

Received 3 August 2022, accepted 17 August 2022, date of publication 26 August 2022, date of current version 2 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3201821

RESEARCH ARTICLE

Multi-Criteria Ranking: Next Generation of Multi-Criteria Recommendation Framework

YONG ZHENG^{ID1}, (Member, IEEE), AND DAVID WANG²

¹College of Computing, Illinois Institute of Technology, Chicago, IL 60616, USA

²Morningstar Inc., Chicago, IL 60602, USA

Corresponding author: Yong Zheng (yzheng66@iit.edu)

ABSTRACT Recommender systems have been developed to assist decision making by recommending a list of items to the end users. The multi-criteria recommender system (MCRS) is a special type of recommender systems, where user preferences on multiple criteria can be taken into account in recommendation models. Traditional algorithms for MCRS usually predict user ratings on these criteria, and finally estimate the overall rating by different aggregation functions. In this paper, we propose a novel multi-criteria recommendation framework, *Multi-Criteria Ranking*, where we can directly infer a ranking score for an item candidate from the predicted ratings on multiple criteria. The proposed framework is general enough and most of the existing algorithms in MCRS can be easily integrated with our framework. Our experimental results can demonstrate the effectiveness of the proposed framework by evaluating top-*N* recommendations over multiple real-world data sets. We believe that multi-criteria ranking opens the door to develop more effective and promising multi-criteria recommendation models.

INDEX TERMS Multi-criteria, decision making, recommender system, Pareto ranking, multi-criteria ranking.

I. INTRODUCTION

Recommender systems can deliver a list of recommendations tailored by user preferences. It has been widely applied in real-world applications, such as Amazon.com in the e-commerce domain [1], [2], Netflix in online streaming [3], [4], Twitter in social media [5], [6], and so forth.

Traditional recommender systems may be built by learning a model from a standard data set which contains user, item and user preferences (e.g., ratings, click-through, etc.) on the items. Multi-criteria recommender systems (MCRS) [7], [8] is a special type of the recommender systems in which we can additionally take the user preferences on different aspects of the items (i.e., criteria) into account to improve the quality of the recommendations. MCRS have been built and utilized in several real-world applications, e.g., hotel bookings at TripAdvisor.com, movie reviews at Yahoo!Movie, restaurant feedbacks at OpenTable.com.

An example from OpenTable.com can be shown by Figure 1. In addition to the overall rating given by a user on

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.



FIGURE 1. Example from OpenTable.com.

a restaurant, users can leave additional ratings on different criteria, including the *food* tastes, *service* quality, *ambience* of the restaurant, the *value* of expenses, etc. MCRS can be built

by taking advantage of these multi-criteria ratings, in order to deliver more effective recommendations.

An example of data in MCRS can be shown by Table 1. The *rating* column refers to the users' overall rating on the items. We also have users' ratings on multiple criteria, such as food, service, ambience and value.

TABLE 1. Example of rating matrix from OpenTable.

User	Item	Rating	Food	Service	Ambience	Value
U_1	T_3	4	4	3	4	4
U_2	T_2	3	3	3	3	3
U_3	T_1	?	?	?	?	?

As a result, the prediction task in MCRS can be referred as how to predict U_3 's rating on the item T_1 without known multi-criteria ratings on the item. The predicted overall rating can also be utilized to sort and rank item candidates, in order to produce a list of top- N recommendations.

Therefore, most of the traditional recommendation algorithms in MCRS tried to build models to predict multi-criteria ratings which will further be used to estimate the overall rating through aggregation functions. By contrast, we propose a novel multi-criteria recommendation framework, *Multi-Criteria Ranking*, which can directly produce a ranking score for an item candidate. The contributions of this paper can be summarized as follows.

- We utilized and examined the technique of Pareto ranking [31] from multi-objective evolutionary algorithms [35] for the purpose of multi-criteria ranking, and it was demonstrated as an effective method for multi-criteria ranking.
- The idea of Pareto ranking is aligned with multi-criteria decision-making theories. There are no learning or optimization processes, and we can demonstrate that it outperforms the traditional methods using rating aggregations where an optimization process is involved.
- Our proposed framework is general enough, and most existing multi-criteria recommendation models can be easily integrated into the framework.
- Our experimental results demonstrate the effectiveness of the proposed framework over four real-world data sets. We consider multi-criteria ranking as the next generation of multi-criteria recommendation framework which open the doors to develop more promising recommendation models in MCRS.

The remainder of this paper is organized as follows. Section II positions the related work. Section III delivers the preliminary introduction of neural matrix factorization technique, and its applications in MCRS. Section IV discusses our proposed multi-criteria recommendation framework. Section V presents the experimental results, followed by the conclusions and future work in Section VI.

II. RELATED WORK

In this section, we first introduce the development of traditional MCRS. In addition, we discuss the deep learning based MCRS.

A. TRADITIONAL MCRS

In MCRS, we need to estimate a user's multi-criteria ratings on an item, and then aggregate these ratings to finally predict the overall rating, as shown in Equation 1. We use R_0 to denote the overall rating, and R_1, R_2, \dots, R_k as the multi-criteria ratings, while the function f is denoted as the aggregation function.

$$R_0 = f(R_1, R_2, \dots, R_k) \quad (1)$$

Adomavicius *et al.* [7] categorized the solutions in MCRS into two folds – one is the *heuristic approach* [7], [9] where we can better calculate user-user or item-item similarities by taking these multi-criteria ratings into consideration, so that better neighborhoods can be identified for traditional collaborative filtering techniques. Another one is the *model-based approach* [7], [10], [11] in which we can predict the multi-criteria ratings through model-based approaches (e.g., matrix factorization [12], tensor factorization [13], [14] or neural network based models [15], [16], [17], [18]) and finally estimate the overall rating by an aggregation function or other modeling methods.

In the process of multi-criteria rating predictions, Adomavicius *et al.* [7] proposed an independent approach, where we can build an individual model to predict user preferences in each criterion. Any traditional recommenders, e.g., user-based collaborative filtering and matrix factorization, can be utilized as the independent model to predict users' ratings on each criterion. However, These independent methodologies ignored the dependency among criteria. For example, the price of the room may be higher if the hotel is located in a better location. By contrast, other researchers tried to improve multiple criteria rating predictions, by taking the correlation or dependency among criteria into account. Sahoo *et al.* [10] proposed a probabilistic approach based on the flexible mixture model, where the dependency of criteria can be incorporated into the probabilistic structure of the flexible mixture model. Zheng *et al.* [11] developed the technique of criteria chains, where we can predict multi-criteria ratings by following a predefined sequence of criteria in the chain. Later on, there are deep learning-based recommendation models [17] which can share the same user and item embeddings to predict ratings on multi-criteria simultaneously, and the models can be optimized through joint learning where the dependency of criteria is considered.

In terms of the aggregation function for estimating the overall ratings, The linear regression [7] and support vector regressions [19] were the most popular choices. Hassan *et al.* [15], [17] proposed to utilize the neural network models to predict the overall rating, since it is able to better capture the non-linear relations between the overall rating and multi-criteria ratings.

