Utility-Based Multi-Stakeholder Recommendations By Multi-Objective Optimization

Yong Zheng
Department of Information Tech & Management
School of Applied Technology
Illinois Institute of Technology
Chicago, Illinois, USA
yong.zheng@iit.edu

Aviana Pu
Department of Information Tech & Management
School of Applied Technology
Illinois Institute of Technology
Chicago, Illinois, USA
apu78@hawk.iit.edu

***Abstract*—In the recommender systems, the receiver of the recommendations may not be the only stakeholder in the system, while others may come into play. For example, job positions cannot be simply recommended to a user according to his or her tastes only without considering the expectations of the recruiters. In this paper, we propose a utility-based recommendation model which produces recommendations by optimizing the utilities of multiple stakeholders. Particularly, we take advantage of the multi-criteria ratings that are associated with user expectations and evaluations. And we propose to learn the user expectations by the learning-to-rank approaches if they are unknown in the data. We also propose to seek the optimal solutions by using the multi-objective optimization techniques. Our experiments based on a speed-dating data set demonstrate the effectiveness of the proposed methods in which we are able to keep the balance between multiple utilities and the recommendation performance by adopting the multi-objective optimization.**

***Index Terms*—recommender system, utility, multi-stakeholder, multi-criteria, multi-objective, optimization**

I. Introduction

Traditional recommender systems produce a list of items recommendations to the users tailed by the user preferences. However, the receiver of the recommendations may not be the only player in the system, and other stakeholders may come into play. There are several real-world examples:

- In the *dating* application [21], a young man may prefer a recommended woman who also wants to date with him, rather than only the pool of ladies that he likes.
- The similar thing may happen in the *job seeking* or *recruitment* case. We may need to recommend some job positions to a user while the corresponding employees may be willing to hire him or her.
- In the advertising area [23], not only the user preferences (e.g., interests on Ads, privacy concerns) but also the interests of the advertisers should be considered. For example, the advertiser would like to present the car advertisement to the appropriate customers, rather than any groups of the viewers who like cars. Teenagers may like cars but they may not have the capability to make purchases, which decreases the utility of the advertisers.
- A list of recommended projects for students may be produced by considering both the student preferences and expectations or requirements form the instructors [43].
- The recommendations for a team or a group of people may need to consider the perspective of each team member, such as group learning [41], [42].

In contrast to the traditional recommendations which only utilize the preferences of the end users, multi-stakeholder recommender systems [3], [40] aim to produce fair recommendations from the perspective of multiple stakeholders. Our initial work [44] use reciprocal recommendations [20], [21], [33] as a case study, build multi-stakeholder recommender systems by taking advantage of the known user expectations and evaluations. However, there are two major weaknesses – we may not know user expectations in the real-practice, and the optimal solutions cannot be always found by an exhaustive search due to the complexity and the scale of the data. In this paper, we propose several methods to address these issues so that the utility-based multi-stakeholder recommender systems can be easily extended to other domains or applications. Our contributions in this paper can be summarized as follows.

- We build utility-based multi-stakeholder recommendation algorithms by utilizing the multi-criteria ratings that are associated with user expectations and evaluations.
- We propose different methods to learn user expectations if they are unknown in the data.
- We propose to find the optimal solutions by using multi-objective optimization which could be more effective and efficient than the exhaustive search.
- Our experimental results based on a speed-dating data set demonstrate the effectiveness of our proposed methods.

II. Related Work

A. Multi-Stakeholder Recommender Systems

The topic of multi-stakeholder recommendations was inspired by the notion of "fairness". Algorithmic fairness [5], [10], [36] could be defined as a type of constrained optimization which is designed to improve both the efficiency and equity of decisions. It is necessary because our machine learning algorithms with the standard utility-maximization objectives (e.g., maximizing prediction accuracy or minimizing prediction errors) sometimes resulted in algorithms that behaved in a way in which a human observer will deem unfair, often

especially towards a certain minority, such as the cases with respect to user gender, race, disabilities, and so forth.

Burke [3] extended this concept to the area of recommender systems. Basically, there could be two ways to incorporate fairness into recommender systems. One is a similar way as the original definition of algorithmic fairness, where we try to balance the equity in the algorithm components with respect to minority groups which may be defined by specific attributes, such as user gender or race. For example, by taking gender and race into consideration, Burke, et al. [4] proposed a way to balance the neighborhood selection in the recommendation algorithms. Another idea is recommending items by considering the role of multiple stakeholders in the system, so that the recommendations are fair from the perspective of these stakeholders. Burke [3] further define and clarify the type of stakeholders in the area of movie watching. For example, the potential stakeholders involved in a movie domain could be the receiver of movie recommendations, the producer of the movies, the cinema which plays the movies, and even the movie director. Our initial work [44] proposed a utility-based model which relies on the information of user expectations and an exhaustive search to look for the optimal solutions in the dating applications.

