Department of Computer Science
George Mason University Technical Reports

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu/ 703-993-1530

A Group Package Recommender Based on Learning Group Preferences, Multi-Criteria Decision Optimization, and Voting

Hanan Mengash and Alexander Brodsky
{hmengash, brodsky}@gmu.edu

Technical Report GMU-CS-TR-2015-6

Abstract

This paper proposes a Group Package Recommender (GPR) framework, which provides recommendations on dynamically defined packages of products and services. It focuses on extending recommender systems in three ways: (1) to consider composite, rather than atomic, recommendations; (2) to deal with multiple, rather than single, criteria associated with recommendations; and, most importantly, (3) to support groups of users rather than individual users. This framework is based on: (1) defining the space of alternatives; (2) eliciting the utility function for each individual decision maker; (3) estimating the group utility function; (4) using the group utility function to find an optimal recommendation alternative; (5) constructing a set of diverse recommendations which contains the optimal recommendation alternative; and (6) applying alternative voting methods from social choice theories, to refine the recommendations. To evaluate the group recommender performance under each applied voting method, a preliminary experimental real-world user study is conducted, which shows that the proposed framework is able to produce a small set of recommendations that retains near optimal recommendations in term of precision and recall.

1. Introduction

Recommender systems aim to help users making effective product and service choices especially over the Internet. They are applied in a variety of applications and have proven to be useful in predicting the utility or relevance of a particular item and providing personalized recommendations. While state-of-the-art recommender systems focus on atomic (single) products or services, and on individual users (e.g. [1-3]), this paper focuses on extending recommender systems in three ways: (1) to consider composite, rather than atomic, recommendations; (2) to deal with multiple, rather than single, criteria associated with recommendations; and, most importantly, (3) to support groups of users rather than individual users. Examples of this new class of recommender systems include group travelling package recommenders, public policy and budget recommendations, and health care plan selection by organizations. These systems' recommendations are composite, e.g. a travel recommendation may involve interrelated air reservation, accommodation, activities, car rental, etc. They are also associated with multiple criteria such as cost, benefit, enjoyment, satisfaction, risk, etc. Finally, there is often a need to support a group of diverse users/decision makers who may have different, or even strongly conflicting, views on weights for different criteria. The challenges for group recommender systems are considerably more complex than for individual user recommenders, as described in [4]. One of the reasons for this complexity is the need to develop methods to effectively aggregate users' preferences in a way that maximizes the group's satisfaction, is fair, and is easy to use.

There has been extensive work on recommender systems mostly focused on single-user rather than group. More recently, researchers have proposed group recommenders in different domains and applications that used different strategies to aggregate individual preferences into a group model. Common examples of group recommender systems appear in social entertainment include: finding a movie for a group of friends (e.g. [5, 6]); finding songs to play at a shared public space (e.g. [7, 8]); finding a TV show for a family (e.g. [9, 10]); or finding tourist attraction for a group of tourists (e.g. [11, 12]).

However, all of the above group recommender systems were designed for atomic products or services rather than for automatically constructed packages of product and services. Package recommendations present a unique challenge because they make the recommendation space very large, or even infinite, and implicitly, rather than explicitly, defined. In addition, the majority of recommender systems rely on a single ranking or utility score, whereas in many applications there are multiple criteria that need to be taken into account.

Recently, there has been some research on package recommendations [13-15]. However, they do not consider and/or use dynamic preference learning and decision optimization. The CARD Framework [16] and the COD framework [17] support packages of product and service definitions, and provide recommendations based on dynamic preference learning and decision optimization. However, both CARD and COD are recommender systems for individuals rather than groups.

We further detail the other related work and research gap in Section 2.

Addressing the above outlined limitations is exactly the focus of this paper. More specifically, the contributions of this paper are two-fold. First, we develop and propose the Group Package Recommender (GPR) framework, based on multi-criteria decision optimization and voting to address the outlined limitations. The framework works on a very large, or even infinite, recommendation space, which is implicitly defined by a constraint representation of the CARD framework [16]. We consider six group decision-making methods for group decision process. Three of them are based on known and commonly used aggregation strategies namely Average, Least Misery, and Average Without Misery strategies [18]; two are existing voting methods based on individuals' ranking namely Instant Runoff Voting (IRV) method, and Hybrid Condorcet-IRV method. We also develop and propose a new aggregation strategy that we call Structurally-Adjusted Average, which takes into account the influence of decision makers within the group and the dissimilarity of opinions among them.