Other researchers seek other different approaches to assist multi-criteria decision-making and recommendations. For example, some researchers tried to combine context-awareness with multi-criteria recommendations [20], [21], [22]. Zhang *et al.* enhanced the accuracy of predicted

criteria ratings by utilizing trust propagation inferred from social relationships [23]. Aysha *et al.* proposed to utilize non-dominated neighbors selected from multi-criteria ratings to enhance the quality of neighborhood-based multi-criteria recommendations [24]. Zheng *et al.* built a utility-based recommendation model [25] which derives the utility of an item from the distance between user preferences and user expectations on multiple criteria. Some others also made the attempts to develop multi-criteria recommender systems for novel applications or domains (e.g., online grocery [26], education [21], pandemic lockdown decisions [27]), rather than the traditional hotel and restaurant domains.

B. DEEP LEARNING BASED MCRS

With the advances in deep learning, the neural network-based recommender systems [28] were well developed recently. These deep learning-based recommendation models were also reused to assist the recommendations in MCRS, especially the neural matrix factorization (NeuMF) model [29].

Hassan *et al.* [15] made the first attempt, where they utilized neural networks in the process of rating aggregations only. More specifically, they utilized an artificial neural network to predict the overall rating from the multi-criteria ratings which were predicted based on singular value decomposition. Batmaz *et al.* [16] used neural network models in the steps of both multi-criteria rating predictions and overall rating aggregations. They applied an auto-encode model to predict a user's rating on each criterion independently, and utilized a multi-layer perceptron (MLP) to estimate the overall rating from predicted multi-criteria ratings. Again, researchers realized that the correlation or dependency among criteria could not be ignored, and they tried to propose neural network models which can take these dependencies into account. Nassar *et al.* [17] proposed a NeuMF model with multiple outputs, where each output refers to a predicted rating on a criterion. By this way, the predictive model for multi-criteria ratings can share the same embeddings for users and items, and the final loss of the model is actually a joint loss function of the loss in multiple criteria ratings. They also utilized MLP to capture the relationships between overall and multi-criteria ratings, so that the predicted multi-criteria ratings can be sent to MLP to finally estimate the overall rating.

It is clear that most of these proposed models rely on an aggregation function (e.g., the MLP model) to capture the relationships between overall and multi-criteria ratings, so that we can estimate the overall rating from the predicted multi-criteria ratings. These estimated overall ratings, therefore, can be finally utilized to sort items in order to produce top- N item recommendations. In this paper, we propose to ignore the overall rating predictions, but directly derive a ranking score from the predicted multi-criteria ratings for the purpose of top- N multi-criteria recommendations.

III. PRELIMINARY: NeuMF AND MCRS

Prior to our proposed multi-criteria recommendation framework, we deliver the preliminary introductions of the NeuMF model and its extensions in MCRS, since we utilize them as baseline approaches in our analysis.

A. NEURAL MATRIX FACTORIZATION

The NeuMF recommendation model [29] is a popular neural network-based recommendation approach. It can be described by a two-tower neural network structure, as depicted by Figure 2.

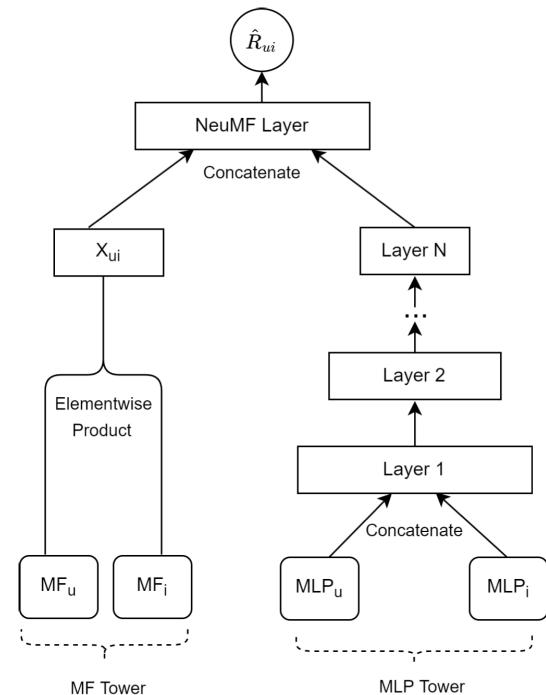


FIGURE 2. The NeuMF model.

The tower on the left is the matrix factorization (MF) tower, where MF_u and MF_i represent the embedding vector for a user u and an item i , respectively. X_{ui} is the result by the elementwise product of the MF_u and MF_i embeddings. The operation in this tower is known as generalized matrix factorization.

The tower on the right is the classical MLP model. The embeddings for user u and item i (i.e., MLP_u and MLP_i) will be concatenated together and entered the MLP structure.

Finally, the output by the MF and MLP towers will be concatenated and sent to the activation function to predict the rating given by u on item i (i.e., \hat{R}_{ui}). In a rating prediction task, the LeakyReLU is usually utilized as the activation function, since it allows a small, non-zero, constant gradient so that it avoids the dying end issue in the original ReLU function. The mean squared error (MSE) can be used as the loss function which is shown by Equation 2. We use T_r to denote the training set, where R_{ui} and \hat{R}_{ui} are used to represent the real and predicted rating, respectively. By this way, NeuMF

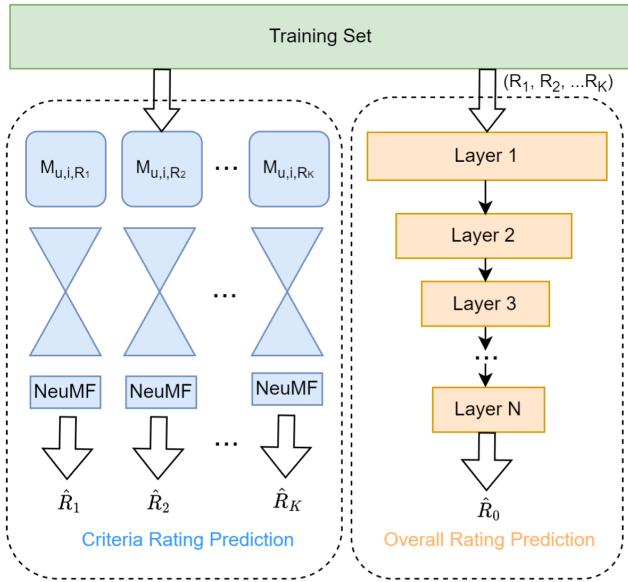


FIGURE 3. The IndNeuMF model.

can train and learn the models by using appropriate optimizer (e.g., adam [30]) and minimizing MSE.

$$MSE = \frac{1}{|T_r|} \sum_{(u,i) \in T_r} (R_{u,i} - \hat{R}_{u,i})^2 \quad (2)$$

B. EXTENDING NeuMF FOR MCRS

The popular NeuMF model above can be reused or extended to serve for multi-criteria recommendations. In this section, we introduce two variants for MC RS based on the NeuMF recommendation model.

The first approach can be considered as an independent NeuMF (IndNeuMF) recommendation model, in which we utilize a NeuMF model to predict user preferences on each criterion individually, as depicted by Figure 3. IndNeuMF is an approach similar to the work by Batmaz *et al.* [16], where they used auto-encoder as the individual recommendation model.

