genetic algorithm to predict the overall ratings from the individual criteria ratings. While the method improves accuracy, it requires intensive computational resources due to the hybrid nature of the algorithm, making it less suitable for real-time systems. Hassan and Hamada [21] [22] applied neural network to evaluate the performance of MCRS on various evaluation metrics. They also trained the neural network with particle swarm optimization (PSO) to infer the relationship between the individual criteria ratings and the overall rating and used the inferred relationship to compute the predicted overall rating from the predicted individual ratings. However, the neural network-based approaches are limited by their reliance on large amounts of training data, making them vulnerable to overfitting when data is sparse. Furthermore, [23] introduced a utility-based MCRS that uses a learning-to-rank method to predict individual criteria ratings. Although this approach incorporates utility functions, it lacks robustness in handling sparse datasets, which can affect its overall performance. Similarly, [24] developed a method using a rank aggregation technique to compute overall ratings from individual criteria ratings, but this method is limited by the simplicity of the aggregation technique, which may not capture the complexity of user preferences.

Various soft computing techniques, such as genetic algorithms, have been used for aggregating credibility scores of items [25]. However, these approaches often involve high computational complexity, limiting their applicability in real-time systems. For example, Gupta and Kant [26] [27] developed a model-based using genetic programming to discover user preferences and to aggregate the predicted ratings of individual criteria and improve the performance of the RS, but it too suffers from scalability issues due to the evolving nature of the genetic program. Furthermore, Mohamed et al. [20] developed an adaptive genetic algorithm to aggregate the predicted multi-criteria ratings into overall rating that improves the predictive accuracy of the RS. Although this improves predictive accuracy, the adaptive nature of genetic algorithms results in high computational costs, which may not scale well with larger datasets. Demirkiran et al. [28] used rough set theory to identify the relationship between the individual criteria ratings and overall rating and predicted the overall rating based the identified relationship. While the method is effective in managing uncertainty, its performance diminishes when dealing with large, high-dimensional datasets. The multidimensionality, sparsity and scalability issues have also been addressed in MCRS using higher order singular value decomposition (HOSVD) and adaptive neuro-fuzzy inference system (ANFIS) [29]. HOSVD is used for dimensionality reduction and to address the scalability issues whereas ANFIS is used to solve sparsity problem and to extract fuzzy-rules. Nilashi et al. [30] [31] used HOSVD and ANFIS with subtractive clustering to improve the predictive power of the MCRS. They further used clustering and support vector regression along with PCA to design effective MCRS [31]. Despite the improvements in predictive accuracy, the hybrid nature of these methods exacerbates their computational demands. Shambour [32] developed a deep learning-based MCRS that uses autoencoders for predicting individual criteria ratings and then averages these to compute the overall rating. While this approach enhances prediction accuracy, deep learning models require extensive computational resources and large datasets, which are limitations when working with sparse data or smaller systems. Although these approaches improve the predictive accuracy of MCRS, they are computationally expensive due to the need to predict ratings for each criterion [33].

The memory-based approach, on the other hand, calculates similarities between users based on individual criteria ratings and aggregates them using simple or weighted averages [34]. Although this approach is computationally less expensive, it is not as robust in handling the complexity of multi-criteria data. For instance, [35] proposed a fuzzy-linguistic approach to handle subjectivity and vagueness in MCRS, but the linguistic nature of the method introduces ambiguity in certain cases, leading to inaccuracies in rating predictions. Nialshi et al. [36] [37] used ANFIS combined with clustering techniques and Gaussian mixture models, but the hybrid nature of these systems adds significant computational overhead.

Kermany and Alizadeh [38] proposed a hybrid MCRS which fuse fuzzy cosine and Jaccard similarities to obtain total similarity among the users and movies. They used ontology and ANFIS to improve the efficiency of rating prediction. Wasid and Ali [39] [40] utilized Mahalanobis distance measure to identify efficient neighborhood set from the user clusters obtained through the multi-criteria rating clustering approach. They also proposed a frequency count approach to incorporate the multi-criteria ratings and employed a particle swarm optimization algorithm to identify evolve set of criteria weights during total similarity computation [13]. However, the use of optimization algorithms makes the system computationally intensive and less scalable for larger datasets. Yalcin and Bilge [41] proposed a similarity aggregation based binary MCRS that utilizes Naïve Bayes algorithm. They also developed two different similarity measures namely aggregation of user similarity and aggregation of item similarity based on binary ratings. They obtained aggregated similarity by using weighted average. The proposed approch improves the performance of MCRS on classification accuracy and F-1 score. Batmaz and Kaleli [42] proposed a hybrid similarity based MCRS that uses autoencoders to extract user ratings and reviews. The approach is effective in handling data sparsity and outperforms other similarity based MCRS on MAE, RMSE and coverage. Although this method addresses sparsity issues, it requires significant computational power, which limits its practical applicability in large-scale systems.

In summary, while the approaches discussed above have significantly improved the performance of MCRS, they introduce complexity due to the use of auxiliary methods for incorporating multi-criteria ratings. Additionally, these systems address issues such as data sparsity, scalability, and cold start, but fail to adequately address the multidimensionality problem in MCRS. Although some methods like HOSVD attempt to solve the scalability issue, they come at the cost of increased computational overhead. We believe that methods like OWA and TOPSIS, commonly used in multi-criteria decision-making, could be effective in handling multidimensionality, yet have not been explored in the development of MCRS in the current literature.

3. Proposed Framework for Multi-Criteria Recommender Systems

In this section, we present and discuss different components of the proposed MCRS recommendation framework, as shown in Figure 1. The developed framework consists of five components viz. (i) pre-processing, (ii) similarity computation using the overall rating, (iii) multi-criteria rating aggregation and similarity computation, (iv) similarity fusion and neighbourhood formation and (v) rating prediction and recommendation.

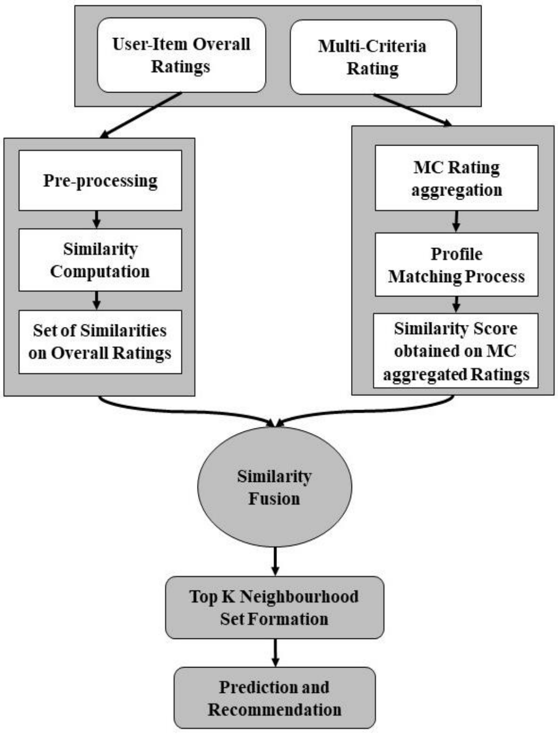


Figure 1 Framework of Rating Aggregation based Multi-Criteria Recommender Systems.

3.1. Pre-processing

In this step, the types of ratings available in the data set are identified. It separates the different types of ratings and also normalizes the ratings, if required. For example, the Yahoo! Movies data set [43] consists of both multi-criteria and overall ratings. So, this step separates the data set into two data parts i.e. (i) multi-criteria ratings and (ii) overall ratings. Also, original ratings in Yahoo! Movie data set are provided on 13 point rating scale (A+ to F). During pre-processing, these alphabetic ratings are converted into numeric form with minimum rating as 1 and maximum rating as 13.

3.2. Similarity Computation using Overall Ratings

As discussed, CF techniques are classified as memory based and model based techniques. The Model based techniques are designed using some modelling techniques such as machine learning techniques or some statistical model. Whereas, the memory based techniques compute the similarities among the users based on the historical user-item rating matrix. Various similarity measures have been used and developed in order to utilize collaborative filtering technique. For example, cosine similarity, Pearson correlation, etc. We identified Common Rating weight similarity (CRS) [13] and Our Similarity (OS) [44] measures as the recently developed highly accurate similarity measures in recommender systems. These measures can be utilized directly for the similarity computation among the users, as illustrated in Figure 2 and discussed below.

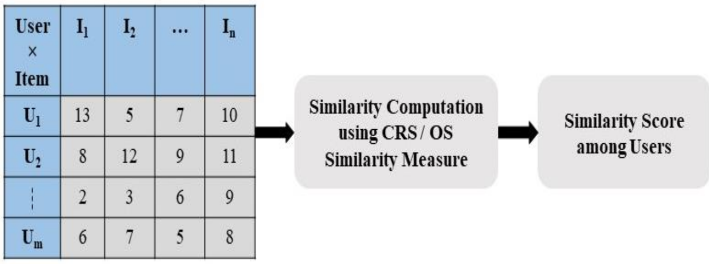


Figure 2 Similarity Computation through CRS or OS similarity measures using overall ratings.