B. Reciprocal Recommendation

Reciprocal recommender systems have been applied in multiple domains, such as dating [20], recruitment [31], [34] and social connections [14]. It can be viewed as a special case of multi-stakeholder environment, in which the "two-sided" or the "bilateral" interactions between the peers are considered. Take online dating [20], [33] for an example, the recommended females to a male user cannot simply be the suggestions who match the male user's preferences. A successful dating recommendation may have to also consider the options from the suggested partners. In the case of recruitment, we cannot only recommend the job positions that a user is interested in without considering the expectations from the recruiters.

However, the corresponding utilities in these studies are usually derived from simulations or implicit information. For example, the RECON [20] approach considers the match in user demographic information and the communications between the peers. Xia, et al. [33] additionally take the dating interactions into considerations. These information may indicate how a user likes others, but it may not be a good representation as the utility of the users. Our previous work [44] utilized a data, in which the explicit information of the user expectations are presented. But we may not know this information in the real-practice, which leaves the limitations in our previous work.

C. Multi-Objective Learning

The task of the multi-stakeholder recommender systems is highly correlated with the multi-objective optimization. For example, the increment of the utility from the perspective of one stakeholder may hurt the utility of the other stakeholders. Our previous work [44] used an exhaustive search to find the optimal solution, but human efforts are required to make the final decision and the process may not be efficient enough when the data is in large-scale. Multi-objective optimization, therefore, is one of the promising solutions to find the optimal solutions efficiently. It has been applied to the area of recommender systems [13], [26], [46] to achieve the balance among precision, novelty and diversity of the item recommendations.

III. Data and Problem Statement

A. The Speed-Dating Data

The speed-dating data was collected and introduced by Fisman, et al. [11]. It is also available on Kaggle.com[1]. The data was gathered from participants in the experimental speed dating events from 2002 to 2004. The subjects would have a four minute "first date" with every other participant of the opposite gender. At the end of their four minutes, participants were asked if they would like to see their date again. They were also asked to rate their date on six attributes: attractiveness, sincerity, intelligence, fun, ambition, and shared interests.

The data contains rich information. Table I shows an example of the data excluding demographic information. We list the useful information that we may need in our work as follows:

- Demographic Information: Each subject was requested to fill in a questionnaire in which they need to provide a list of demographic information, such as age, gender, country, field of studies, degree, race, income level, and so forth.
- Dating Goals and Expectations: In the questionnaire, each subject defined their dating goals and expectations in advance. This information was stored in either textual options (e.g., the primary goal of looking for a partner, etc.) or their expectations by numerical ratings in six criteria: attractiveness (i.e., Exp-1), sincerity (i.e., Exp-2), intelligence (i.e., Exp-3), fun (i.e., Exp-4), ambition (i.e., Exp-5) and shared interests (i.e., Exp-6). The summation of the ratings should be 100 in total. We utilize these multi-criteria ratings only in our work.
- Dating Experience: Each subject will represent their dating experience in multiple ways. First of all, each subject will leave the ratings on the same six criteria mentioned above to indicate how the partner meets their expectations (i.e., the columns Eval-1 to Eval-6 in Table I). In addition, each subject also gave an overall rating in scale 1 to 10 (i.e., the column "Like" in Table I) to the partner. Finally, each subject needs to present the willing which tells whether they want to have a further date with the same partner in the future or not. The binary variable, "match", indicates whether both of them would like to keep the relationship or continue the dating in the future.

There are 392 subjects in total. The number of dates by each subject is at least 9 and at most 21. The whole data contains 8,378 records which describe the information for each date. Note that a date between user a and user b will be described by two rows or records in the data. Only 16.5% of the records present matched dating experience.

[1]https://www.kaggle.com/annavictoria/speed-dating-experiment

TABLE I
EXAMPLE OF THE DATA

User	Partner	Like	Match	Eval-1	Eval-2	Eval-3	Eval-4	Eval-5	Eval-6	Exp-1	Exp-2	Exp3	Exp-4	Exp-5	Exp-6
2	16	7	0	8	7	8	3	6	2	45	5	25	20	0	5
2	11	7	0	5	7	8	4	6	3	45	5	25	20	0	5
2	13	10	1	5	8	9	6	3	4	45	5	25	20	0	5
6	16	6	0	3	7	6	1	1	1	10	25	20	25	5	15
6	13	10	1	4	7	9	4	3	2	10	25	20	25	5	15

B. Problem Statement

Take suggesting user b to user a for an example, we define user a as the *target user*, while b will be the recommended *partner*. Our major task is to recommend a list of appropriate partner users to a target user. Particularly, the appropriateness is defined by the *matched* dating experience which can be indicated by the binary variable "match". Our previous work [44] has shown case of a utility-based model which we will discuss for more details in the next section. In this paper, we specifically focus on the following research problems:

- The proposed utility model may need the information of user expectations. We would like to figure out how to learn these information if they are not observed in the data or the applications.
- In addition, we attempt to use multi-objective learning to seek the optimal solutions instead of the exhaustive search in our previous work [44].
- We will compare the optimal solutions by using the original and the learned user expectations to examine whether we can obtain equivalent and even better solutions.

IV. METHODOLOGY

In this section, we first introduce the workflow of our methods, and discuss two major challenges in our work – the learning of user expectations, and the multi-objective optimization.