As shown by Figure 3, IndNeuMF is composed of two models. The first model is used to predict ratings in multiple criteria. More specifically, we decompose the multi-criteria rating matrix into individual matrices which are associated with ratings in each criterion, i.e., M_{u,i,R_1} , M_{u,i,R_2} , ..., M_{u,i,R_K} . Afterwards, we can build a NeuMF model based on each rating matrix above, and these models can produce the predicted rating by given a user, an item and a criterion. The second model is a MLP model for predicting the overall rating. Typically, it is used to capture the non-linear relationships between the overall rating (i.e., R_0) and the multi-criteria ratings (i.e., R_1 , R_2 , ..., R_K). Once all of these models are built and trained, the predicted multi-criteria ratings can be feed into the MLP model to produce the estimated overall rating which can be further utilized to rank items and produce the top- N item recommendations. Apparently, one of the

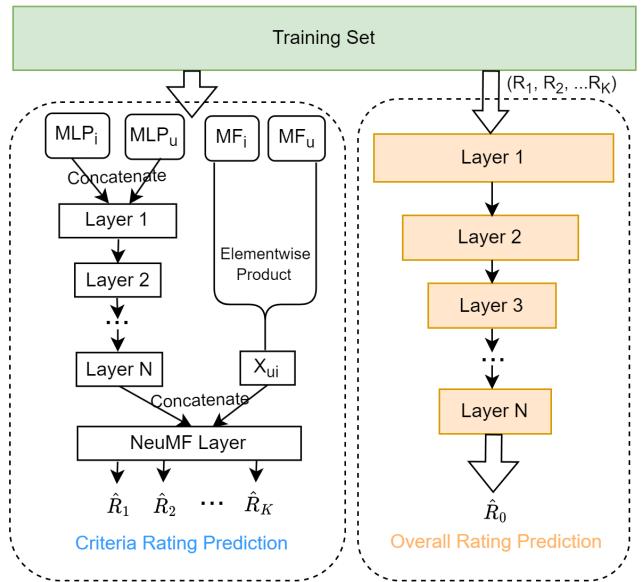


FIGURE 4. The MONeuMF model.

weaknesses in the IndNeuMF model is that we ignore the dependency among different criteria, which may result in weak predictions on these multi-criteria ratings.

In addition, we introduce the approach proposed by Nassar *et al.* [17], as depicted by Figure 4. The part for overall rating predictions is the same with IndNeuMF. In terms of the criteria rating predictions, they utilized a single NeuMF model which has multiple outputs, where each output refers to a predicted rating for a criterion. We name this approach as multi-output NeuMF (MONeuMF) model. By this way, the whole model shares the same embeddings associated with users and items, but produces multiple outputs. The loss function for the model can be considered as a joint MSE loss that can be described by Equation 3, where we use w_t to denote the associated weight with the MSE loss for R_t ($t = 1, 2, \dots, K$).

$$Loss = w_1 \cdot MSE_{R_1} + w_2 \cdot MSE_{R_2} + \dots + w_K \cdot MSE_{R_K} \quad (3)$$

In comparison with IndNeuMF, MONeuMF can be optimized by using the joint loss function above, where the dependency among user preferences in multiple criteria is considered in the model training.

IV. PROPOSED FRAMEWORK: MULTI-CRITERIA RANKING

In this section, we introduce and discuss our proposed multi-criteria recommendation framework.

A. PARETO RANKING

Multi-objective optimization (MOO) [34], [35] techniques have been reused to help produce multi-objective recommendations [36]. Our proposed framework was inspired by the

notion of *Pareto ranking* [31] in the area of multi-objective evolutionary algorithms [33], [35], [44].

MOO is a dedicated discipline of scientific research, especially in the area of decision making due to the fact that a set of objectives are usually involved. The general multi-objective problem (MOP) can be described as follows [37], by assuming that we would like to minimize M objectives.

$$\min_x (f_1(x), f_2(x), \dots, f_M(x)) \quad (4)$$

$$\text{subject to } g_k(x) \leq 0, \quad k = 1, 2, \dots, K; \quad x \in X \subset R^n \quad (5)$$

M is the number of objective functions, and $x = (x_1, x_2, \dots, x_n)$ is a n -dimension decision vector in space R^n . X defines the lower bound and upper bound of x : $X = \{x \mid a_i \leq x_i \leq b_i, i = 1, 2, \dots, n\}$. We define $F(x)$ as $(f_1(x), f_2(x), \dots, f_M(x))$ which is a vector function in the objective space R^M . $g_k(x)$ is used to indicate possible constraints in MOP. A *feasible solution* is defined as any solutions that can satisfy Equation 5.

In order to determine which solution is better by given two feasible solutions x and x^* , the notion of *dominance* or *Pareto dominance* [38] is introduced to MOP: x is dominated by x^* if and only if

$$\begin{aligned} f_m(x^*) &\leq f_m(x) \text{ for all } m = 1, 2, \dots, M, \\ \exists m' : f_{m'}(x^*) &< f_{m'}(x) \end{aligned} \quad (6)$$

Accordingly, the feasible solution x^* is usually named as a *non-dominated solution* or *Pareto optimal solution*, if there are no other feasible solutions that dominates x^* [38]. Therefore, the goal in MOO is seeking a set of Pareto optimal solutions which is also referred to *Pareto set*.

The optimization strategies in MOO can be classified into *Scalarization methods* which can transform a MOP to a single-objective problem, and *population-based heuristic methods* which are also known as the multi-objective evolutionary algorithms (MOEAs) [31], [35], [38].

In most of the MOEAs (especially the ones based on genetic algorithms), the algorithm is required to have a selection process, *elitism* [33], [35], [44], in which a subset of the solutions in the current iteration should be selected to perform the next learning iterations. For example, both parent and children solutions should be ranked in order to select the best solutions to form the new generation of population in multi-objective genetic algorithms [33]. Researchers built different ranking methods or fitness functions for the purpose of ranking in elitism.

Pareto ranking is one of these ranking methods where it relies on the Pareto dominance relationships only. Pareto ranking can rank the individual solutions in a manner such that the non-dominated solutions can have a higher probability of being selected [31]. Researchers have developed different approaches for Pareto ranking, and three of them are usually considered as classical approaches [31]:

- Belegundu's ranking [39]: all non-dominated individuals are assigned rank 0, and the dominated ones rank 1.

- Goldberg's ranking [40]: The method assigns rank 1 to the initial set of non-dominated solutions, then removes them from the solution set to seek the next set of non-dominated solutions which will get ranked 2, and so forth.
- Fonseca and Fleming's ranking [41]: A solution's rank equals to the number of individuals in the current population by which it is dominated.

It is worth noting that there are other ranking methods for elitism, such as ranking dominance [32] and different fitness assignment methods [33]. Pareto ranking purely relies on the Pareto dominance relationships, but other ranking methods may consider other aspects or factors.

B. MCRS BASED ON MULTI-CRITERIA RANKING

Accordingly, we propose our multi-criteria recommendation framework – a series of recommendation models which rely on multi-criteria rankings. The notion of *multi-criteria ranking* refers to a process of generating ranking scores for items by utilizing the predicted multi-criteria ratings. In other words, it is a ranking scheme which can derive a ranking score for an item candidate by a ranking process based on multi-criteria preferences. Theoretically, any methods utilized for elitism in MOEAs can be directly reused for the purpose of multi-criteria ranking in MCRS, if we consider multiple criteria in MCRS as “objectives”.