However, group decision-making methods can be applied only when there are a small number of alternatives to vote on. Whereas, in the case of composite alternatives, the search space of recommendations is exponentially large in the number of recommendation components, or even infinite if some choices are continuous. Therefore, it is impractical to use a voting method on such space directly. The idea of the proposed GPR framework is to filter-out the very large original recommendation space into a very small set of near optimal and diverse alternatives, which can then be refined through group decision-making methods. On one hand, it is important that these alternatives be optimal, or near optimal, in terms of the estimated group utility function. On the other hand, since the group utility is only an estimate, it is also important to have alternatives that are sufficiently diverse in terms of individual decision makers' preferences. To do that, we follow six steps: (1) eliciting the utility function for each member of the group; (2) estimating the group utility function; (3) using the group utility function to find an optimal recommendation alternative; (4) diversity layering to generate a diverse set of l recommendations which contains the optimal recommendation alternative; (5) ranking the set of l recommendations by each individual; (6) applying a group decision-making method to refine the final top-k diverse recommendations. Note that the group utility estimation is parameterized based on the final target group decision-making method.

Second, we conduct a preliminary experimental real-world user study to evaluate the proposed framework performance under each applied group decision-making method. In this study we use alternative methods to model the "actual" group preferences in order to fit the choice of the group decision-making method used in the framework. The experimental study shows that for each target group decision-making method, the average precision and recall achieved by the proposed GPR framework for the top-1 recommendations were exactly the same as the ideal precision and recall (which are obtained under the assumption of complete knowledge), and that they were between (0 to 15%) off from the ideal for the top-2 recommendations, and between (8 to 27%), and (23% to 34%) for the top-3, and the top-4 recommendations respectively.

Preliminary extended abstracts for parts of this paper were presented in conferences [19, 20].

The rest of the paper is organized as follows: Section 2 details the related work and research gap. Section 3 presents an overview of the proposed GPR framework. Section 4 gives an overview of group decision-making methods that are used in the paper, and propose the Structurally-Adjusted Average that we develop. Section 5 explains the user utility functions' extraction. Section 6 explains the group utility estimation. Section 7 presents the optimization and diversity layering. Section 8 discusses the preliminary experimental study for the purpose of evaluating the framework. Finally, Section 9 concludes the paper and discusses some of our future work.

2. Related Work and Research Gap

There have been a wide number of works addressing group recommenders in different domains and applications in the past two decades which used different strategies to aggregate individual preferences into a group model [18]. For example, PolyLens [5] is a group movie recommender that is extended from the MovieLens system, and uses the Least Misery strategy which takes the minimum of individual ratings to avoid "misery" for members. The authors addressed some important issues for group recommenders, e.g., groups' privacy, members' rights, and its social value functions. MusicFx [7] is a group recommender that chooses background music to suit a group in a fitness center. To aggregate a group preference, it uses an average without the minimum rating. Intrigue [11] recommends tourist attractions to groups of users by using the Weighted Average strategy and socio-demographic information about the participants. It takes the preferences of relatively homogeneous subgroups, e.g. children, into account, where each subgroup may have a different degree of influence on the group preferences' estimation. Yu's TV Recommender [9] selects a TV program for a group of users depending on the average of individuals' rating of program features. Travel Decision Forum [4] allows each group members to view the preferences of other members to help the group reaching an agreement on the desired features of a joint holiday. E-Tourism [12] is a web-based services to make group recommendations of tourist activities. To compute the group profile, it uses three different mechanisms: aggregation, intersection and incremental intersection. The Collaborative Advisory Travel System (CATS) [21] is a critique-based group recommender that helps a group of users plan a joint ski holiday, by allowing users to view ski packages and critique their features. The system then recommends a new ski package based on these critiques. The work of [10] proposes to use a voting mechanism to recommend a TV show to a group of people. Specifically, it focuses on the Range voting method, in which users assign ratings within a specified range for items, and the item with the maximum total ratings is recommended to the group. Finally, some recent group recommenders have been implemented on Facebook. For example, GroupFun [8] is a music group recommender that recommends a common set of music items to groups. It uses voting algorithms to state users' true preferences and aggregates them based on the probabilistic weighted sum method. Happy Movie [6], is another group recommender application on Facebook that recommends a movie to groups.

However, most of the above group recommender systems require specific group characteristics rather than provide a general framework for the development of group recommender systems. For instance, the aggregation method in [9] is applicable when the group is quite homogenous, while [5] worked well only for small size groups.