A. Workflow

The initial idea of utility-based multi-stakeholder recommender systems was proposed in our previous work [44]. We formally summarize it as follows.

Recall that we have multi-criteria ratings in the data set which are potentially helpful in multi-criteria recommender systems [1], [39]. In our work, we take advantage of them in a different way. More specifically, we define the utility of the stakeholder as the similarity between the vector of user expectations and the vector of user evaluations in terms of the multi-criteria ratings. More specifically, we use $\overrightarrow{c_u}$ to represent the vector of user expectations, and $\overrightarrow{r_{u,i}}$ denotes the u's rating vector on the item i, as shown below. Note that these two vectors denote user's expectations and evaluations on the same six criteria in our speed-dating data – attractiveness, sincerity, intelligence, fun, Ambition and shared interests.

$$\overrightarrow{c_u} =< c_u^1, c_u^2, c_u^3, c_u^4, c_u^5, c_u^6 > \quad (1)$$

$$\overrightarrow{r_{u,i}} =< r_{u,i}^1, r_{u,i}^2, r_{u,i}^3, r_{u,i}^4, r_{u,i}^5, r_{u,i}^6 > \quad (2)$$

The value of the utility can be obtained by the corresponding similarity measures between two vectors, such as Pearson correlation, cosine similarity, Jaccard similarity, and so forth. In our speed-dating data, the information of user expectations are observed. We propose to learn user expectations in Section IV-B, if these information is unknown in other domains or applications. Given an item i, $\overrightarrow{r_{u,i}}$ represents the user evaluations on the item i in shape of the multi-criteria ratings. For the recommendation purpose, we need to predict $\overrightarrow{r_{u,i}}$ for the user u on all of the candidate items. Simply, we apply the biased matrix factorization [17] to the corresponding user-item-criterion rating matrix for the prediction purpose.

Assume we are going to recommend user b to date with user a, the corresponding utilities can be defined as follows:

$$Utility(a) = similarity(\overrightarrow{c_a}, \overrightarrow{r_{a,b}}) \quad (3)$$

$$Utility(b) = similarity(\overrightarrow{c_b}, \overrightarrow{r_{b,a}}) \quad (4)$$

Note that $\overrightarrow{r_{a,b}}$ and $\overrightarrow{r_{b,a}}$ represent how a evaluates b and how user b likes a in terms of the multi-criteria ratings respectively. We tried cosine similarity, Jaccard similarity and the Pearson correlation to calculate the similarity values, and our experiments reveal that Jaccard similarity did not work well, and Pearson correlation slightly worked better than the cosine similarity. We introduce and discuss our experimental results by using Pearson correlation in the following sections.

In the utility-based recommender systems, we will use the utility to rank the items for the recommendation purpose. Therefore, a ranking score based on the utilities can be described by Equation 5, if we are going to recommend user b to the user a in the reciprocal recommendations.

$$Score(a \leftarrow b) = \beta \times Utility(a) + (1 - \beta) \times Utility(b) \quad (5)$$

It is a linear combination of the utilities from the perspective of user a and b, so that the fairness between stakeholders can be achieved. β is a factor to indicate the weights of the utilities. The value of β ($0 \leq \beta \leq 1$) should be carefully defined for the following reasons. On one hand, the overall utility (i.e., the ranking score in Equation 5) will be used to produce the recommendations, where the β should balance the utilities and the recommendation performance. On the other hand, β can be adjusted due to the characteristics of the applications. For example, in the dating applications, a VIP user (e.g., a user with paid membership) may have the priority to receive the recommendations with more utilities from their perspectives

than the users without membership. We simply ignore the issue of membership in our work, and treat all users equally.

Finally, we need to learn the optimal β value, in order to achieve the balance among different objectives, such as individual utilities and the overall utility, as well as the recommendation performance, such as precision in the top-N recommendations. The specific learning methods will be discussed in Section IV-C.

B. Learning User Expectations

Our previous work [44] shows case of how useful the user expectations are in the multi-stakeholder recommender systems. In our speed-dating data, this information is observed in shape of the multi-criteria ratings. Unfortunately, the information of user expectations may not be explicitly available in other data or domains. There are usually two ways to alleviate this problem: one is to generate user expectations by aggregating the implicit preferences. However, domain knowledge may be required to correlate the implicit information (such as user behaviors, textual comments, etc) with the corresponding criteria. Another method is to learn the user expectations by minimizing the prediction errors or maximizing the user preferences. We discuss this learning approach in this section.

Let l_i represent the ranking label for the item i. If $l(i) > l(j)$, then the item i should be ranked before the item j. Let f denotes the ranking function, while $f(u,i)$ will produce a ranking score for the item i from the perspective of a user u.

To simplify the learning process, the ranking function in our work, $f(u,i)$, is represented by the dot product of these two vectors, as shown in Equation 6. The dot product function is usually viewed as a simplified representation of cosine similarity to be used in the learning and optimization process.

$$f(u,i) = \overrightarrow{c_u} \cdot \overrightarrow{r_{u,i}} \tag{6}$$

We can learn user expectations in two ways: one is that we learn the user expectations first, and then utilize these learned expectations to perform the following steps which are related to multi-objective optimizations. Another way is that we can consider these user expectations as the variables to be learned in the process of multi-objective optimizations. In this section, we only discuss our approach of the first way as follows.