In this paper, we made the attempt to use Pareto ranking for the purpose of multi-criteria ranking. More specifically, we directly utilized the Fonseca and Fleming's ranking in our experiments. This is the first time to utilize Pareto ranking to sort and rank items directly in MCRS. The only similar work to ours is the one by Ribeiro *et al.* [43], where they utilized Pareto ranking to sort items based on the predicted scores by a set of different recommendation algorithms, for the purpose of obtaining a balance among accuracy, novelty and diversity in the recommendation list.

TABLE 2. Matrix with predicted criteria ratings.

User	Item	Rating	Food	Service	Value
U ₁	T ₁	?	5	5	5
U ₁	T ₂	?	4	4	4
U ₁	T ₃	?	3	3	3
U ₁	T ₄	?	4	3	3
U ₁	T ₅	?	5	4	3

By adapting Pareto ranking to MCRS, we can directly derive a ranking score for each item candidate. Assume there is one user U₁, and we have predicted criteria ratings for the associated five candidate items, as shown by Table 2, the research problem here is how to figure out the ranking of these candidate items without knowing or predicting the overall rating. Namely, we are not going to utilize the aggregation function to estimate the overall rating, but we are going to derive a ranking score for the items directly based on the predicted criteria ratings.

By an observation from Table 2, it is clear that T_1 is the best item, since the multi-criteria ratings associated with T_1 are higher than the ones on other items with corresponding criteria. For the purpose of simplicity, we discuss the ranking of the first four items by excluding the item T_5 . The ranking by considering all five items will be discussed in later paragraphs.

By using a same approach, we can easily derive the rankings for the remaining items. Namely, the ranking of the first four items could be T_1, T_2, T_4, T_3 . This process of multi-criteria decision making is consistent with the idea of Fonseca and Fleming's ranking in MOO.

In our proposed framework, we simply consider these criteria as the “objectives”, and our dominance relation can be defined as – one item T_m can *dominate* another item T_n , if and only if the predicted ratings on all criteria associated with T_m are no less than the ratings associated with T_n , and there must be at least one rating in a specific criterion on the item T_m is larger than the criterion rating on T_n .

Accordingly, by reusing the notion in the Fonseca and Fleming's ranking [41], we are able to derive a *ranking score*

$$\text{Rank Score}(u, i) = \text{number of items that item } i \text{ can dominate in the candidate list for user } u \quad (7)$$

By this way, the ranking score can be derived and shown by Table 3.

TABLE 3. Matrix with ranking score.

User	Item	Rating	Food	Service	Value	Ranking Score
U1	T1	?	5	5	5	4
U1	T2	?	4	4	4	2
U1	T3	?	3	3	3	0
U1	T4	?	4	3	3	1
U1	T5	?	5	4	3	2

This ranking score can be utilized to sort and rank items, in order to produce the top- N recommendation list. We name this recommendation framework as multi-criteria ranking (MCRank). There are at least two advantages to adopt MCRank:

- On one hand, we do not need to build a predictive model to estimate the overall rating by aggregation from multi-criteria ratings. The calculation of the ranking score as shown in Equation 7 is derived from Pareto dominance, and there are no machine learning or optimization involved.
- On the other hand, the framework is general enough to be integrated with any existing multi-criteria recommendation models. For example, we can utilize the process of criteria rating prediction in IndNeuMF (as shown by Figure 3) and MONeuMF (as shown by Figure 5) to predict multi-criteria ratings. Afterwards, these predicted ratings will be sent to Equation 7 to derive the ranking score.

One of the challenges in the ranking is that it is possible to have several items with the same ranking score. If we consider T_5 in Table 2 in the recommendation list, it is not clear which item is better between T_2 and T_5 , since T_5 is better than T_2 in terms of the criterion “food”, but worse if we consider the criterion “value”. As we can observe that T_2 and T_5 have a same ranking score in Table 3. There are several solutions to provide the *differential strategies* which can add a small value in scale [0, 1] to the ranking score in Equation 7. In this paper, we tried two differential strategies:

- The averaging (Avg) strategy: we calculate the average rating of the predicted multi-criteria ratings, and normalized it to a value in scale [0, 1] to be added in Equation 7.
- The crowding distance (Crowd): it is an estimate of the density of solutions surrounding that solution in MOO, so that the selection process can select the individual solutions with a better rank and a decent diversity. We utilize the crowding distance proposed in NSGA-II [44] which is a popular MOEA. The crowding distance in NSGA-II is calculated based on the cardinality of the solution sets and their distance to the solution boundaries. Again, we normalize it to a value in scale [0, 1] to be added in Equation 7.

C. A HYBRID FRAMEWORK

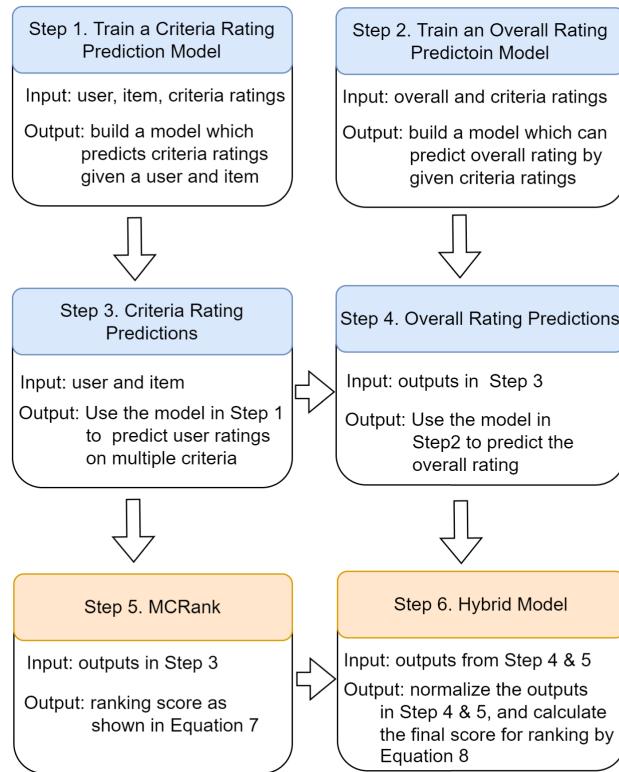
The MCRank model above is a novel recommendation framework in which we can utilize any exiting algorithms in MCRS to predict multi-criteria ratings, and thereby derive the ranking scores based on multi-criteria ranking, in order to produce a list of top- N recommendations. Particularly, we utilize and examine the Pareto ranking by referring to the Fonseca and Fleming's ranking [41] method in this paper.