3.2.1 Our Similarity (OS) measure: The OS measure is the recently developed efficient similarity measure that consists of two parts [44]. The OS measure is given in the following equation.

$$Sim_{uv}^{OS} = Sim_{uv}^{PNCR} \cdot Sim_{uv}^{ADF} \tag{1}$$

In the first part of the OS measure the cardinality of co-rated items between the given users u and v were considered. In this, the Percentage of Non-Common Rating (PNCR) were computed by using the following equation.

$$Sim_{uv}^{PNCR} = \exp\left(-\frac{N - |I_u \cap I_v|}{N}\right) \tag{2}$$

where I_u and I_v show the number of items that were rated by user u and user v, respectively and the total number of items in the data set were represented by N.

In the second part of the OS measure the Absolute Difference of Ratings (ADF) of user u and user v was calculated. The mathematical formulation of the ADF is given in the following equation.

$$Sim_{uv}^{ADF} = \frac{\sum_{i \in I} \exp\left(-\frac{|r_{u,i} - r_{v,i}|}{\max(r_{u,i} - r_{v,i})}\right)}{|I_u \cap I_v|} \tag{3}$$

where $max(r_{u,i}, r_{v,i})$ illustrates the maximum rating between $r_{u,i}$ and $r_{v,i}$.

3.2.2. Common Rating Weight Similarity (CRS) measure: This is another efficient similarity measure [13] comprised of two parts, as shown in the following equation.

$$Sim_{uv}^{CRS} = Sim_{uv}^{JTC} \cdot Sim_{uv}^{CRWF} \tag{4}$$

The first part of the CRS measure calculates the similarity among users using the Jaccard or Tanimoto coefficient (JTC) which is shown by following equation.

$$Sim_{uv}^{JTC} = \frac{|I_u \cap I_v|}{|I_u| + |I_v| - |I_u \cap I_v|} \tag{5}$$

The second part of the CRS measures serves as a similarity modifier and gives weight to the JTC based on density factor (df), as shown in the following equation.

$$Sim_{uv}^{CRWF} = \frac{1}{1 + e^{\left(-\frac{|I_u \cap I_v|}{df}\right)}} \tag{6}$$

The value of the df is inversely proportional to the level of sparsity in the dataset.

3.3. Multi-Criteria Rating Aggregation

In this step, we aggregate the multi-criteria ratings into an aggregated score through two extensively used and efficient MCDM approaches, namely OWA and TOPSIS. The aggregation performed using these MCDM techniques considers the conflict among the goals of various criteria and assigns weights to each criteria such that the importance of individual criteria is present in the aggregated score. Figure 3 presents the process of criteria aggregation followed by similarity computation on scores obtained using MC rating aggregation.

5

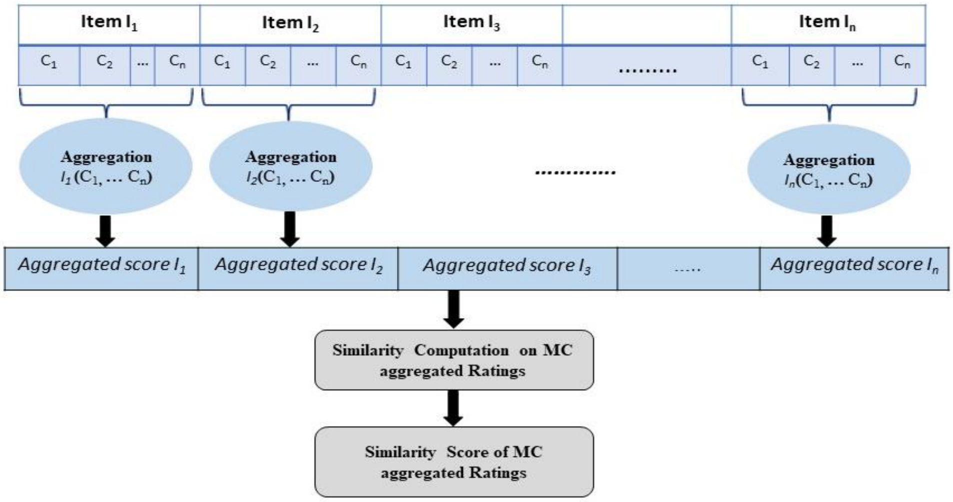


Figure 3 Process of criteria aggregation and obtaining similarity score.

3.3.1. Multi-Criteria Rating Aggregation using OWA

OWA is a weighted sum MCDM operator proposed by Yagar [45]. It reduces uncertainty in decision making process by using different linguistic quantifiers; 'at least half', 'at most half' and 'as many as possible' [46]. In general, OWA is defined as: $R^n \rightarrow R$, the n dimensional mapping over n dimensional vector of associated weights. Mathematically, the OWA score is calculated using the following equation.

$$OWA(c_1, c_2, c_3, \dots, c_n) = \sum_{i=1}^{n} w_i \, d_i \tag{7}$$

The sequence of criterion *c* is arranged in descending order to obtain d_i and the weight w_i ε *W*.

The OWA satisfies the following axiomatic properties to give the indication that it is a meaningful aggregation.

Axiomatic properties of OWA
P1. *It satisfies boundary condition, such as if all criteria values are 0s (1s) than it will give 0s (1s) as an output.* $OWA\ (0, 0, 0, \dots, 0) = 0 \text{ and } OWA\ (1,1,1, \dots ,1) = 1$ ***P2***. *It is continuous.* ***P3.*** *It is monotonically increasing, that is* If $OWA(a_1, a_2, a_3, \dots, a_n)$ and $OWA(b_1, b_2, b_3, \dots, b_n)$ are pairs of n tuples satisfying the following conditions, $a_i,\ b_i \varepsilon\ [0,1]$ and $a_i \leq b_i, \forall\ i\ \varepsilon\ N$, then $OWA(a_1, a_2, a_3, \dots, a_n) \leq OWA(b_1, b_2, b_3, \dots, b_n)$ ***P4.*** *It is idempotent, that is* $OWA\ (a, a, a, \dots, a) = a \quad \forall a\ \varepsilon\ [0, 1]$ ***P5.*** *It is symmetric, that is* $\forall a\ \varepsilon[0,1]$ and $\forall p\ \varepsilon\ N$ $OWA(a_1, a_2, a_3, \dots, a_n) \leq OWA\big(a_{p(1)}, a_2, a_{p(3)}, \dots, a_{p(n)}\big)$

A very important aspect of OWA is that criteria *c* is not associated with any weight but weights are associated with the specific order of the criteria. This is achieved by sorting and rearranging of criteria into descending order.

3.3.2. Weight Calculation using Fuzzy Linguistic Quantifier

The OWA weights are calculated using three different relative fuzzy linguistic quantifiers; *at least half, at most half* and *as many as possible*. These quantifiers are associated with different values of variables *a* and *b* that are used to calculate *Q(r)*, a relative fuzzy quantifier *Q(r)*: [0, 1] → [0, 1], calculated using following equation

$$Q(r) = \begin{cases} 0 & if\ r < a \\ \frac{r-a}{b-a} & if\ a \leq r \leq b \\ 1 & if\ r > b \end{cases} \tag{8}$$

and satisfies the following properties.

Properties of relative fuzzy quantifier.
P1. *Q(0) = 0,*
P2. *∃r ∈ [0, 1] such that Q(r) =1*
P3. *a, b and r ∈ [0, 1]*
P4. The relative quantifier is monotonic and increasing such as,
∀a, b ∈ [0, 1], if a> b then Q(a) ≥ Q(b)

While, different values of variables *a* and *b* provide different fuzzy linguistic quantifiers like, *at least half* (*a*=0.5 and *b*=1), *at most half* (a=0 and b=0.5) and *as many as possible* (a=0.3 and b=0.8).

The weight vector *W* is calculated as,

$$W_i = \left\{Q\left(\frac{j}{k}\right) - Q\left(\frac{j-1}{k}\right)\right\} \tag{9}$$

where *j = 1, 2… k.*

$$W_{OWA} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ . \\ . \\ . \\ w_k \end{pmatrix} \tag{10}$$

where W_{OWA} satisfies the $w_i \,\varepsilon$ [0,1] and $\Sigma\, w_i = 1$ constraints.

- **Example 1.** For the sake of better understanding multi-criteria rating aggregation using OWA, we consider Table 1 of five different items with their ratings on four different criteria on the scale of [1-13].

Table 1 Sample criteria ratings of different items.

Items	Criteria Ratings			
	C_1	C_2	C_3	C_4
I_1	10	4	7	9
I_2	6	1	9	3
I_3	8	3	7	12
I_4	11	5	13	2
I_5	7	8	9	10

As OWA takes the values in the scale of [0-1], the ratings given in Table 1 are normalized using the following equation. The normalized ratings for each criteria are presented in Table 2.

$$r_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{11}$$

where x_{ij} is the rating to be normalized and *max*(x_j) and *min*(x_j) are the maximum and minimum ratings supported by the system, respectively.

Table 2 Normalized Criteria Ratings and aggregated score.