There are two possible optimization goals, if we decide to learn the user expectations first. On one hand, we can learn the $\overrightarrow{c_u}$ by minimizing the sum of squared prediction errors, with respect to the user's overall rating on the item (e.g., the variable "Like" in Table I). On the other hand, we can learn the $\overrightarrow{c_u}$ by maximizing the rankings. We tried both, and found that the learning-to-rank method is significantly better. We specifically introduce our learning-to-rank methods as follows. Note that the ground truth of the rank can be obtained by the ratings in the variable "Like" as shown in Table I.

1) Pointwise Ranking: The pointwise approach only looks at a single item at a time in the loss function. The ranking score for an item is independent of other items that are in the list of candidates. We use a similar loss function defined in the subset regression [6], and the loss function is as follows:

$$L = \sum_{u=1}^{M} (\sum_{i=1}^{|T_u|} (f(u,i) - l_i)^2 + \lambda \sum_{j=1}^{6} {c_u^j}^2) \tag{7}$$

Assume there are M users in total, while T_u represents the set of candidate items to be recommended to the user u. $\sum_{j=1}^{6} {c_u^j}^2$ is the regularization terms used to alleviate the overfitting problem, and λ is the regularization rate. l_i can be produced by the user's overall rating on the item (i.e., the partner to date in our speed-dating example). By using stochastic gradient descent (SGD), we are able to learn the vector of user expectations, $\overrightarrow{c_u}$.

2) Pairwise Ranking: Given a pair of the items, the pairwise method comes up the optimal ranking for this pair and compares it to the ground truth. It looks at a pair of the items at a time in the loss function and the goal becomes minimizing the number of inversions in the rankings. The corresponding loss function can be denoted by Equation 8.

$$L = \sum_{u=1}^{M} (\sum_{i=1}^{|T_u|-1} \sum_{\substack{k=1 \\ l_i > l_k}}^{|T_u|} \psi(f(u,i) - f(u,k)) + \lambda \sum_{j=1}^{6} {c_u^j}^2) \tag{8}$$

where $\psi(z)$ is the function adopted to convert z to be continuous and differentiable, so that we can use SGD as the optimizer. $\psi(z)$ could be the hinge function $(1-z)_+$ [15], the logistic function $log(1+e^{-z})$ [2] or the exponential function e^{-z} [12]. In our work, we use the exponential function which was also adopted by the learning to rank collaborative filtering [19] in the area of recommender systems.

3) Listwise Ranking: The weakness in the pairwise ranking is that it only compares each pair of the items for the ranking optimizations. By contrast, the listwise method is defined on the basis of all of the items. There are two major techniques to realize the listwise ranking – either optimizing the ranking metrics directly, e.g., ListRank-MF [27], or minimizing a loss function that is defined based on the properties of rankings to be achieved, such as ListMLE [32].

In our work, we choose to optimize the normalized discounted cumulative gain (NDCG) metric [30] directly. NDCG is a measure from information retrieval, where positions are discounted logarithmically. Assuming each user u has a "gain" g_{ui} from being recommended an item i. The gain information can be obtained from the user preferences on the items in the area of recommender systems, while it may refer to the relevance of the documents in the area of information retrieval. The average Discounted Cumulative Gain (DCG) for a list of J items is defined as shown in Equation 9.

$$DCG = \frac{1}{N} \sum_{u=1}^{N} \sum_{j=1}^{J} \frac{g_{uij}}{max(1, log_b j))} \tag{9}$$

NDCG is the normalized version of DCG given by Equation 10, where DCG^* is the maximum possible DCG.

$$NDCG = \frac{DCG}{DCG^*} \tag{10}$$

Genetic and evolution algorithms have been demonstrated as effective solutions for listwise ranking in the area of information retrieval [35]. In our work, we use the $f(u,i)$ function to rank the items, and adopt differential evolution [29] and particle swarm optimization (PSO) [22] to learn the user expectations by maximizing the NDCG metric. Our experiment shows that the PSO can give better solutions.

In addition, we also try the second way which consider user expectations as the variables to be learned in the process of multi-objective optimizations. We will compare these two ways in order to figure out which approach is better.

C. Multi-Objective Optimization

As mentioned before, the multi-objective learning has been applied to the area of recommender systems [13], [26], [46] to achieve the balance among precision, novelty and diversity of the item recommendations.

The multi-objective problem [25] can be defined as follows.

$$Minimize f(\overrightarrow{x}) = [f_1(\overrightarrow{x}), f_2(\overrightarrow{x}), ..., f_k(\overrightarrow{x})] \quad (11)$$

subject to:

$$g_i(\overrightarrow{x}) \leq 0, i = 1, 2, ..., m \quad (12)$$

$$h_i(\overrightarrow{x}) = 0, i = 1, 2, ..., p \quad (13)$$

where $\overrightarrow{x}$ represents the vector of the decision variables. f_i are the objective functions, and g_i and h_i denote the constraint functions. Note that the Equation 11 uses the minimization of the objectives as the example, while it is actually able to solve any minimization and maximization problems.