A more general framework is a hybrid approach of the MCRank model which derive a ranking score and existing MCRS models which predict the overall rating. Therefore, the eventual ranking score can be calculated as follows:

$$\text{Score}(u, i) = w \cdot \text{Rank Score}(u, i) + (1 - w) \cdot \hat{R}_0 \quad (8)$$

In Equation 8, the *Rank Score* refers to the score derived from the MCRank as shown in Equation 7, while \hat{R}_0 refers to the predicted overall rating, such as the predicted rating by using the MLP model which aggregates the predicted multi-criteria ratings, as shown in Figure 3 and 4. Note that, we normalize both the ranking score and the predicted overall rating \hat{R}_0 into a same scale before using them in Equation 8. The value w refers to the weight in scale [0, 1]. The hybrid model becomes the MCRank model when we set w as 1, and it is the regular MCRS model, if we set w as 0.

The workflow of the hybrid framework can be depicted by Figure 5, while the steps in color blue are the ones associated with any existing MCRS models (e.g., IndNeuMF or MONeuMF), and the steps in color orange refer to the proposed approaches in our workflow. Most existing MCRS models can be easily integrated with our hybrid framework. Take the MONeuMF model for example, we train the MONeuMF model, which results in the completion of Step

**FIGURE 5.** The workflow of the hybrid framework.

1 and 2 in Figure 5. Given a user and item, therefore, we are able to utilize the trained model in MONeuMF to predict the multi-criteria ratings (i.e., Step 3) and the overall rating (i.e., Step 4). The ranking score by MCRank (i.e., Step 5) can be produced by following Equation 7. The score in the hybrid model (i.e., Step 6) can be generated by following Equation 8.

V. EXPERIMENTS AND RESULTS

In this section, we first introduce the real-world data sets, and evaluation protocols, followed by our experimental results.

A. DATA SETS

We use four real-world data sets for the experimental purpose. The statistics of the data sets can be described by Table 4.

TABLE 4. Data sets.

	ITM	OpenTable	Yahoo5	Yahoo13
# of users	269	1,309	2,162	6,078
# of items	70	91	3,078	976
# of ratings	3,306	19,537	62,739	62,156
Rating sparsity	82.4%	83.6%	99.1%	99.0%
Rating scale	[1, 5]	[1, 5]	[1, 5]	[1, 13]
Criteria	App Data Ease	Food Service Ambience Value	Story Direction Acting Visual	Story Direction Acting Visual

The *ITM* data was collected for the educational project recommendations [45], while students' preferences on Kaggle

data sets were collected from questionnaires. There are 3,306 rating entries given by 269 users on 70 items. Each rating entry is also associated with three criteria: App (how students like the application or domain of the data), Data (the ease of data preprocessing) and Ease (the overall ease of the project), in addition to the overall rating.

The *OpenTable* data was collected from a Web crawler, where we crawled user reviews on 91 restaurants in Chicago, USA. There are 19,537 rating entries given by 1,309 users, while each entry is associated with four criteria: food, service, ambience and value.

The *Yahoo5* and *Yahoo13* data were obtained from Yahoo!Movies, but we have ratings in difference scales in the two data sets. The *Yahoo5* data is in scale 1 to 5, and it was collected by Jannach *et al.* [46]. There are 62,739 ratings given by 2,162 users on 3,078 movies. Each user gave at least 10 ratings which are associated with multi-criteria ratings on four criteria: story, direction, acting effect and visual effect. By contrast, the *Yahoo13* data was collected by Lakiotaki *et al.* [47]. The data contains the same criteria as the ones in the *Yahoo5* data, but there are different rating entries given by different users and items.

B. EVALUATION PROTOCOLS

We selected IndNeuMF and MONeuMF¹ as baseline approaches and tried to incorporate the multi-criteria ranking based on Pareto ranking into these baselines. More specifically, we compare four models with the baseline models:

- The *MCRank* model refers to the recommendation models which only utilize Pareto ranking to produce the top-*N* recommendations. It is equivalent to the hybrid model, where we set *w* as 1 in Equation 8.
- The *Hybrid* model refers to the best performing hybrid approach by Equation 8. The optimal weight *w* is found by a grid search from [0.1, 0.9] with a step of 0.1 in the process.
- The *Best* model refer to the best performing model between the *MCRank* and *Hybrid* model. We additionally apply the two differential strategies (e.g., Avg or Crowding distance) to the best model to observe whether they can bring additional improvements.

We evaluate and compare all of these models based on top-10 item recommendations through a process of 5-fold cross validation. We utilize the median value of the rating scale (i.e., 7 for the *Yahoo13* data, and 3 for all other data sets) as the cut-off value to distinguish positives and negatives in the data sets.

We use *F*₁ measure to examine the relevance. *F*₁ is a fusion of both precision and recall, as shown by Equation 11 [48], [49]. Precision is the fraction of relevant items among the recommended items, while recall is the fraction of relevant items that were retrieved. The number of matched relevant items is usually referred as the number of true positives (TP), while false positives (FP) denote the number of items

¹We use equal weights in Equation 3 to simplify the parameter tunings.

which are not relevant but shown in the recommendation list, and false negatives (FN) refer to the number of items which are actually relevant but not recommended.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (10)$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

In addition, we adopt normalized discounted cumulative gain (NDCG) [49], [50] to examine the ranking quality. NDCG is a metric used for listwise ranking in the well-known learning-to-rank methods. Assuming each user u has a “gain” g_{uij} from being recommended an item i at rank j , the average Discounted Cumulative Gain (DCG) for a list of J items is defined in Equation 13.

$$DCG = \frac{1}{N} \sum_{u=1}^N \sum_{j=1}^J \frac{g_{uij}}{\max(1, \log_2 j)} \quad (12)$$

i_j refers to the item at the j^{th} position of the recommendation list. g_{uij} indicates the gain from this specific item, and it can be calculated from the utility function [51] below, where Rel_{ij} refers to the relevance score (e.g., real rating or rank) of item i_j in the ground truth.

$$g_{uij} = 2^{\text{Rel}_{ij}} - 1 \quad (13)$$

NDCG, therefore, is the normalized version of DCG given by Equation 14, where DCG^* is the ideal DCG, i.e., the maximum possible DCG computed from the real ranking of items.

$$NDCG = \frac{DCG}{DCG^*} \quad (14)$$

Through the experiments, we desire to answer the following questions:

- By fusing multi-criteria ranking into the baseline only, can it outperform the baseline approach?
- By the hybrid model which combines the multi-criteria ranking model and the traditional overall rating prediction model, can we obtain more improvements?
- What is the impact by the differential strategies? any further improvements?

C. EXPERIMENTAL RESULTS

As indicated above, we utilize IndNeuMF and MONeuMF as baselines, and incorporate multi-criteria ranking to build the MCRank and hybrid models. Figure 6 presents our results by integrating multi-criteria ranking with IndNeuMF, while Figure 7 describes the experimental results by integrating multi-criteria ranking with MONeuMF.