Items	Criteria Ratings				Aggregated scores		
	C_1	C_2	C_3	C_4	***OWA-Most***	***OWA-Many***	***OWA-Least***
I_1	0.800000	0.333333	0.000000	0.700000	0.750000	0.446667	0.166667
I_2	0.000000	0.000000	0.333333	0.100000	0.216667	0.040000	0.000000
I_3	0.400000	0.166667	0.000000	1.000000	0.700000	0.243333	0.083333
I_4	1.000000	0.500000	1.000000	0.000000	1.000000	0.650000	0.250000
I_5	0.200000	1.000000	0.333333	0.800000	0.900000	0.506667	0.266667

The following weights for different linguistic quantifiers (W_{most}, W_{many}, W_{least}) are calculated using equations (8) and (9).

$$W_{most} = [0.0, 0.0, 0.5, 0.5]$$

$$W_{many} = [0.1, 0.5, 0.4, 0.0]$$

$$W_{least} = [0.5, 0.5, 0.0, 0.0]$$

The final step in aggregation with OWA is ordering the criteria, multiply the criteria values with respective weights and aggregate the criteria using equation (7), as discussed in Section 3.3.1. The aggregated OWA ratings corresponding to each item for different linguistic quantifiers are presented in Table 2.

3.3.3. Multi-Criteria Rating Aggregation using TOPSIS

TOPSIS is one of the extensively used MCDM techniques which identifies the positive ideal and negative ideal solutions and considers the distance of alternatives from the ideal solutions for decision-making and ranking [47]. The steps for aggregating the criteria ratings using TOPSIS are given below.

Step 1. Normalize the multi-criteria decision matrix

The first step in aggregation of multi-criteria rating with TOPSIS is normalization of values of decision matrix. If there are *p* alternatives and *q* criteria in MCDM problem then the decision matrix is represented as,

$$D = \begin{bmatrix} d_{11} & \cdots & \cdots & d_{1q} \\ \vdots & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ d_{p1} & \dots & \dots & d_{pq} \end{bmatrix} \tag{12}$$

The values of the decision matrix can be normalized using following equation.

$$n_{ij} = \frac{d_{ij}}{\sqrt{\sum_{k=1}^{p} d_{ij}^2}} \tag{13}$$

Step 2. Obtain the weighted decision matrix

After normalization, weights of the criteria are calculated using some standard weight calculation method. The weights of the criteria and the normalized criteria values are used to calculate weighted normalized matrix (*WN*) as shown below.

$$WN = \begin{bmatrix} w_1 n_{11} & \cdots & \cdots & w_q n_{1q} \\ \vdots & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ w_1 n_{p1} & \dots & \dots & w_q n_{pq} \end{bmatrix} \tag{14}$$

Step 3. Determine the Ideal solutions

There are two types of criteria in a MCDM problem. The criteria with maximum preferred value are termed as benefit criteria and the criteria with minimum preferred value are termed as cost criteria. After calculating the weighted normalized matrix, the next step is to identify the cost and benefit criteria and determine the positive ideal solution (I^+) and negative ideal solution (I^-). The ideal solutions for benefit criteria are calculated using following equation.

$$I^+ = \max(w_i n_{ij}) \text{ and } I^- = \min(w_i n_{ij}) \tag{15}$$

Similarly, the ideal solutions for cost criteria are calculated using following equation.

$$I^+ = \min(w_i n_{ij}) \text{ and } I^- = \max(w_i n_{ij}) \tag{16}$$

where *i =1, 2,…, p* and *j=1, 2,…, q*

Step 4. Calculate the distance from the ideal solutions

After finding the ideal solution, the distance values of weighted normalized matrix from ideal solutions are calculated using following equations.

$$D_i^+ = \sqrt[2]{\sum_{j=1}^{q} (I^+ - w_i n_{ij})^2} \tag{17}$$

$$D_i^- = \sqrt[2]{\sum_{j=1}^{q} (I^- - w_i n_{ij})^2} \tag{18}$$

8

where $i=1, 2,..., p$, D_i^+ is the distance from the ideal positive solution and D_i^- is the distance from the ideal negative solution.

Step 5. Calculate the relative closeness from the ideal solutions

After computing the separation measures, the final step is to calculate the relative closeness (RC) of the alternatives which is calculated using the following equation.

$$RC = \frac{D_i^-}{D_i^- + D_i^+} \tag{19}$$

3.3.4. CRITIC Weights

Criteria Importance Through Inter-Criteria Correlation (CRITIC) is one of the techniques to calculate objective weights of the criteria [47]. It is based on the correlation among different criteria and standard deviation (SD) of the criteria of the decision problem. The steps to calculate the CRITIC weights are listed as follows.

Steps to calculate the CRITIC weights

Step 1. Identify the cost criteria and benefit criteria and normalize them using the following formula, respectively

$$r_{ij} = \frac{\max(x_j) - x_{ij}}{\max(x_j) - \min(x_j)}$$

$$r_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - min\ (x_j)}$$

where $i = 1,2,\dots, m$ and $j = 1,2,\dots, n$

Step 2. The correlation between different criteria is calculated using the following formula.

$$E_{ij} = \frac{\sum_{i=1}^{m}(r_{ij} - \bar{r}_j)(r_{ik} - \bar{r}_k)}{\sqrt[2]{\sum_{i=1}^{m}(r_{ij} - \bar{r}_j)^2 \sum_{i=1}^{m}(r_{ik} - \bar{r}_k)^2}}$$

where $k = 1,2,\dots,n$

Step 3. Calculate of standard deviation of each criteria using the following formula.

$$\sigma_j = \sqrt[2]{\frac{(r_{ij} - r_j)^2}{n}}$$

Step 4. Calculate the amount of information contained in criteria j using the following formula.

$$c_j = \sigma_j \sum_{k=1}^{n}(1 - E_{jk})$$

where σ_j is standard deviation of j^{th} criteria.

Step 5. Calculate the weights using the following formula.

$$W_j = \frac{c_j}{\sum_k^n c_k}$$

- **Example 2.** In order to better understand the multi-criteria aggregation using TOPSIS with CRITIC weights, we use the sample rating data for different items on different criteria as given in Table 1. The following weights are obtained by following the steps discussed in the Section 3.3.4.

$$W_{CRITIC} = [0.224129, 0.216715, 0.266750, 0.292406]$$

Now, these weights are used by TOSIS to find the aggregated ratings (presented in Table 3) of each item by following the steps discussed in the Section 3.3.3.

Table 3 Aggregated scores obtained using TOPSIS with CRITIC weight.

Items	I_1	I_2	I_3	I_4	I_5
Aggregated Score	0.511391	0.130245	0.552997	0.401048	0.698443

3.3.5. Similarity computation using aggregated scores

After aggregating multiple criteria ratings into an aggregated score, a traditional collaborative filtering technique can be realized for the similarity computation followed by neighborhood set identification. In this step, we calculated the similarity among the aggregated scores of the users through a modified Euclidian distance (MED) measure, as shown below.

$$Sim_{uv}^{MED} = \frac{1}{1 + \sqrt{\sum_{i=1}^{N}(u_i - v_i)^2}} \tag{20}$$

9

3.4. Similarity fusion and neighborhood set formation

In this step, the similarities computed using single criteria ratings and MC aggregated scores are fused together to identify the total similarity among the users. The similarities are fused with an objective to fetch better and more reliable neighbours of users by considering their preferences on both single criteria and multi-criteria levels. This fusion may also be considered as a method to overcome the multidimensionality issue that exist in the multi-dimensional recommender systems. Moreover, the benefit of this fusion is that the similarity obtained through overall ratings is augmented by the similarity obtained through the aggregated scores. This similarity can be seen as a concept related to the similarity modifier which gives weight or priorities to the calculated similarity so that the most accurate preferences of the users can be identified. The total similarity between user *u* and *v* can be computed using the following equation.

$$Sim_{uv}^{total} = Sim_{uv} \cdot Sim_{uv}^{MED} \tag{21}$$

where Sim_{uv}^{total} is total similarity between user *u* and *v*, Sim_{uv} and Sim_{uv}^{MED} are the similarities computed using overall ratings and multi-criteria aggregated scores, respectively.

After computing the total similarities among the users, the most similar users are identified for each user to make predictions and recommendations in the next step.

3.5. Prediction and Recommendation

The neighbourhood set generated in the above step is utilized to predict the ratings of unknown items for an active user using the Resnick's formula [12]. The predicted rating $R_{u,i}$ for an active user *u* on item *i* is calculated using the following Resnick's prediction formula.

$$R_{u,i} = \bar{r}_u + N_f \sum_{v=1}^{k} Sim_{uv} \times \left(r_{v,i} - \bar{r}_v\right) \tag{22}$$

where $\bar{r}_u$ is the mean of the ratings given by user *u*, *k* represents users in the neighbourhood set, Sim_{uv} is the similarity between active user *u* and neighbour user *v* and N_f is the normalization factor and is computed using following equation.

$$N_f = \frac{1}{\sum_{v=1}^{k} |Sim_{uv}|} \tag{23}$$

4. Experimental settings

The experiment to evaluate and compare the performance of our proposed RS is conducted on the Yahoo! Movie data set [43]. It is a publically available benchmark data set for MCRS. In this data set, users have rated the movies on four different criteria, namely story, acting, direction, visuals, along with the overall rating. The ratings are given on a scale of [1-13]. 1 represents least favoured and 13 represents most favoured. There are 62,156 ratings in the data set given by 6078 users on 976 movies with a sparsity of 98.95%.