Any solutions that meet the requirements above are defined as feasible solutions. Assume $\overrightarrow{x_A}$ and $\overrightarrow{x_B}$ are two feasible solutions. We define $\overrightarrow{x_A}$ ***dominates*** $\overrightarrow{x_B}$, if $\overrightarrow{x_A}$ does not present worse objectives than $\overrightarrow{x_B}$ in all the objective functions, and $\overrightarrow{x_A}$ must produce at least one better objectives than $\overrightarrow{x_B}$. The learning process is going to seek the ***Pareto optimal set*** which is a set of non-dominated $\overrightarrow{x}$ as the optimal solutions.

In our case, we need to keep the balance between the utilities and the recommendation performance. Our initial work [44] utilizes an exhaustive search which may not be feasible and efficient in the large-scale data. Our previous work found that we need to consider both the individual utilities (i.e., the utilities in Equation 3 and 4) and the overall utility (i.e., as shown in Equation 5) as the objectives. It is because the utility of one stakeholder may be significantly larger than the ones of other stakeholders, where the utility of other stakeholders may be overwhelmed. We may obtain high overall utility but one of the individual utilities may be significantly lower than the others. It makes us fail to produce fair recommendations from the perspective of different stakeholders.

Therefore, we choose four objectives in our experiments – the utility of the target user, the utility of the partner and the overall utility, as well as precision to indicate the recommendation performance. Note that, we want to maximize all of these objectives in our experiments.

V. Experiments

A. Techniques and Baseline Approaches

We have introduced most of the techniques that we propose in the Section IV. We additionally introduce the multi-objective learning techniques in this section.

We use the MOEA library[2] which is a Java-based open source framework for multi-objective optimization. It defined the whole learning framework, implements the state-of-the-art multi-objective optimization algorithms, and suggests empirical settings for quick experiments. We adopt six mainstream multi-objective learning techniques in the MOEA library:

- NSGA-II [8] is one of the most popular multi-objective learning techniques. It is composed of two principal parts: a fast non-dominated sorting solution part and the preservation of the solution's diversity.
- NSGA-III [9] is an improved version of NSGA-II, in which adopts many new selection mechanisms and it can handle more than two objectives at the same time.
- MSOPS [16] is a multiple single objective which optimize single objectives respectively and aggregate them together to produce the final solution.
- ϵ-MOEA [7] is a steady-state algorithm, meaning only one individual in the population is evolved per step, and uses an ϵ-dominance archive to maintain a well-spread set of Pareto-optimal solutions.
- SMPSO [18] is a multi-objective learning approach based on the particle swarm optimizer (PSO).
- OMOPSO [28] is an improved multi-objective PSO by using crowding, mutation and ϵ-dominance, and it was demonstrated as one of the top PSO methods to address the multi-objective issues.

We use the suggested empirical settings in the MOEA framework for these multi-objective learning approaches. MOEA setups these quick-run environments to avoid complicated parameter tuning in the experiments.

In terms of the baseline approaches, our previous work [44] has demonstrated that the utility-based multi-stakeholder recommenders can outperform multiple baseline methods, including the RECON [20] which is a popular reciprocal recommendation algorithm for online dating. In this paper, our goal is to compare the proposed approaches which learn the user expectations and find optimal solutions by the multi-objective learning techniques with the method in our previous work [44] which uses the known user expectations and find the solution by an exhaustive search. In the exhaustive search method, we vary the value of β (in Equation 5) from 0 to 1 with 0.1 increment in each step to find the optimal option to balance the multiple objectives.

As a summary, we use the overall utility shown in Equation 5 to produce the item recommendations. We can learn the user expectations by either the learning-to-rank methods or a single process in the multi-objective learning. By using the learning-to-rank methods to obtain the user expectations, we

[2]MOEA, http://moeaframework.org/

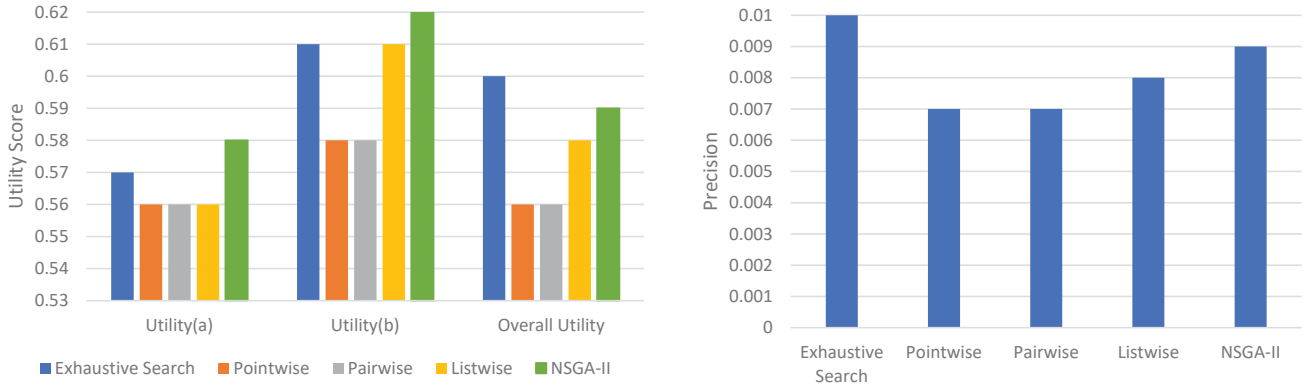

Fig. 1. Solutions By Balancing Utilities and Precision

need to further find the optimal solution (i.e., the value of β in Equation 5) by using several multi-objective optimization techniques. By using the other method, both the user expectations and the β are the variables in the process of multi-objective optimizations. We compare the proposed methods with our previous work [44] to see whether we can obtain equivalent and even better solutions.