In addition, the majority of these group recommender systems assume that individual preferences are already known [18]. However, [21] is the only known group recommender that assumes that users' preferences are not known. It is based on the members' critiques on desired package features which requires an experience in the package features that is not always possible.

Furthermore, many group recommender systems are intrusive and require significant feedback from users. For example, Travel Decision Forum [4] and CATS [21] require the group to negotiate the group model. While feedback continues to be a main factor in the recommender system concept, it might be better to implicitly extract information from users.

In addition, most of the above group recommender systems aggregate preferences without using the fairness criteria. For instance, in [4] and [9], group members whose preferred features are not selected, are "left out" and not compensated by other desirable features. In addition, using the Average and Plurality Voting strategies such as in [22] does not help avoid the fairness issues.

Furthermore, most of previous work based on aggregation strategies which always combine the members ratings in the same way without considering how group's members interact with each other. In fact, when aggregating the opinions of individual members, it is natural to have members with more influence than others, i.e. users who have authority, more expertise, or are more trusted. These members must be treated differently in order to improve the group decision-making process. Work [23], is a family-based recipe recommender that focuses on the most appropriate recommendation strategy and user weighting model. Its evaluation showed that the best performance of group recommendations is obtained when the individual data of group members are aggregated in a weighted manner. However, it (like the majority of previous work) focused on the content relevance of group members, and ignored the key characteristics within the group such as the size of the group and interest dissimilarity among group members which resulting in sub-optimal group recommendations.

As proved in work [24, 25], the performance of group recommender systems discovered to depend on group characteristics. Based on these findings, group recommenders could incorporate the content relevance of group members and the group's key characteristics to perform more accurate recommendations. By using rank aggregation techniques, the work of [26] addressed the interest dis(similarity) among group members and observed that the more alike the group members are, the more effective the group recommendations are. It also addressed the affect of the group's size on the group recommender system and showed that the group recommendations are less effective than the individual recommendations only for large groups (of size 8), but this difference is very small for groups of moderate size (2, 3, and 4). Work [24] proposed a group recommendation method that studied the key characteristics of groups and proposed a group consensus function that captured the social, expertise, and interest dissimilarity among multiple group members; The work of [25] took both item relevance to a group and disagreements among group members into accounts for their proposed group recommender system.

However, none of the above group recommender systems were designed for packages of product and services, which makes the recommendation space very large, or even infinite, and implicitly, rather than explicitly, defined.

Recently, there has been a host of research that supports packages recommendations [13-15]. However, they do not consider and/or use dynamic preference learning and decision optimization. The CARD Framework [16] and the COD framework [17] support packages of product and service definitions, and provide recommendations based on dynamic preference learning and decision optimization. The packages of services in CARD are characterized by a set of sub-services, which, in turn, can be package or atomic. CARD uses a decision-guidance query language [27, 28] to define recommendation views, which specify multiple utility metrics, as well as the weighted utility function. The CARD packages of services are described using the constraint representation, following [29-34]. COD is based on CARD, and provides an efficient method to elicit individuals' utility functions.

However, both CARD and COD are recommender systems for individuals rather than groups.

In addition, the majority of recommender systems rely on a single ranking or utility score, whereas in many applications there are multiple criteria such as cost, quality, enjoyment, satisfaction, risk, etc., that need to be taken into account. Recently, multi-criteria ranking has been explored in recommendation set retrieval [35-37]. These methods choose a set of alternatives based on the distance measure calculated for each of the multiple criteria. Multi-criteria ranking can support both similarity and diversity based ranking, however, as mentioned in [16], these methods are based on distance measures to increase the quality of each individual recommendations, which competes with the ability diversify recommendations. In addition, they focus on individual users rather than groups of users.

Multi-Criteria Decision Making (MCDM) techniques have been extensively studied in the field of decision science. The two main families in the MCDM techniques are those based on the Multi-Attribute Utility Theory (MAUT) [38, 39], and the Outranking methods [40, 41].

The family of MAUT methods based on aggregating the different criteria into a function that has to be maximized. It includes: The Simple Multi-Attribute Rating Technique (SMART) [42, 43], which is the simplest form of the MAUT methods; The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [44]; and the Analytic Hierarchy Process (AHP) [45], which based on pairwise comparisons to determine the criteria weights. The two main families of the Outranking methods are ELECTRE [40, 41] and PROMETHEE methods [46, 47].