For this experiment, the users with fewer ratings were discarded and users who have rated at least 20 movies were considered. Following this, there were only 19,050 ratings left given by 484 users on 945 movies. Furthermore, we split the dataset into different proportions of train and test size. These splits are termed as, Split 1 (60%-40%), Split 2 (70%-30%), Split 3 (80%-20%), and Split 4 (90%-10%). The training data is used to compute the similarity while the testing data is applied to analyse the performance of the system. We also evaluated the performance of the different approaches by varying the number of similar users in neighborhood set *k*=10, 20, 30, 40, 50, 60 and 70. We computed the precision, recall and f-measure by setting the threshold of predicted rating as 3. If the value of predicted rating is greater than or equal to 3 then it is considered as relevant rating otherwise not relevant. We evaluated and compared our proposed approaches with following MCRS approaches.

- Side Information Clustering based Multi-Criteria Recommender System (Clust-SI) [11]
- Fuzzy Side Information Clustering based Multi-Criteria Recommender System (Clust-FSI) [40]
- Common Rating weight Similarity based Multi-Criteria Recommender System (MCCRS) [13]

4.1 Evaluation Measures

Furthermore, we used different types of evaluation measures to validate the effectiveness of our proposed approach. Each of these evaluation measures is discussed below.

4.1.1 Predictive accuracy measures

These measures compute the closeness between the predicted ratings and the actual ratings. In this paper, the predictive accuracy of the RS is measured using the following evaluation measures.

10

- **Root Mean Square Error**
  The Root Mean Square Error (RMSE) is used to determine the difference between the predicted and actual rating. The RS with the lower RMSE value has preferable predictive ability and higher efficiency. If is r_i is the actual rating of an item, R_i is the predicted rating and *N* is the number of items, then the following equation can be used to calculate RMSE.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(R_i - r_i)^2} \tag{24}$$

- **Mean Absolute Error**
  Mean Absolute Error (MAE) is another error calculation measure that calculates the error between actual and predicted ratings. MAE imposes less penalty on calculated error as compared to RMSE. For a sample, if actual rating of an item is r_i and the predicted rating for the item is R_i and there are *N* items in the data set. Then MAE for the items can be computed using the equation given below.

$$MAE = \sum_{i=1}^{N}\frac{|R_i - r_i|}{N} \tag{25}$$

4.1.2 Recommendation Accuracy Metrics

These metrics are defined as the ratio of the relevant items to the recommended items. Precision, Recall, F-measure and accuracy are the four recommendation accuracy metrics used in this work.

- **Precision**
  Precision is the ratio of the number of useful recommendations to the total number of recommendations. Then precision of the RS can be found using the following equation.

$$Precision = \frac{TP}{TP+FP} \tag{26}$$

  True Positive (TP) shows number of items correctly predicted true by the system, False Positive (FP) represent number of items which are false but predicted true by the system.

- **Recall**
  Recall is a pertinent measure for evaluating the RS's performance. Recall is calculated using the following equation.

$$Recall = \frac{TP}{TP+FN} \tag{27}$$

  False Negative (FN) shows the number of items that are correctly predicted negative by the system.

- **F-Measure**
  The F-measure is a prediction metric that uses the harmonic mean of precision and recall to evaluate an RS's performance. The mathematical notation for F-measure is given below.

$$\mathrm{F-Measures} = \frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \tag{28}$$

- **Accuracy**
  Accuracy is an important evaluation metric which computes the proportion of correctly predicted ratings to the total number of ratings.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{29}$$

  where *TP* is true positive, *TN* is true negative, *FP* is false positive and *FN* is false negative.

- **Coverage**
  This metric reflects the proportion of items for which the system can generate predictions. For an active user, Coverage is calculated as the ratio of items for which the system provides predictions to the total number of unrated items. A higher Coverage indicates better system performance.

$$Coverage = \frac{\sum_{i=1}^{N} R_i}{\sum_{i=1}^{N} n_i} \tag{22}$$

  where R_i is the cardinality of predicted items for active user and *N* is the ratings in test set.

11

5. Results and Discussion

Results presented in Table 4 compare the performance of our proposed approaches with the existing MCRS approaches in terms of MAE, RMSE, accuracy, and coverage. The results are computed by varying the train-test split size while keeping the neighborhood size constant at k=40. We utilized two recently developed similarity measures, namely OS and CRS, to compute the similarities among the users. The MC ratings are aggregated using OWA with fuzzy linguistic weights and TOPSIS with CRITIC weights. The developed approaches with CRS similarity measures are termed as CMO-Most, CMO-Many, CMO-Least, and CMT-Critic. Similarly, the developed approaches with OS similarity measures are termed as OMO-Most, OMO-Many, OMO-Least, and OMT-Critic. It can be seen in Table 4 that our proposed MCRS approaches provides the best results and outperforms the existing multi-criteria approaches (Clust-SI, Clust-FSI, and MCCRS) in terms of MAE, RMSE, accuracy, and coverage. Among the CRS based and OS based approaches, the OS based approaches show superior performance in terms of MAE, RMSE, and accuracy performance measures while the CRS based approaches has the best coverage compare to all approaches. It can also be seen that our approach with OS similarity is best among all the approaches as shown in Table 4 and specifically OMO-Many show the best results on almost every experimental setting. However, there are two instances where OMO-Many does not perform the best: in terms of RMSE at Split 3, where OMT-Critic performs better and in terms of accuracy at Split 4, where OMO-Most provides the best results. Furthermore, the results of our proposed approach with the CRS similarity measure also show better performance compared to other existing approaches. While our OS based approach demonstrates lower coverage compared to the other approaches, it consistently achieves superior results in terms of MAE, RMSE and accuracy across all experimental settings. This indicates that, although our model makes predictions for a slightly smaller subset of items, it achieves the lowest error and highest accuracy on the items it does predict. Therefore, the reduced coverage is balanced by a significant gain in prediction quality, highlighting the robustness and precision of our approach in generating accurate recommendations. We have highlighted the best results in bold for each split and performance measure and to further illustrate that our second based approaches also perform better than existing MCRS approaches, these results have been underlined in the table.

Table 4 Performance Comparison of different approaches on the basis of MAE, RMSE, Accuracy and Coverage.

Evaluation	Split	Clust-SI	Clust-FSI	MCCRS	CRS based approaches				OS based approaches			
					CMO-Most	CMO-Many	CMO-Least	CMT-Critic	OMO-Most	OMO-Many	OMO-Least	OMT-Critic
MAE	60-40	0.8723	0.8709	0.802	0.7901	0.7829	0.7812	0.7839	0.771	**0.7447**	0.7722	0.7561
	70-30	0.874	0.8729	0.7973	0.7705	0.7715	0.774	0.7725	0.7619	**0.7457**	0.7653	0.7468
	80-20	0.8676	0.8761	0.8029	0.7877	0.7851	0.7834	0.7875	0.7673	**0.7464**	0.7813	0.7546
	90-10	0.8889	0.887	0.8432	0.8076	0.802	0.8069	0.8041	0.7998	**0.7627**	0.8028	0.7733
RMSE	60-40	1.0885	1.0894	1.0185	1.0021	0.9949	0.9916	0.9954	0.9874	**0.9558**	0.9884	0.971
	70-30	1.0752	1.0785	1.0001	0.9616	0.9643	0.9689	0.9645	0.956	0.9444	0.9662	**0.9409**
	80-20	1.0465	1.0558	0.9851	0.9671	0.9621	0.96	0.9663	0.9371	**0.9167**	0.9554	0.9248
	90-10	1.0248	1.0223	0.981	0.9386	0.9308	0.9351	0.9328	0.9143	**0.8755**	0.9189	0.8852
Accuracy	60-40	0.8338	0.8322	0.843	0.8481	0.8487	0.8532	0.8488	0.8514	**0.8565**	0.8514	0.8506
	70-30	0.8351	0.8335	0.8437	0.8519	0.8506	0.8503	0.8534	0.8623	**0.8678**	0.8563	0.8673
	80-20	0.8453	0.8468	0.8526	0.8592	0.8592	0.8612	0.8568	0.8749	**0.8756**	0.867	0.8698
	90-10	0.8436	0.8438	0.8546	0.8664	0.8653	0.8661	0.8697	**0.8938**	0.8872	0.8867	0.8895
Coverage	60-40	0.91	0.9087	0.9214	0.9274	0.9266	0.926	**0.9292**	0.7471	0.7295	0.724	0.7451
	70-30	0.9112	0.9087	0.9238	0.9293	0.928	0.9294	**0.9303**	0.7364	0.7265	0.7188	0.7416
	80-20	0.908	0.9048	0.9172	**0.9264**	0.9232	0.923	0.9259	0.7303	0.7166	0.7069	0.7287
	90-10	0.8968	0.8909	0.9105	**0.9134**	0.9107	0.9107	0.9112	0.7134	0.6914	0.6759	0.6989

12

Table 5 Performance Comparison of different approaches on the basis of Precision, Recall and F-Measure.