B. Evaluation Protocols

Due to the small size of our data, we use a 5-fold cross validation strategy for the evaluation purpose. In terms of the recommendation performance, we use precision as the major metrics in the top-N recommendations, where we examine the performance for both top-5 and top-10 recommendations. Both of them present similar patterns. We only present the results based on the top-5 recommendations in the following sections.

C. Results and Findings

1) Learning User Expectations: One of the major contributions in this paper is that we propose to learn the user expectations if these information are unknown in the data, so that our proposed utility-based models can be generally applied to other domains or the applications. As mentioned in the Section IV-B, there are two ways to learn these user expectations: one way is that we try to learn the vector of user expectations by maximizing the rankings. Another way is that we can consider user expectations as the variables to be learned in the process of multi-objective optimizations. We perform both of these two possible ways to learn the user expectations, and the results can be shown in Figure 2.

The method "Like" in the Figure above represents the baseline method which reports the NDCG by using the Bayesian personalized ranking algorithm [24] based on the numerical ratings in the variable "Like". "NSGA-II" is the multi-objective learning method which learns the user expectations and the value of β in Equation 5 in a single process. We present the results by NSGA-II since it is demonstrated as the best performing one, which will be further discussed in the following analysis. Others in Figure 2 are the learning-to-rank methods which are defined in Section IV-B. We can observe that the pairwise and listwise ranking outperform the pointwise ranking method, and the listwise ranking seems to be slightly better than the pairwise method. In addition, the NSGA-II approach can help obtain similar NDCG as the one by the listwise ranking. Both NSGA-II and listwise ranking can obtain almost the equivalent results as the "Like" method, which tells that we are able to successfully learn the user expectations by using the proposed methods.

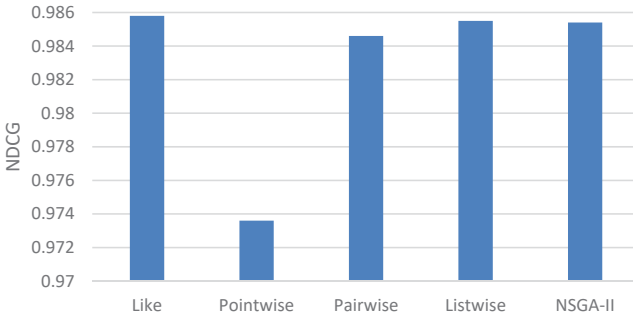

Fig. 2. NDCG Results By Learning User Expectations

2) Multi-Objective Learning: Recall that we have four objectives – utility of the target user, utility of the partner, the overall utility, the precision in the top-5 recommendations. Note that there are two factors which may affect the quality of the results – the value of β, and the vector of user expectations.

We seek the optimal solutions which meet at least two requirements – 1). The individual utilities should be almost equally balanced; 2). The solution may not hurt the overall utility and precision too much. The results by the identified solutions can be shown by Figure 1. We use "Exhaustive Search" to represent our previous work [44], where we manually vary the values of β from 0 to 1. "NSGA-II" in Figure 1 denotes the best performing multi-objective learning method which learns the value of β and the vector of user expectations in a single process. Other methods are the ones which learn the user expectations by the learning-to-rank methods first, and then find the optimal β by the multi-objective learning techniques.

First of all, we focus on the proposed methods which learn the user expectations by using the learning-to-rank approach.