The vast majority of multi-criteria recommender systems (e.g., [17, 37, 48, 49] use the MAUT approach, which provides prediction in the form of additive utility function. In addition, only few of existing multi-criteria recommender systems (e.g., [50, 51]) use Multi-Objective mathematical programming methodologies, where finding the Pareto optimal solution for the optimization problem is the goal for these approaches. Similarly to the previous approach, only few of existing multi-criteria recommender systems (e.g., [52]) use the Outranking Relations approach, where preferences are expressed as a system of outranking relations between the items.

Note that while this paper focuses on composite products or services, and supports group of users, all of the examples of existing multi-criteria recommender systems that mentioned above focus on atomic (single) products or services, and on individual users.

There are several approaches to aggregate individuals' utility functions. Some earlier MAUT methods of group decision are reviewed by [53], and used by number of works, such as: The simple additive theory to aggregate the individuals' utility functions proposed on [54]; The Simple Additive Weighting (SAW) [55]; The MAUT group decision model in [56], which considers both preferential differences and preferential priorities to the model construction; and the use of weighted algebraic means, which is applied in the WINGDSS software [57]. The group utility function computed in WINGDSS is appropriate in the respect of satisfying the axioms given in [58].

In this paper, we extract the group utility function based on the additive multi-attribute group utility function proposed on earlier literature (e.g., [58, 59] to aggregate the individuals' utility functions. However, the aggregated utility function is only an approximation, and using it directly may limit the flexibility of decision makers to refine their choices.

In addition, the AHP [60] and the outranking methods are extended to group decision support; for example, [61] developed a PROMETHEE technique for group decision support, and [62] developed a method for group decision support based on ELECTRE methodology.

However, most of the techniques mentioned above focuses on decision making problems where the number of the criteria and alternatives is finite, and explicitly defined.

3. Overview of The Proposed GPR Framework

In this section, we first describe the recommendation space, then, we explain the recommendation process implemented by the proposed GPR framework and the intuition behind this process.

Recommendation space R, consists of composite products and services; each recommendation alternative $a \in R$ is mapped to a utility vector $\vec{u} = (u_1 \dots, u_n)$from an n dimensional utility space, such that: $\forall_i, 1 \le i \le n$, $u_i: R \to [0,1]$. The components of a utility vector $\vec{u} = (u_1, u_2, \cdots, u_n)$, are associated with criteria such as Enjoyment, Saving, Location attractiveness, etc., which are previously defined. Each criterion has an associated domain D_i, $1 \le i \le n$, and each domain D_i has a total ordering "better than" denoted $\succcurlyeq_{Di}$. For example, for domain Saving, $a_1 \succcurlyeq_{\text{Saving}} a_2 \Leftrightarrow a_1 \ge a_2$.

For a given group of m decision makers, the utility of each decision maker j, is denoted by: $\forall_j, 1 \le j \le m$, $U_j: [0,1]^n \to [0,1]$, and the group utility is denoted by: $U: [0,1]^n \to [0,1]$.

U_j and U define a utility associated with each alternative $a \in R$. Therefore, the user recommendation alternative utility for recommendation a is defined by: $RU_j: R \to [0,1]$, where $RU_j(a) = U_j(u_1(a),\dots,u_n(a))$, and the group recommendation alternative utility is defined by: $RU: R \to [0,1]$, where $RU(a) = U(u_1(a),\dots,u_n(a))$.

The recommendation process implemented by the proposed GPR framework is depicted in Fig.1.

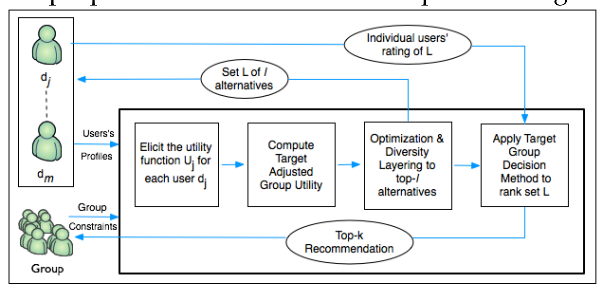

Fig. 1 GPR Framework

As shown in the diagram, the process starts when a group of decision makers submits a request to the group recommender. This request specifies the group's decision constraints on recommendation alternatives. To generate top-k recommendations, the recommender follows six steps: (1) eliciting the utility function for each member of the group; (2) estimating the group utility function; (3) using the group utility function to find an optimal recommendation alternative; (4) diversity layering to generate a diverse set of l recommendations which contains the optimal recommendation alternative; (5) ranking or rating (depending on the group decision-making method used in the last step) the set of l recommendations by each individual; (6) applying a group decision-making method to refine the final top-k diverse recommendations.