Evaluation	Split	Clust-SI	Clust-FSI	MCCRS	CRS based approaches				OS based approaches			
					CMO-Most	CMO-Many	CMO-Least	CMT-Critic	OMO-Most	OMO-Many	OMO-Least	OMT-Critic
Precision	60-40	0.9096	0.9129	0.9155	0.9219	0.9241	0.9249	0.9229	0.927	**0.9342**	0.9299	0.9292
	70-30	0.9114	0.9146	0.9161	0.9263	0.9234	0.9234	0.9268	0.9375	0.9409	0.9359	**0.9417**
	80-20	0.9143	0.9169	0.921	0.9285	0.9266	0.9311	0.9265	0.9436	**0.948**	0.9453	0.9422
	90-10	0.9149	0.9196	0.9305	0.9367	0.9373	0.9382	0.9368	**0.9537**	0.9473	0.9528	0.9532
Recall	60-40	**0.8881**	0.8825	0.8866	0.887	0.8856	0.887	0.8866	0.8852	0.884	0.884	0.8827
	70-30	0.8868	0.8845	0.8879	0.8903	0.889	0.8903	0.8912	0.8994	0.8986	0.8884	**0.9004**
	80-20	0.902	0.8988	0.8986	0.8999	0.9013	0.8993	0.8991	**0.9177**	0.9092	0.9041	0.9125
	90-10	0.916	0.9188	0.9162	0.9168	0.9162	0.9179	0.9199	0.9335	**0.9384**	0.9329	0.9323
F-Measure	60-40	0.8894	0.8864	0.8935	0.8968	0.8969	0.8981	0.897	0.8979	**0.9008**	0.8986	0.8973
	70-30	0.8874	0.8865	0.8919	0.8975	0.8955	0.8964	0.8986	0.9077	0.9093	0.9018	**0.9114**
	80-20	0.8957	0.8943	0.8971	0.9023	0.9015	0.9028	0.9004	**0.9192**	0.9167	0.9128	0.9152
	90-10	0.8964	0.9001	0.9084	0.9114	0.9119	0.9136	0.9146	**0.9301**	0.9289	0.9278	0.9282

The results presented in Table 5 compare the performance of our proposed approaches with other state-of-the-art MCRS approaches in terms of precision, recall and f-measure. It can be observed from the results that our developed approach with OS similarity outperforms the existing approaches on all measures and almost on all train-test splits except the one case where recall of Clust-SI is performing better at Split 1. Similarly, Table 5 shows that our proposed MC rating aggregation approach with CRS similarity also outperforms the existing MCRS in terms of precision, recall and f-measures except for the recall at Split 3. The results of the comparative analysis of our multi-criteria rating aggregation approach signify that our developed approach has significantly improved the performance of multi-criteria recommender system.

By seeing the above presented results, one can clearly infer that our aggregation scores based approaches are performing well compared to other existing approaches in terms of all evaluation measures. Now, in order to see the comparative performance of our developed approaches on each split and varying neighborhood size, we have presented the results in Table 6 by taking the maximum, minimum, and average results on different neighborhood set and split size. The results show that the aggregation approach with the OS similarity measure performs better than the aggregation approach with CRS similarity across all performance measures except coverage, where CRS shows a slight advantage. There are also a few instances in Table 6 suggesting that the CRS based approach has an edge over OS in specific cases, such as having a lower maximum MAE than OS. However, OS maintains lower average and minimum MAE values, which are more critical for accurate recommendations. Additionally, while the minimum recall of CMO-Least is marginally higher than that of OMO-Least, the difference is negligible. Overall, the comparison of our two developed approaches across various evaluation metrics shows that the OS based aggregation provides superior recommendation quality, while the CRS-based approach excels in coverage. Furthermore, as shown in Tables 4 and 5, both approaches consistently outperform the existing methods Clust-SI, Clust-FSI, and MCCRS across all evaluation metrics.

Table 6 Performance Comparison of various proposed approaches in terms of different parameters.

Performance M	Parameter	CMO-Most	OMO-Most	CMO-Least	OMO-Least	CMO-Many	OMO-Many	CMT-Critic	OMT-Critic
MAE	Maximum	**0.818**	0.8568	**0.808**	0.8737	**0.8027**	0.8092	**0.8145**	0.8306
	Minimum	0.7691	**0.7514**	0.7696	**0.7516**	0.7668	**0.7376**	0.7681	**0.7306**
	Average	0.7915	**0.7803**	0.7888	**0.7825**	0.7853	**0.7626**	0.7875	**0.7638**
RMSE	Maximum	1.0093	**1.0004**	1.0149	**0.9892**	1.0009	**0.9903**	1.001	**0.9791**
	Minimum	0.9001	**0.893**	0.9033	**0.8859**	0.8862	**0.8627**	0.8797	**0.8741**
	Average	0.9667	**0.9459**	0.9638	**0.9485**	0.9594	**0.9266**	0.9627	**0.9262**
Precision	Maximum	0.9609	**0.975**	0.9632	**0.9744**	0.962	**0.9857**	0.963	**0.977**
	Minimum	0.9184	**0.9233**	0.9198	**0.9251**	0.9209	**0.9268**	0.9212	**0.927**
	Average	0.9309	**0.9427**	0.9310	**0.9435**	0.9315	**0.9464**	0.9315	**0.9441**
Recall	Maximum	0.9255	**0.9564**	0.9316	**0.9595**	0.922	**0.9661**	0.9206	**0.9578**
	Minimum	0.8792	**0.8852**	**0.8784**	0.8781	0.8804	**0.8832**	0.8785	**0.8827**
	Average	0.8994	**0.9134**	0.8988	**0.9067**	0.8986	**0.9101**	0.8992	**0.9120**
F-	Maximum	0.9317	**0.9556**	0.934	**0.957**	0.9283	**0.9692**	0.9281	**0.9578**
	Minimum	0.8943	**0.8963**	0.8937	**0.894**	0.8955	**0.8967**	0.8947	**0.8972**
	Average	0.9035	**0.9171**	0.9033	**0.9134**	0.9033	**0.9171**	0.9039	**0.9170**
Accuracy	Maximum	0.8934	**0.9334**	0.902	**0.9377**	0.8968	**0.953**	0.8921	**0.936**
	Minimum	0.8457	**0.8476**	0.8466	**0.8488**	0.8487	**0.8514**	0.8477	**0.8506**
	Average	0.8579	**0.8744**	0.8583	**0.8709**	0.8584	**0.8764**	0.8583	**0.8754**
Coverage	Maximum	**0.9682**	0.855	**0.9688**	0.8535	**0.9696**	0.8505	**0.9689**	0.8592
	Minimum	**0.6845**	0.3679	**0.6775**	0.3471	**0.6797**	0.3326	**0.6877**	0.3647
	Average	**0.8891**	0.6887	**0.8878**	0.6692	**0.8873**	0.6581	**0.8898**	0.6822

The results presented in Table 4 to Table 6 signify that our developed approach with OS similarity measure gives the best performance among all other comparative approaches, including our developed aggregation with CRS similarity approach. Therefore, we now perform further analysis of the most efficient approach to see its performance by varying the neighborhood and split size. Figure 4 shows the variation in average MAE with respect to the variation in neighborhood size and split. It shows that MAE of all four approaches with OS similarity improves with the increase in neighborhood size and average MAE of OMO-Many is best on most of the cases. Similarly, the variation in average MAE with respect to change in split shows that the developed approaches with OS similarity give best performance at Split 2 and worst performance of all approaches is observed at Split 4. Furthermore, it shows that OMT-Critic gives best performance on most experimental settings.

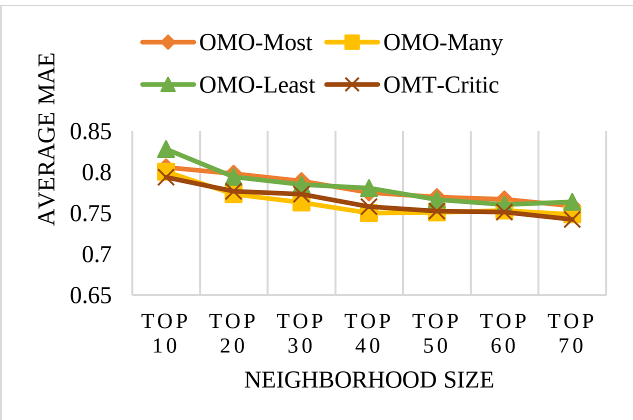


Figure 4 Comparison of average MAE on varying neighborhood size and splits.

Similarly, Figure 5 shows the variation in average RMSE with change in neighborhood size and split. The nature of curves in Figure 5 shows that RMSE does not follow a uniform pattern with the change in neighborhood size. But, it shows that RMSE is best for all approaches for neighborhood size k=70 and OMT-Critic gives best results in most of the instances. The variation of average RMSE with change in split follows a uniform pattern with worst performance at Split 1 and best performance at Split 4. Figure 5 also shows that the performance of OMT-Critic is best for all split and at all neighborhood size except k=30.