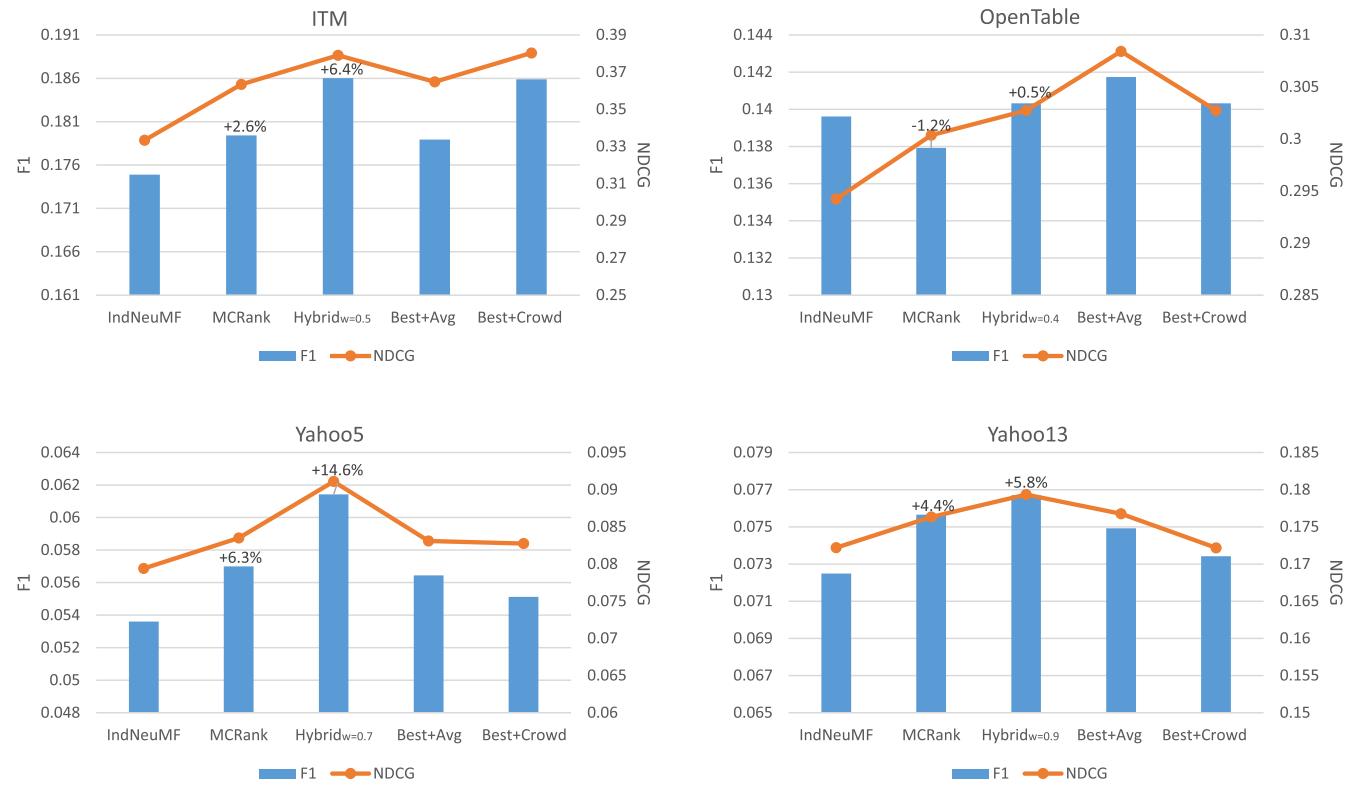
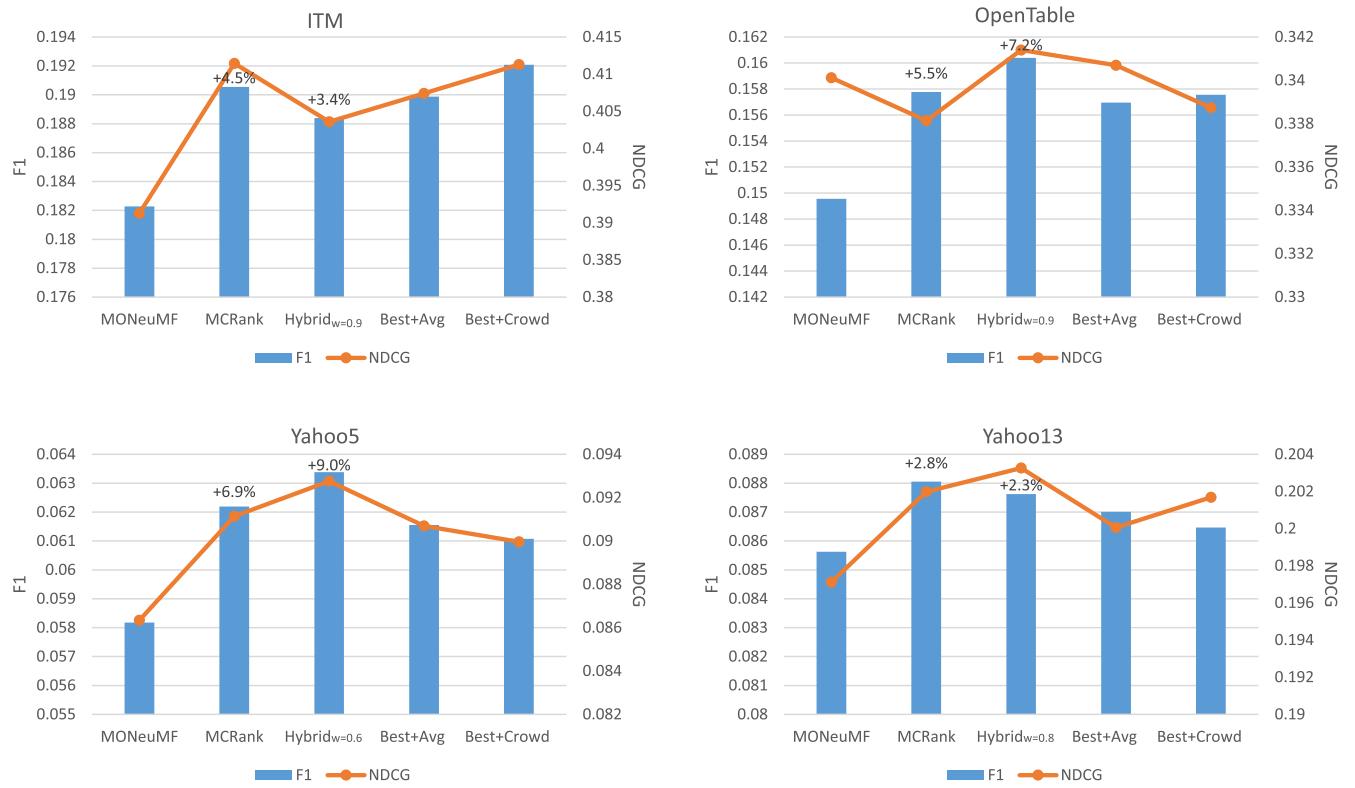
In Figure 6 and 7, we use bars to represent the results in F_1 with respect to the y-axis on the left, while the curve is used to depict the results in NDCG with respect to the y-axis on the right. In terms of the hybrid model, we indicate the optimal weight w as subscript in these figures. Once we identify

the besting performing model (i.e., the best model between MCRank and Hybrid), we additionally apply the differential strategies and use “Best + Avg” and “Best + Crowd” to denote these models. Moreover, we add the improvement percentage in the F_1 metric for the MCRank and Hybrid model, in comparison with the baseline approach (i.e., IndNeuMF in Figure 6 and MONeuMF in Figure 7).