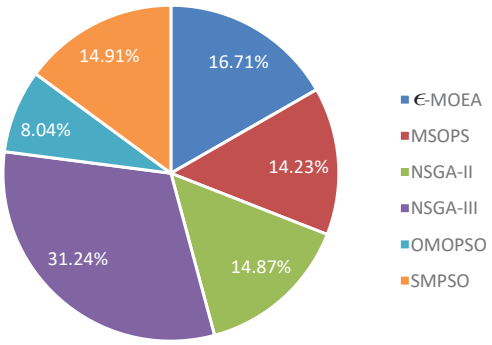

Fig. 3. Distribution of the Number of Optimal Solutions

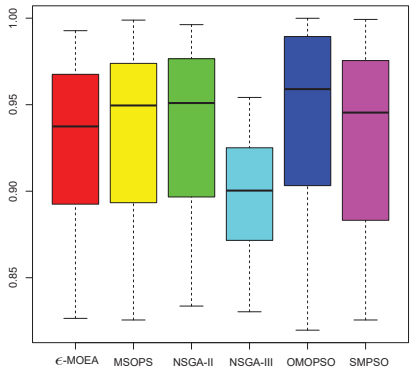

Fig. 4. Distribution of the Similarity of the Solutions

We can observe that the optimal solutions by these methods can find similar results by the exhaustive search in Figure 1. More specifically, they obtain similar individual utilities and the overall utility at the cost of precision. Learning user expectations by the listwise ranking can lead to the best results among the three learning-to-rank techniques. Note that, the proposed methods in our paper can help find solutions with better precision too, but these solutions may hurt either the individual utilities or the overall utility significantly. We believe the results here are meaningful since they are able to balance the utilities by limited loss in the precision values. It also confirms the effectiveness of the learned user expectations. The value of β in the optimal solutions is set as 0.68, 0.67 and 0.66 by using the user expectations learned by the pointwise, pairwise and listwise respectively. All of these solutions are identified by the NSGA-II approach.

Then, we take the "NSGA-II" method into consideration, while it is the best performing multi-objective learning method which learns the value of β and the vector of user expectations in a single process. We can see that it is able to find an optimal solution which helps obtains higher individual utilities at the cost of slightly lower overall utility and the precision results. It outperforms the methods which learn the user expectations by using the learning-to-rank, since it is able to obtain higher utilities and precision values. It infers that there may be several solutions which can produce similar user expectations, but it is better to consider these user expectations as the variables to be learned in the multi-objective optimizations.

In addition, we are interested in which multi-objective learning method is the best choice. Note that the multi-objective optimization techniques will seek a set of the non-dominated solutions as the Pareto optimal set. We blend all of the solutions in the Pareto optimal sets, regardless which learning-to-rank method we used to produce the user expectations. The pie chart in Figure 3 shows the distribution of the number of the identified solutions. There are 2,663 solutions in total by using the three ranking methods with the six multi-objective learning techniques. NSGA-III is able to produce the most number of the solutions, while the number of solutions by the OMOPSO technique is the least.

Furthermore, we explore the quality of these identified solutions. We represent each solution by a vector in which we store the values of the utility of the target user, the utility of the partner, the overall utility and the precision at the top-5 recommendations. We choose the optimal solution by the exhaustive search as the base solution, and calculate the cosine similarity between the base solution and each solution in the Pareto optimal set by each multi-objective learning technique. Afterwards, we analyze the distribution of these similarities and present the results as the box plots shown by the Figure 4. We can observe that the NSGA-III is able to identify the most number of the solutions in Figure 3, but the quality is significantly lower than others, since the similarities between its solutions and the base solution are significantly lower than others. The average quality of the solutions by the NSGA-II method is still one of the best among these approaches, since the median value is generally higher than others. Based on these findings, we conclude that NSGA-II could be the best choice for the multi-objective learning technique in our utility-based multi-stakeholder recommendation models.

VI. Conclusions and Future Work

In this paper, we propose a utility-based multi-stakeholder recommendation algorithm, in which we utilize the user expectations and evaluations to produce the utility scores. In addition, we find that it is possible to learn the user expectations if this information is not observed in the data. More specifically, we discover that the listwise ranking optimization is able to help us learn better user expectations. By applying the multi-objective learning techniques, we are able to find the optimal solutions based on the utility-based multi-stakeholder recommender systems. The identified solution can keep the balance among the individual utilities, the overall utility and the recommendation performance. We also evaluate several multi-objective learning methods, which reveals that NSGA-II could be the best multi-objective optimizer.

In our further work, we would like to explore how to generalize the proposed methods to other applications, rather than the reciprocal recommendations only. Multi-stakeholder recommender system is a promising and novel research direction, and there are many more challenges to be solved

as the future work. For example, there could be conflicting interests among multiple stakeholders. Also, the utilities of each stakeholder may vary in different context situations [37], such as user's emotional states [38], [45]. In addition, there may be other types of the recommendation models rather than the utility-based approaches.

REFERENCES

[1] G. Adomavicius and Y. Kwon. Multi-criteria recommender systems. In *Recommender Systems Handbook*, pages 847–880. Springer, 2015.

[2] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[3] R. Burke. Multisided fairness for recommendation. *arXiv preprint arXiv:1707.00093*, 2017.

[4] R. Burke, N. Sonboli, and A. Ordonez-Gauger. Balanced neighborhoods for multi-sided fairness in recommendation. In *Conference on Fairness, Accountability and Transparency*, pages 202–214, 2018.

[5] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806. ACM, 2017.

[6] D. Cossock and T. Zhang. Statistical analysis of bayes optimal subset ranking. *IEEE Trans on Information Theory*, 54(11):5140–5154, 2008.

[7] K. Deb. A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. *KanGAL Report No 2003002*, 2003.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[9] K. Deb and J. Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th conference on Genetic and evolutionary computation*, pages 635–642, 2006.

[10] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

[11] R. Fisman, S. S. Iyengar, E. Kamenica, and I. Simonson. Gender differences in mate selection: Evidence from a speed dating experiment. *The Quarterly Journal of Economics*, 121(2):673–697, 2006.

[12] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969, 2003.