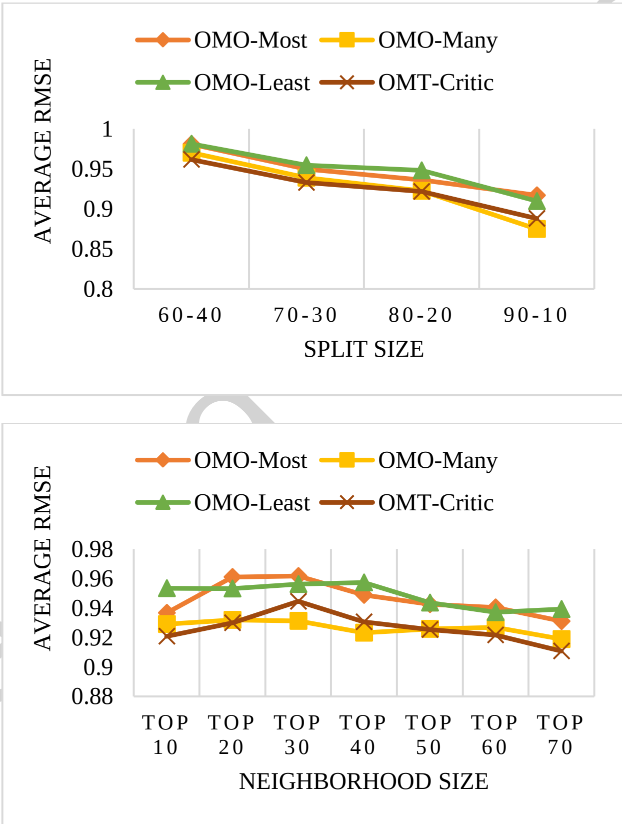


Figure 5 Comparison of average RMSE on varying neighborhood size and splits.

The variation in terms of average precision with respect to changes in neighborhood size and split is shown in Figure 6. It can be observed that the average precision of the developed approach decreases with an increase in neighborhood size and best performance is shown by OMO-Many at k=10. It can also be observed from Figure 6 that average precision increases with the increase in training set and best performance for all developed approaches is obtained at Split 4. Moreover, the performance of OMO-Many is best at all splits.

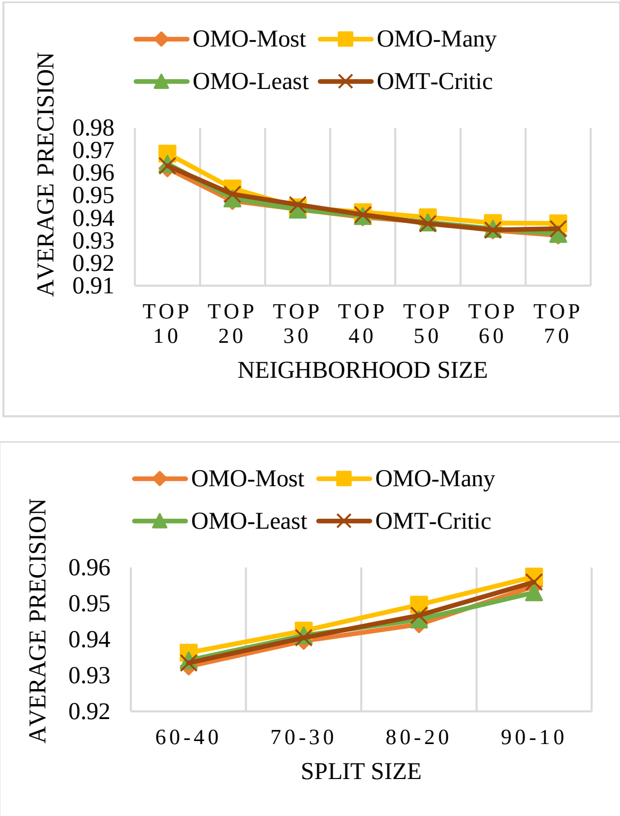


Figure 6 Comparison of average Precision on varying neighborhood size and splits.

We present the variation in average recall with respect to change in neighborhood size and splits in Figure 7. In can be seen from Figure 7, the average recall of the developed approach is inversely proportional to neighborhood size and directly proportional to the training size. Furthermore, it can also be seen that the performance of OMO-Most is best at most of the neighborhood size and most of the splits.

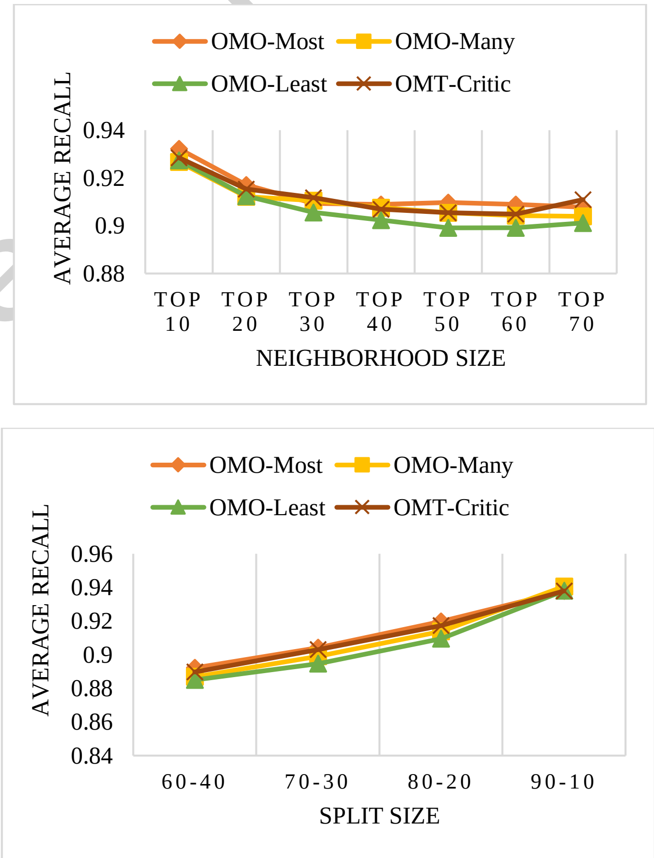


Figure 7 Comparison of average Recall on varying neighborhood size and splits.

16

Figure 8 depicts the variation in average F-Measure with respect to change in neighborhood set and split. It shows that F-Measure follow the similar trend as shown by precision and recall. It also depicts that the performance of OMO-Least is worst at all neighborhood size at all splits and the performance of OMO-Many is best at Split 4.

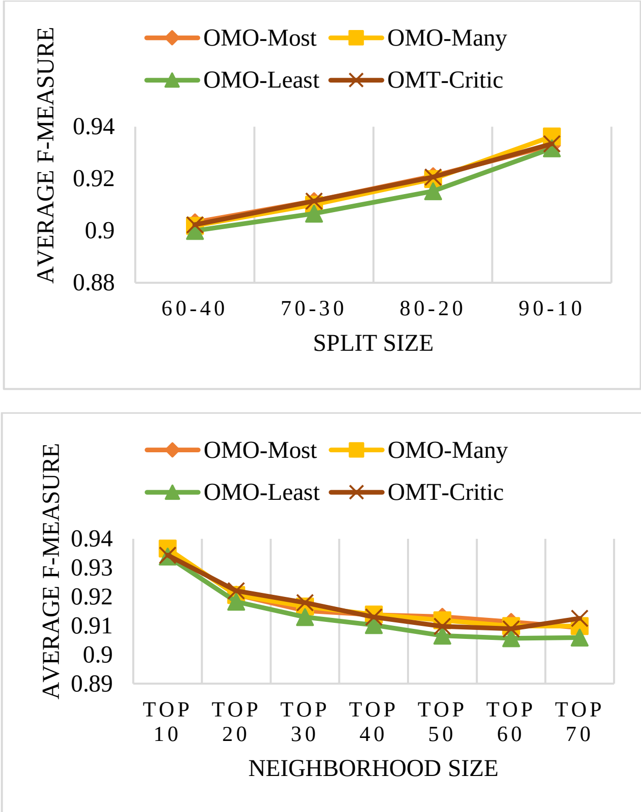
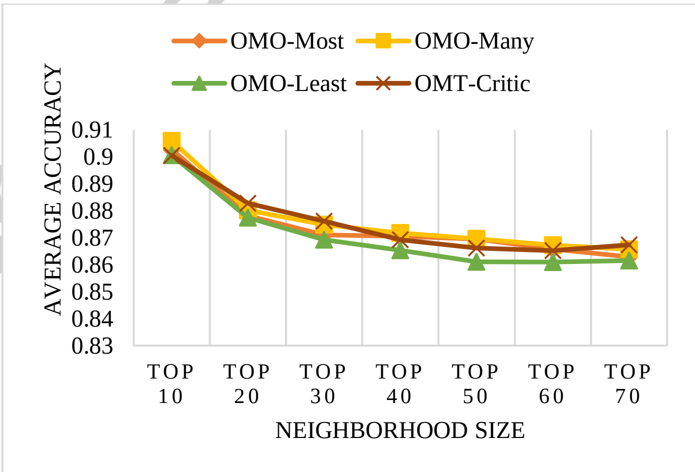


Figure 8 Comparison of average F-measure on varying neighborhood size and splits.