Before we discuss each of these steps in detail, we describe the intuition behind this process. First, we apply alternative group decision-making methods to make the final recommendations for a group of decision makers. Different group decision-making methods are used by different people, and usually, the decision of which method to use is depend on the domain, groups' characteristics, and what desirable property people want to satisfy.

In this paper, we consider six group decision-making methods, three of them are based on known and commonly used aggregation strategies namely Average, Least Misery, and Average Without Misery strategies [18]; two are existing voting methods based on individuals' ranking namely Instant Runoff Voting (IRV) method, and Hybrid Condorcet-IRV method. We also develop and propose a new aggregation strategy that we call Structurally-Adjusted Average, which takes into account the influence of decision makers within the group and the dissimilarity of opinions among them. Note that applying alternative group decision-making methods is the last step of the process in Fig.1.

However, group decision-making methods can be applied only when there are a small number of alternatives to vote on. Whereas, in the case of composite alternatives, the search space of recommendations is exponentially large in the number of recommendation components, or even infinite if some choices are continuous. Therefore, it is impractical to use a group decision-making method on such space directly. Consequently, we need to restrict the large original space of recommendations to a very small set that is highly relevant to the whole group, so that it can then be refined through voting.

To do the reduction, we apply mathematical optimization to come with a small set of recommendations that are: 1) close to optimal, 2) sufficiently diverse; so that the group members would have enough flexibility. This explains the second last step in Fig.1, (Optimization and Diversity Layering). However, to perform optimization and diversification, we need to be able to estimate the group utility function that captures the whole group's preferences, and this explains the second step in Fig.1. This group utility function is parameterized based on the final target group decision-making method, and it must be based on the utility functions of the individual users, which is also not known to the system and need to be extracted from individuals, and this is the first step in Fig.1.

We now discuss each of these steps in detail starting with an overview of the group decision-making methods that are used in this paper.

4. Overview of Group Decision Methods

In this section, we first review the most commonly used aggregation strategies for group recommender systems namely Average, Least Misery, and Average Without Misery strategies. We then review two existing voting methods, which based on individuals' ranking namely Instant Runoff Voting (IRV) method, and Hybrid Condorcet-IRV method. We then end this section by proposing a new group aggregation strategy, which we develop and call Structurally-Adjusted Average.

4.1. Aggregation Strategies

To illustrate how the existing aggregation strategies work, consider the following example: Let m be the number of users in a group, r_{ji} be the rating of user j for alternative i, which ranges from 10 (really like) to 1 (really hate), then the group rating for alternative i, denoted by GR_i, is shown in Table 1, which is computed using any of the strategies, as explained in the following subsections. Finally, the alternatives will be ranked in descending order based on the resulted group rating values.

Table 1. Example of Group Ratings Using Different Aggregation Strategies

	*a*1	*a*2	*a*3	*a*4	*a*5
User1	10	8	3	7	10
User2	1	7	8	6	7
User3	10	5	2	8	9
GR_i (by Average)	7	6.7	4.3	7	8.7
GR_i (by Least Misery)	1	5	2	6	7
GR_i (by Average Without Misery	-	6.7	-	7	8.7

Average Strategy

This strategy is the most straightforward one, which assumes the same influence of decision makers within the group. As shown in Table 1, it computes the group rating for alternative i by averaging its individual ratings, as follows:

$$\mathrm{GR}_i = \frac{1}{m}\left(\sum_{j=1}^{m} r_{ij}\right) \qquad (1)$$

Least Misery Strategy

This aggregation strategy is applicable when the group recommender system needs to avoid "misery" for members, which may occur by recommending items that are strongly disliked by any of the group members. As shown in Table 1, this strategy computes the group rating for an alternative i as the lowest rating assigned for that alternative by any of the group members, as follows:

$$\mathrm{GR}_I = \min_j(r_{ij}) \qquad (2)$$

Average Without Misery Strategy

This strategy averages individual ratings as in the Average strategy, but the difference here is that those alternatives with any individual rating below a certain threshold are not considered in the group recommendations. Note that for the example in Table 1, the threshold rating is 4.