First of all, we focus on the comparison between the baseline model and MCRank which refers to the top- N recommendation model by the multi-criteria ranking shown in Equation 7. In Figure 6, MCRank is able to outperform IndNeuMF in terms of both F_1 and NDCG in all of the four data sets, except the F_1 metric in the OpenTable data set. The improvement is significant, particularly in the Yahoo5 and Yahoo13 data sets. In Figure 7, MCRank is able to outperform MONeuMF in terms of both F_1 and NDCG in all of the four data sets, except NDCG in the OpenTable data set. The improvement by MCRrank is generally above 4%, except the one in Yahoo 13 data. It is also worth noting that the MONeuMF performs much better than the IndNeuMF model in our experiments. It is probably because IndNeuMF ignores the criteria dependency in the model. All of the observed results can demonstrate that we are able to improve the baseline by fusing multi-criteria ranking. Recall that there are no optimization or machine learning process in calculating the ranking score in Equation 7. The experimental results confirm the effectiveness of ranking items by using the Pareto ranking technology.

Moreover, we are interested in whether they hybrid model can provide more improvements. According to the observations in Figure 6 and 7, the hybrid model can generally outperform the MCRank model, except when we utilize the hybrid approach together with MONeuMF on the ITM and Yahoo13 data sets. More specifically, both F_1 and NDCG declined in the ITM data set, and F_1 was decreased a little bit in the Yahoo13 data, as shown in Figure 7. It tells that it is better to utilize the pure multi-criteria ranking, rather than a hybrid estimation of the ranking score with the predicted overall rating. It is also interesting to take a look at the optimal weights in the hybrid model. In the Yahoo5 and Yahoo13 data sets, the optimal weight is general higher than 0.5, which indicates the importance of the multi-criteria rankings. In terms of the ITM and OpenTable data sets, the optimal weight depends on the specific baseline we use. The weight is 0.9 if we fuse multi-criteria ranking into MONeuMF model, but the weight is much lower if we integrate the ranking with IndNeuMF model. It is not surprising, since the quality of the Pareto ranking depends on the quality of predicted criteria ratings – that’s why the performance and the optimal weight in the hybrid model may differ, when we integrating the multi-criteria ranking with different MCRS models.

Finally, we validate whether the differential strategies are helpful to improve the best performing model. Unfortunately, we only observe the improvement in the OpenTable data set, when we used the averaging strategy together with the hybrid model, as shown in Figure 6. Using the crowding distance

**FIGURE 6.** Fusing multi-criteria ranking (MCRank) into IndNeuMF.**FIGURE 7.** Fusing multi-criteria ranking (MCRank) into MONeuMF.

together with the MCRank model on the ITM data set can bring a small improvement on the F_1 metric, as shown by

Figure 7. Even so, we still believe that the differential strategy is necessary, especially when there are many criteria.

increased. It is because that it is more possible to have items with the same Pareto rank, when there are several criteria or objectives in the data set. We will continue to seek other differential strategies which may be helpful to improve the recommendation performance in our future work.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose multi-criteria ranking which is a general multi-criteria recommendation framework, where most existing multi-criteria recommendation algorithms can be integrated with it. In multi-criteria ranking, we can estimate a ranking score directly from the predicted multi-criteria ratings. We claim that the ranking methods for the purpose of elitism in MOEAs can be reused for multi-criteria ranking, though multi-criteria ranking is not limited to these methods only. In our experiments, we only examined an example of Pareto ranking (i.e., Fonseca and Fleming's ranking) for the purpose of multi-criteria ranking in our MCRS model. The experimental results demonstrate the effectiveness of using multi-criteria ranking:

- By integrating existing multi-criteria recommendation models (i.e., IndNeuMF and MOneuMF) with Pareto ranking, the model (i.e., MCRank) can outperform the baseline approaches over all data sets, except the OpenTable data when we use IndNeuMF as baseline. In the OpenTable data, we can observe that MCRank can outperform the baseline approach if we use MOneuMF as the baseline, but it fails to outperform IndNeuMF when IndNeuMF was integrated with Pareto ranking. The performance by IndNeuMF may be the key factor that leads to the decreased F_1 result. Note that MOneuMF works much better than IndNeuMF on the OpenTable data.
- By a linear hybrid of MCRank and the baseline approach, we are able to obtain further improvements, except the results by the hybrid model on Yahoo13 and ITM data when we use MOneuMF as baseline. In these two cases, our MCRank model without a hybrid aggregation works the best. It is also worth noting that most of the optimal weights are no less than 0.5 (except the Hybrid model on OpenTable data). It further reveals the effectiveness of MCRank, since this weight w represents the importance of MCRank in the hybrid model.

In addition to the effectiveness, the proposed multi-criteria ranking framework is general enough to be integrated with any existing multi-criteria recommendation models. It opens the door to novel research direction in MCRS. In our future work, we will try to examine other existing ranking technologies in elitism, and even build our own multi-criteria ranking methods. In addition, the differential strategy is one of the interesting directions which may bring further improvement in the multi-criteria recommendations.

ACKNOWLEDGMENT

The authors would like to thank the support and grant by Google Cloud Platform and Amazon Web Services which

provide us the cloud computing resources (e.g., GPU) to assist our experimental evaluations.