We also analyzed the performance in terms of accuracy. It can be seen from Figure 9 that accuracy of all developed approaches improves with decrease in neighborhood size. Moreover, it can also be seen that accuracy increases with increase in training size.



17

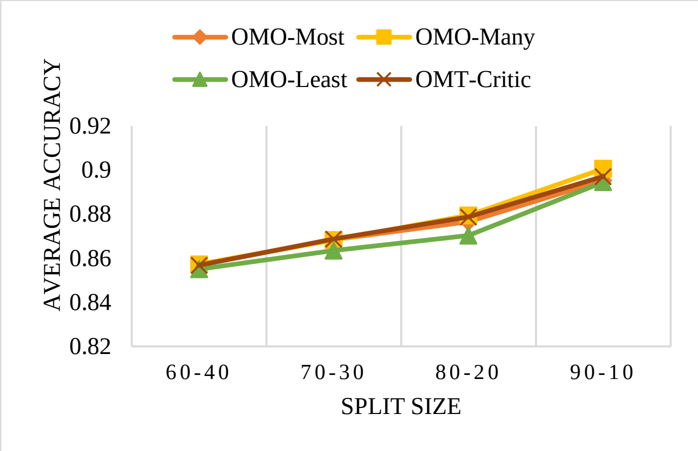


Figure 9 Comparison of average Accuracy on varying neighborhood size and splits.

The results presented in the Figure 10 compare different CRS based and OS based approaches for correct predictions percentage using maximum, minimum, and average parameters. Correct predictions mean that the predicted rating matches the actual product rating. The results show that the OS based approaches outperform the CRS based approaches across all comparison parameters. The OS based methods consistently provide better overall predictions, demonstrating superior accuracy and reliability. While the CRS based methods perform reasonably, they are not as effective as the OS based approaches, which excel in both peak performance and worst case scenarios. Overall, the OS based approaches are more effective across all evaluation measures.

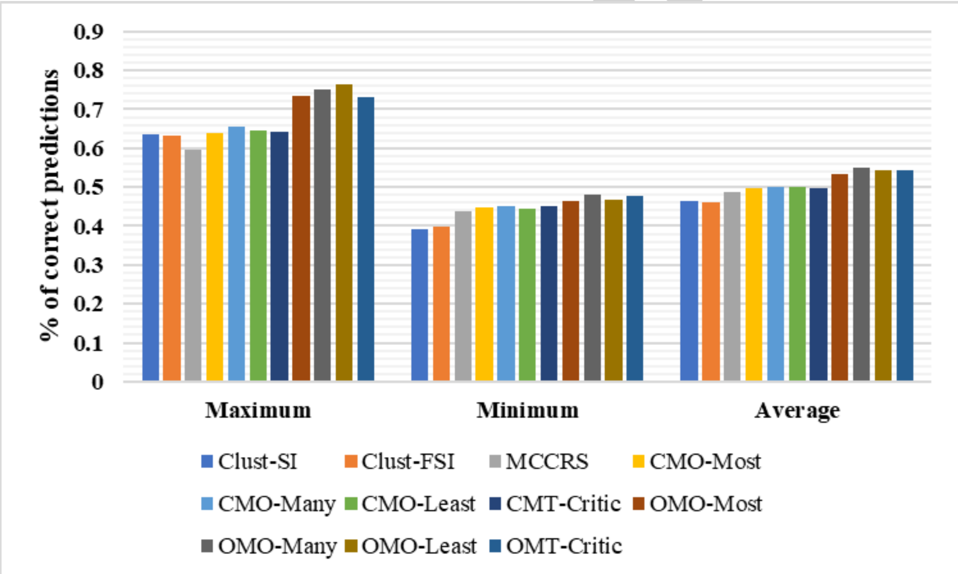


Figure10 Correct predictions percentage for CRS based and OS based approaches.

The discussion of the presented results is concluded with the following points.

- The developed criteria rating aggregation based MCRS outperformed the existing MCRS on various evaluation measures.
- The performance of the developed MCRS with OS similarity measure gives best results except coverage performance measure.
- The use of OWA with linguistic weights and TOPSIS with objective weights have improved the performance of MCRS and reduced the uncertainty, inconsistency and vagueness in aggregation process.
- The total similarity measure gives best results because of its working similar to an augmented similarity measure where the similarity computed using aggregated score is acting as a similarity modifier to the similarity obtained through the overall ratings.
- The aggregation approach aggregates the various criteria together and thus reducing the dimensionality at the feature level. While the total similarity is obtained by combining the two similarities together and thus avoiding the multidimensionality during the similarity computation.
- The results show that the fusion of local similarities to obtain total similarities significantly improves the performance of the multi-criteria recommender system on various measures.

18

6. Conclusion

In this paper, we developed a framework for multi-criteria recommender systems to overcome the multidimensionality issue by utilizing different rating aggregation methods known as OWA and TOPSIS. These aggregation methods were used to consider the conflicting nature of different features and to preserve the importance of individual criteria. In general, the multidimensionality is one of the severe issues that may arise while incorporating the multi-criteria ratings into the traditional recommender systems. Our developed approaches avoid this issue by aggregating the various criteria and computed similarity using the aggregated scores obtained through OWA and TOPSIS methods. This calculated similarity was then fused with the similarity obtained using overall ratings and developed a total similarity. This fusion of similarities helped in identifying the most accurate neighborhood set and thus improved the prediction accuracy of the systems. The experiments conducted on Yahoo! Movies data set and comparison with state-of-the-art approaches on various evaluation measures show the effectiveness of our proposed approaches. Also, the aggregation based approaches significantly improved the performance of the OS and CRS similarity measures when MC ratings aggregated using OWA and TOPSIS methods. This shows the success of our aggregation based approaches over the traditional approaches. In future, one can extend the proposed work by developing new similarity measures or by designing efficient similarity modifiers for the OS and CRS measures, which can be used for both overall and aggregated score similarity calculations. Moreover, formation of different neighborhood set using both the similarities separately and linking them together using some technique can also be explored in future work in this direction.

Declarations

Ethics approval: Not Applicable

Data Availability: The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflict of Interests: The authors declare that they have no conflict of interest.

Funding: Not Applicable

Author Contributions: [Khalid Anwar and Mohammed Wasid] Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **[Aasim Zafar and Arshad Iqbal]** Resources, Visualization, Supervision, Writing – review & editing.

References

[1] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst.*, vol. 74, pp. 12–32, 2015, doi: 10.1016/j.dss.2015.03.008.

[2] K. Anwar, J. Siddiqui, and S. S. Sohail, "Machine learning-based book recommender system: A survey and new perspectives," *International Journal of Intelligent Information and Database Systems*, vol. 13, no. 2–4. pp. 231–248, 2020. doi: 10.1504/IJIIDS.2020.109457.

[3] W. Wang, G. Zhang, and J. Lu, "Collaborative filtering with entropy-driven user similarity in recommender systems," *Int. J. Intell. Syst.*, vol. 30, no. 8, pp. 854–870, 2015, doi: 10.1002/int.21735.

[4] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," in *UAI'98: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 1998, pp. 43–52. [Online]. Available: http://arxiv.org/abs/1301.7363

[5] M. Wasid and K. Anwar, "Incorporating Contextual Information and Feature Fuzzification for Effective Personalized Healthcare Recommender System," pp. 197–211, 2023, doi: 10.1007/978-3-031-33183-1_11.

[6] E. Yalcin and A. Bilge, "A novel target item-based similarity function in privacy-preserving collaborative filtering," *J. Supercomput.*, vol. 80, no. 13, pp. 19433–19461, 2024, doi: 10.1007/s11227-024-06221-7.

[7] K. Anwar, J. Siddiqui, and S. Saquib Sohail, "Machine Learning Techniques for Book Recommendation: An Overview," *SSRN Electron. J.*, pp. 1291–1297, 2019, doi: 10.2139/ssrn.3356349.

[8] S. S. Sohail, J. Siddiqui, and R. Ali, "Classifications of recommender systems: A review," *J. Eng. Sci. Technol. Rev.*, vol. 10, no. 4, pp. 132–153, 2017, doi: 10.25103/jestr.104.18.

[9] R. Burke, "Hybrid Recommender Systems : Survey and Rxperiments," *User Model. User-Adapted Interact.*, vol. 12, pp. 331–370, 2002.

[10] E. Yalcin and A. Bilge, "Evaluating unfairness of popularity bias in recommender systems: A comprehensive user-centric analysis," *Inf. Process. Manag.*, vol. 59, no. 6, p. 103100, 2022, doi: 10.1016/j.ipm.2022.103100.

[11] M. Wasid and R. Ali, "Multi-criteria clustering-based recommendation using Mahalanobis distance," *Int. J. Reason. Intell. Syst.*, vol. 12, no. 2, pp. 96–105, 2020, doi: 10.1504/IJRIS.2020.106803.

[12] K. Anwar, A. Zafar, and A. Iqbal, "An efficient approach for improving the predictive accuracy of multi-criteria recommender system," *Int. J. Inf. Technol.*, 2023, doi: 10.1007/s41870-023-01547-6.