[13] B. Geng, L. Li, L. Jiao, M. Gong, Q. Cai, and Y. Wu. Nnia-rs: A multi-objective optimization based recommender system. *Physica A: Statistical Mechanics and its Applications*, 424:383–397, 2015.

[14] I. Guy. Social recommender systems. In *Recommender Systems Handbook*, pages 511–543. Springer, 2015.

[15] R. Herbrich. Large margin rank boundaries for ordinal regression. *Advances in large margin classifiers*, pages 115–132, 2000.

[16] E. J. Hughes. Multiple single objective pareto sampling. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 4, pages 2678–2684. IEEE, 2003.

[17] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[18] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna, and E. Alba. Smpso: A new pso-based metaheuristic for multi-objective optimization. In *Computational intelligence in miulti-criteria decision-making, 2009. mcdm'09. ieee symposium on*, pages 66–73. IEEE, 2009.

[19] J.-F. Pessiot, V. Truong, N. Usunier, M. Amini, and P. Gallinari. Learning to rank for collaborative filtering. In *Proceedings of the International Conference on Enterprise Information Systems*, pages 145–151, 2007.

[20] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay. Recon: a reciprocal recommender for online dating. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 207–214. ACM, 2010.

[21] L. Pizzato, T. Rej, T. Chung, K. Yacef, I. Koprinska, and J. Kay. Reciprocal recommenders. In *8th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems, UMAP*, 2010.

[22] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.

[23] M. J. Polonsky and M. R. Hyman. A multiple stakeholder perspective on responsibility in advertising. *Journal of Advertising*, 36(2):5–13, 2007.

[24] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.

[25] M. Reyes-Sierra, C. C. Coello, et al. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2(3):287–308, 2006.

[26] M. T. Ribeiro, A. Lacerda, A. Veloso, and N. Ziviani. Pareto-efficient hybridization for multi-objective recommender systems. In *Proceedings of the 6th ACM conference on Recommender systems*, pages 19–26, 2012.

[27] Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the 4th ACM conference on Recommender systems*, pages 269–272, 2010.

[28] M. R. Sierra and C. A. C. Coello. Improving pso-based multi-objective optimization using crowding, mutation and ϵ-dominance. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 505–519. Springer, 2005.

[29] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[30] H. Valizadegan, R. Jin, R. Zhang, and J. Mao. Learning to rank by optimizing ndcg measure. In *Advances in neural information processing systems*, pages 1883–1891, 2009.

[31] H. Wenxing, C. Yiwei, Q. Jianwei, and H. Yin. ihr+: A mobile reciprocal job recommender system. In *Computer Science & Education (ICCSE), 2015 10th International Conference on*, pages 492–495. IEEE, 2015.

[32] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th conference on Machine learning*, pages 1192–1199, 2008.

[33] P. Xia, B. Liu, Y. Sun, and C. Chen. Reciprocal recommendation system for online dating. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 234–241. ACM, 2015.

[34] M. Yagci and F. Gurgen. A ranker ensemble for multi-objective job recommendation in an item cold start setting. In *Proceedings of the Recommender Systems Challenge 2017*, page 2. ACM, 2017.

[35] J.-Y. Yeh, J.-Y. Lin, H.-R. Ke, and W.-P. Yang. Learning to rank for information retrieval using genetic programming. In *Proceedings of SIGIR Workshop on Learning to Rank for Information Retrieval*, 2007.

[36] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

[37] Y. Zheng. A revisit to the identification of contexts in recommender systems. In *Proceedings of the Conference on Intelligent User Interfaces Companion*, pages 133–136. ACM, 2015.

[38] Y. Zheng. Adapt to emotional reactions in context-aware personalization. In *EMPIRE@ RecSys*, pages 1–8, 2016.

[39] Y. Zheng. Criteria chains: A novel multi-criteria recommendation approach. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 29–33. ACM, 2017.

[40] Y. Zheng. Multi-stakeholder recommendation: Applications and challenges. *arXiv preprint arXiv:1707.08913*, 2017.

[41] Y. Zheng. Exploring user roles in group recommendations: A learning approach. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, pages 49–52. ACM, 2018.

[42] Y. Zheng. Identifying dominators and followers in group decision making based on the personality traits. In *Companion Proceedings of the 23rd International on Intelligent User Interfaces: 2nd Workshop on Theory-Informed User Modeling for Tailoring and Personalizing Interfaces*, 2018.

[43] Y. Zheng. Personality-aware decision making in educational learning. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*, page 58. ACM, 2018.

[44] Y. Zheng, T. Dave, N. Mishra, and H. Kumar. Fairness in reciprocal recommendations: A speed-dating study. In *Adjunct Proceedings of the ACM conference on User Modelling, Adaptation and Personalization*. ACM, 2018.

[45] Y. Zheng, B. Mobasher, and R. Burke. Emotions in context-aware recommender systems. In *Emotions and Personality in Personalized Services*, pages 311–326. Springer, 2016.

[46] Y. Zuo, M. Gong, J. Zeng, L. Ma, and L. Jiao. Personalized recommendation based on evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 10(1):52–62, 2015.