REFERENCES

- [1] B. Smith and G. Linden, "Two decades of recommender systems at Amazon.com," *IEEE Internet Comput.*, vol. 21, no. 3, pp. 12–18, May 2017.
- [2] B. M. Shoja and N. Tabrizi, "Customer reviews analysis with deep neural networks for e-commerce recommender systems," *IEEE Access*, vol. 7, pp. 119121–119130, 2019.
- [3] C. A. Gomez-Uribe and N. Hunt, "The Netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, pp. 1–19, Jan. 2016.
- [4] Z. Su, Z. Lin, J. Ai, and H. Li, "Rating prediction in recommender systems based on user behavior probability and complex network modeling," *IEEE Access*, vol. 9, pp. 30739–30749, 2021.
- [5] J. Tang, X. Hu, and H. Liu, "Social recommendation: A review," *Social Netw. Anal. Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.
- [6] A. Belhadi, Y. Djennouri, J. C. Lin, and A. Cano, "A data-driven approach for Twitter hashtag recommendation," *IEEE Access*, vol. 8, pp. 79182–79191, 2020.
- [7] G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," *IEEE Intell. Syst.*, vol. 22, no. 3, pp. 48–55, May/Jun. 2007.
- [8] S. M. Al-Ghuribi and S. A. M. Noah, "Multi-criteria review-based recommender system—The state of the art," *IEEE Access*, vol. 7, pp. 169446–169468, 2019.
- [9] N. Manouselis and C. Costopoulou, "Experimental analysis of design choices in multiattribute utility collaborative filtering," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 21, no. 2, pp. 311–331, 2007.
- [10] N. Sahoo, R. Krishnan, G. Duncan, and J. Callan, "Research note—The halo effect in multicomponent ratings and its implications for recommender systems: The case of Yahoo! Movies," *Inf. Syst. Res.*, vol. 23, no. 1, pp. 231–246, Mar. 2012.
- [11] Y. Zheng, "Criteria chains: A novel multi-criteria recommendation approach," in *Proc. 22nd Int. Conf. Intell. User Interfaces*, Mar. 2017, pp. 29–33.
- [12] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
- [13] M. Hong and J. J. Jung, "Multi-criteria tensor model for tourism recommender systems," *Expert Syst. Appl.*, vol. 170, May 2021, Art. no. 114537.
- [14] M. Hong and J. J. Jung, "Sentiment aware tensor model for multi-criteria recommendation," *Int. J. Speech Technol.*, vol. 2022, pp. 1–20, Mar. 2022.
- [15] M. Hassan and M. Hamada, "A neural networks approach for improving the accuracy of multi-criteria recommender systems," *Appl. Sci.*, vol. 7, no. 9, p. 868, Aug. 2017.
- [16] Z. Batmaz and C. Kaleli, "AE-MCCF: An autoencoder-based multi-criteria recommendation algorithm," *Arabian J. Sci. Eng.*, vol. 44, no. 11, pp. 9235–9247, Nov. 2019.
- [17] N. Nassar, A. Jafar, and Y. Rahhal, "Multi-criteria collaborative filtering recommender by fusing deep neural network and matrix factorization," *J. Big Data*, vol. 7, no. 1, pp. 1–12, Dec. 2020.
- [18] Q.-H. Le, T. N. Mau, R. Tansuchat, and V.-N. Huynh, "A multi-criteria collaborative filtering approach using deep learning and Dempster-Shafer theory for hotel recommendations," *IEEE Access*, vol. 10, pp. 37281–37293, 2022.
- [19] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Nov. 2003.
- [20] I. B. Sassi, S. B. Yahia, and I. Liiv, "MOREc: At the crossroads of context-aware and multi-criteria decision making for online music recommendation," *Expert Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115375.
- [21] Y. Zheng, S. Shekhar, A. A. Jose, and S. K. Rai, "Integrating context-awareness and multi-criteria decision making in educational learning," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 2453–2460.
- [22] R. Dridi, L. Tamine, and Y. Slimani, "Exploiting context-awareness and multi-criteria decision making to improve items recommendation using a tripartite graph-based model," *Inf. Process. Manage.*, vol. 59, no. 2, Mar. 2022, Art. no. 102861.
- [23] 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, Aug. 2021, Art. no. 114868.

- [24] S. Aysha and S. Tarun, "A Pareto dominance approach to multi-criteria recommender system using PSO algorithm," in *Proc. Int. Conf. Innov. Comput. Commun.* Singapore: Springer, 2022, pp. 737–755.
- [25] Y. Zheng, "Utility-based multi-criteria recommender systems," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 2529–2531.
- [26] M. M. Hafez, R. P. D. Redondo, A. F. Vilas, and H. O. Pazó, "Multi-criteria recommendation systems to foster online grocery," *Sensors*, vol. 21, no. 11, p. 3747, May 2021.
- [27] E. Hikmawati, N. U. Maulidevi, and K. Surendro, "Multi-criteria recommender system model for lockdown decision of COVID-19," in *Proc. 10th Int. Conf. Softw. Comput. Appl.*, Feb. 2021, pp. 39–44.
- [28] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, Feb. 2019.
- [29] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [31] I. Alberto, C. Azcarate, F. Mallor, and P. Mateo, "Multiobjective evolutionary algorithms. Pareto rankings," *Monografias Seminario Matematico Garcia Galdeano*, vol. 27, pp. 25–35, Jan. 2003.
- [32] S. Kukkonen and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 3983–3990.
- [33] M. Garza-Fabre, G. T. Pulido, and C. A. C. Coello, "Ranking methods for many-objective optimization," in *Proc. Mex. Int. Conf. Artif. Intell.* Berlin, Germany: Springer, 2009, pp. 633–645.
- [34] K. Deb, "Multi-objective optimization," in *Search Methodologies*. Boston, MA, USA: Springer, 2014, pp. 403–449.
- [35] C. A. C. Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.
- [36] Y. Zheng and D. Wang, "A survey of recommender systems with multi-objective optimization," *Neurocomputing*, vol. 474, pp. 141–153, Feb. 2022.
- [37] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidisciplinary Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004.
- [38] C. Lücken, B. Barán, and C. Brizuela, "A survey on multi-objective evolutionary algorithms for many-objective problems," *Comput. Optim. Appl.*, vol. 58, no. 3, pp. 707–756, Jul. 2014.
- [39] A. Belegundu, E. Constans, R. Salagame, and D. Murthy, "Multi-objective optimization of laminated ceramic composites using genetic algorithms," in *Proc. 5th Symp. Multidisciplinary Anal. Optim.*, Sep. 1994, p. 4363.
- [40] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
- [41] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation discussion and generalization," in *Proc. ICGA*, vol. 93, Jul. 1993, pp. 416–423.
- [42] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1995.
- [43] M. T. Ribeiro, N. Ziviani, E. S. D. Moura, I. Hata, A. Lacerda, and A. Veloso, "Multiobjective Pareto-efficient approaches for recommender systems," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, pp. 1–20, Jan. 2015.
- [44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2000.
- [45] Y. Zheng, "Personality-aware decision making in educational learning," in *Proc. 23rd Int. Conf. Intell. User Interface Companion*, Mar. 2018, p. 58.
- [46] D. Jannach, M. Zanker, and M. Fuchs, "Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations," *Inf. Technol. Tourism*, vol. 14, no. 2, pp. 119–149, 2014.
- [47] K. Lakiotaki, N. F. Matsatsinis, and A. Tsoukias, "Multicriteria user modeling in recommender systems," *IEEE Intell. Syst.*, vol. 26, no. 2, pp. 64–76, Mar./Apr. 2011.
- [48] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.
- [49] A. Gunawardana, G. Shani, and S. Yosef, "Evaluating recommender systems," in *Recommender Systems Handbook*. New York, NY, USA: Springer, 2022, pp. 547–601.
- [50] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, Oct. 2002.
- [51] H. Steck, "Evaluation of recommendations: Rating-prediction and ranking," in *Proc. 7th ACM Conf. Recommender Syst.*, Oct. 2013, pp. 213–220.



YONG ZHENG (Member, IEEE) received the Ph.D. degree in computer and information sciences from DePaul University, Chicago, IL, USA, in 2016.

He is currently an Assistant Professor at the Department of Information Technology and Management, College of Computing, Illinois Institute of Technology, Chicago. His research interests include user modeling, personalization, and recommender systems. He is active in professional services for the communities of user modeling, recommender systems, and STEM educations. He was invited to serve on the organization committee in several relevant conferences, such as ACM RecSys, ACM UMAP, ACM IUI, ACM Hypertext, and ACM SIGITE. He served as a PC Member and a Reviewer for several well-known academic conferences and journals, such as IJCAI, KDD, SIGIR, WWW, WSDM, RecSys, UMAP, ACM TOIS, ACM TIST, ACM TiiS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and UMUAI.



DAVID (XUEJUN) WANG received the Ph.D. degree in mathematics from the University of Illinois at Chicago, Chicago, IL, USA. He is currently working as a Principal Data Scientist at Morningstar Inc., an investment research and management firm headquartered in Chicago. His research interests include AI and machine learning, and its applications in financial services.