[13] M. Wasid and R. Ali, "A frequency count approach to multi-criteria recommender system based on criteria weighting

using particle swarm optimization," *Appl. Soft Comput.*, vol. 112, p. 107782, 2021, doi: 10.1016/j.asoc.2021.107782.

[14] G. Adomavicius and Y. O. Kwon, "New recommendation techniques for multicriteria rating systems," *IEEE Intell. Syst.*, vol. 22, no. 3, pp. 48–55, 2007, doi: 10.1109/MIS.2007.58.

[15] M. Wasid, R. Ali, and S. Shahab, "Adaptive genetic algorithm for user preference discovery in multi-criteria recommender systems," *Heliyon*, vol. 9, no. 7, p. e18183, 2023, doi: 10.1016/j.heliyon.2023.e18183.

[16] G. Adomavicius, N. Manouselis, and Y. Kwon, "Multi-Criteria Recommender Systems," *Recomm. Syst. Handb.*, pp. 769–803, 2011, doi: 10.1007/978-0-387-85820-3_24.

[17] Z. Batmaz and C. Kaleli, "AE-MCCF: An Autoencoder-Based Multi-criteria Recommendation Algorithm," *Arab. J. Sci. Eng.*, vol. 44, no. 11, pp. 9235–9247, 2019, doi: 10.1007/s13369-019-03946-z.

[18] V. Kant, T. Jhalani, and P. Dwivedi, "Enhanced multi-criteria recommender system based on fuzzy Bayesian approach," *Multimed. Tools Appl.*, vol. 77, no. 10, pp. 12935–12953, 2018, doi: 10.1007/s11042-017-4924-2.

[19] M. Hassan, "Enhancing Learning Objects Recommendation Using Multi-criteria Recommender Systems," in *IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2016, pp. 62–64.

[20] H. Mohamed, L. Abdulsalam, and H. Mohammed, "Adaptive genetic algorithm for improving prediction accuracy of a multi-criteria recommender system," *Proc. - 2018 IEEE 12th Int. Symp. Embed. Multicore/Many-Core Syst. MCSoC 2018*, pp. 79–86, 2018, doi: 10.1109/MCSoC2018.2018.00025.

[21] M. Hassan and M. Hamada, "Evaluating the performance of a neural network-based multi-criteria recommender system," *Int. J. Spat. Data Sci.*, vol. 1, no. 1, p. 54, 2019, doi: 10.1504/ijstds.2019.10018848.

[22] M. Hamada and M. Hassan, "Artificial Neural Networks and Particle Swarm Optimization Algorithms for Preference Prediction in Multi-Criteria Recommender Systems," *informatics*, vol. 5, no. 25, pp. 1–16, 2018, doi: 10.3390/informatics5020025.

[23] Y. Zheng, "Utility-based multi-criteria recommender systems," *Proc. ACM Symp. Appl. Comput.*, vol. Part F1477, pp. 2529–2531, 2019, doi: 10.1145/3297280.3297641.

[24] A. Kouadria, O. Nouali, and M. Y. H. Al-Shamri, "A Multi-criteria Collaborative Filtering Recommender System Using Learning-to-Rank and Rank Aggregation," *Arab. J. Sci. Eng.*, vol. 45, no. 4, pp. 2835–2845, 2020, doi: 10.1007/s13369-019-04180-3.

[25] S. Gupta and V. Kant, "Credibility score based multi-criteria recommender system," *Knowledge-Based Syst.*, vol. 196, no. xxxx, p. 105756, 2020, doi: 10.1016/j.knosys.2020.105756.

[26] S. Gupta and V. Kant, "An aggregation approach to multi-criteria recommender system using genetic programming," *Evol. Syst.*, vol. 11, no. 1, pp. 29–44, 2020, doi: 10.1007/s12530-019-09296-3.

[27] S. Gupta and V. Kant, "A model-based approach to user preference discovery in multi-criteria recommender system using genetic programming," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 11, pp. 1–17, 2022, doi: 10.1002/cpe.6899.

[28] E. T. Demirkiran, M. Y. Pak, and R. Cekik, "Multi-criteria collaborative filtering using rough sets theory," *J. Intell. Fuzzy Syst.*, pp. 1–11, 2020, doi: 10.3233/jifs-201073.

[29] M. Nilashi, O. Bin Ibrahim, and N. Ithnin, "Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and Neuro-Fuzzy system," *Knowledge-Based Syst.*, vol. 60, pp. 82–101, 2014, doi: 10.1016/j.knosys.2014.01.006.

[30] M. Nilashi, O. Bin Ibrahim;, and R. Z. Norafida Ithnin, "A multi-criteria recommendation system using dimensionality reduction and Neuro-Fuzzy techniques," *Soft Comput.*, vol. 7, no. 1, pp. 55–64, 2014, doi: 10.1007/s00500-014-1475-.

[31] M. Nilashi, D. Jannach, O. Bin Ibrahim, and N. Ithnin, "Clustering- and regression-based multi-criteria collaborative filtering with incremental updates," *Inf. Sci. (Ny).*, vol. 293, pp. 235–250, 2015, doi: 10.1016/j.ins.2014.09.012.

[32] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems," *Knowledge-Based Syst.*, vol. 211, p. 106545, 2021, doi: 10.1016/j.knosys.2020.106545.

[33] R. Rismala, N. U. Maulidevi, and K. Surendro, "Personalized neural network-based aggregation function in multi-criteria collaborative filtering," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 1, p. 101922, 2024, doi: 10.1016/j.jksuci.2024.101922.

[34] K. Zhang, X. Liu, W. Wang, and J. Li, "Multi-criteria recommender system based on social relationships and criteria preferences," *Expert Syst. Appl.*, vol. 176, no. March, p. 114868, 2021, doi: 10.1016/j.eswa.2021.114868.

[35] K. Palanivel and R. Sivakumar, "A study on collaborative recommender system using fuzzy-multicriteria approaches," *Int. J. Bus. Inf. Syst.*, vol. 7, no. 4, pp. 419–439, 2011, doi: 10.1504/IJBIS.2011.040566.

[36] M. Nilashi, O. Bin Ibrahim, and N. Ithnin, "Hybrid recommendation approaches for multi-criteria collaborative filtering," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3879–3900, 2014, doi: 10.1016/j.eswa.2013.12.023.

[37] M. Nilashi, O. Bin Ibrahim, N. Ithnin, and N. H. Sarmin, "A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA-ANFIS," *Electron. Commer. Res. Appl.*, vol. 14, no. 6, pp. 542–562, 2015, doi: 10.1016/j.elerap.2015.08.004.

[38] N. R. Kermany and S. H. Alizadeh, "A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques," *Electron. Commer. Res. Appl.*, vol. 21, pp. 50–64, 2017, doi: 10.1016/j.elerap.2016.12.005.

[39] M. Wasid and R. Ali, "An improved recommender system based on multi-criteria clustering approach," *Procedia Comput. Sci.*, vol. 131, pp. 93–101, 2018, doi: 10.1016/j.procs.2018.04.190.

[40] M. Wasid and R. Ali, "Fuzzy Side Iinformation Clustering-based Framework for Effective Recommendations," *Comput. Informatics*, vol. 38, pp. 597–620, 2019, doi: 10.31577/cai.

[41] E. Yalçin and A. BiLge, "Binary multicriteria collaborative filtering," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 28, no. 6, pp. 3419–3437, 2020, doi: 10.3906/ELK-2004-184.

20

[42] Z. Batmaz and C. KALELİ, “A hybrid multi-criteria recommendation algorithm based on autoencoders,” *Pamukkale Univ. J. Eng. Sci.*, vol. 30, no. 2, pp. 212–221, 2024, doi: 10.5505/pajes.2023.68253.
[43] K. Lakiotaki, N. F. Matsatsinis, and A. Tsoukiàs, “Multicriteria user modeling in recommender systems,” *IEEE Intell. Syst.*, vol. 26, no. 2, pp. 64–76, 2011, doi: 10.1109/MIS.2011.33.
[44] A. Gazdar and L. Hidri, “A new similarity measure for collaborative filtering based recommender systems,” *Knowledge-Based Syst.*, vol. 188, p. 105058, 2020, doi: 10.1016/j.knosys.2019.105058.
[45] K. Anwar, A. Zafar, A. Iqbal, and S. S. Sohail, “An OWA-Based Feature Extraction and Ranking for Performance Evaluation of the Players in Cricket,” *Lect. Notes Data Eng. Commun. Technol.*, vol. 106, pp. 169–179, 2022, doi: 10.1007/978-981-16-8403-6_15.
[46] K. Anwar *et al.*, “Artificial Intelligence-Driven Approach to Identify and Recommend the Winner in a Tied Event in Sports Surveillance,” *Fractals*, 2023, doi: 10.1142/S0218348X23401497.
[47] K. Anwar, A. Zafar, and A. Iqbal, “Neutrosophic MCDM Approach for Performance Evaluation and Recommendation of Best Players in Sports League,” *Int. J. Neutrosophic Sci.*, vol. 20, no. 01, pp. 128–149, 2023.

21