4.2. Voting Methods

Because of the Arrow's Impossibility Theorem, which is proved by Kenneth Arrow in 1950 [63], and which states that there is no such perfect voting method that is fully fair (i.e. that is satisfying all the consensus desirable properties such as: Majority, Monotonicity, Independence of Irrelevant Alternatives, and the Condorcet criterion), different voting methods are all still used. Usually, the decision of which method to use is depend on what desirable property people want to satisfy, and what seems most fair for the situation [64]. In this paper, we consider two voting methods namely Instant Runoff Voting (IRV) method, and Hybrid Condorcet-IRV method. These methods are relatively a strong resistance to strategic manipulation [65], which we believe is a critical feature. However, other voting methods such as Borda, Kemeny, Copeland, etc., are possible to apply in our proposed framework.

Initial Definitions:

- *One-to-One Comparisons.* Each pair of alternatives is compared to determine which of the two is more preferred. Let $P(a_x , a_y)$ be the number of decision makers who preferred alternative a_x over a_y. If $P(a_x , a_y) > P(a_y , a_x)$, then a_x wins the one-to-one comparison and beats a_y [65] .
- *Condorcet Winner Criterion.* It states that: if there is an alternative that is preferred in every one-to-one comparison with the other alternatives, then that alternative should be the winner and it is called the Condorcet Winner. Formally: An alternative a_x is a Condorcet winner if and only if $P(a_x , a_y) > P(a_y , a_x), \forall a_y \neq a_x$ [65] .

Instant Runoff Voting (IRV) Method

It is a voting method in which each voter ranks the alternatives in order of his preference. For each recommendation alternative, the system counts the number of voters (decision makers) who ranked it as their first choice. If there exists an alternative that has a majority (over 50%), then that alternative is selected for the whole group of voters. Otherwise, the alternative with the least first-place votes is eliminated from the election, and any votes for that alternative are redistributed to the voters' next choice. This procedure is repeated until an alternative exists that obtains a majority of votes among alternatives not eliminated [64, 66]. If there is a tie for last place in numbers of votes, special tie-breaking rules are applied to select which alternative to eliminate [64, 67].

This method is quite resistant to the need for voters to vote strategically for an alternative that is not their true first choice but has a better chance of winning, because in the IRV method, second or third votes still count if first choices are eliminated.

In GPR framework, in order to end with a total order of eliminated alternatives from which the final top-k recommendations are selected and displayed to the group, we use the same IRV method explained above except that the system continues eliminating the last place alternatives even if the winner alternative is declared. Total order associated with the IRV is a list of eliminated alternatives ordered by which round they are eliminated in, starting with the alternative that is eliminated earliest, and ending with the winner alternative (which actually remains in the last round without being eliminated). If an exact tie exists for last place in numbers of votes, the system decides which alternative to eliminate according to the following tie-breaking rules:

- *Rule1:* if the number of decision makers who vote for these alternatives as their first choice = 0, (i.e., the alternatives are not the first choices of any decision maker), then, the first alternative to eliminate is randomly selected.
- *Rule2:* if the number of decision makers who vote for these alternatives as their first choice ≠ 0, (i.e., the alternatives are the first choice of at least one decision maker), then, the alternative from among these tied with the least votes in the previous round is eliminated. If there is still a tie, then look back to the next most recent round and then, if necessary, to further progressively earlier rounds until one alternative can be eliminated.

To illustrate how IRV works, suppose that we have a group of 9 decision makers who initially ranked the generated diversity set of 3 recommendations as shown on Table 2.

Table 2. IRV Initial Votes

Total number of voters	4	2	3
1st choice	A_1	A_2	A_3
2nd choice	A_2	A_3	A_2
3rd choice	A_3	A_1	A_1

It is clear from Table 2 that alternative A_2 has the least first-choice votes, which is 2, so by applying the IRV method, alternative A_2 is eliminated in the first round shifting everyone's options to fill the gaps (see Table 3).

Table 3. IRV Round 1

Total number of voters	4	2	3
1st choice	A_1	A_3	A_3
2nd choice	A_3	A_1	A_1

Finally, A_3 has the majority votes, and wins the election under the IRV method.

By analyzing the initial preferences in Table 2, the one-to-one comparisons are as follows: (A_1 vs A_2: A_2 beats A_1), (A_1 vs A_3: A_3 beats A_1), (A_2 vs A_3: A_2 beats A_3). So even though A_2 had the smallest number of first-place votes over A_1 and A_3, it is the Condorcet Winner.

From the previous election example, we can immediately notice that IRV violates the Condorcet Criterion. A_2 is the Condorcet Winner, being preferred in every one-to-one comparison with the other alternatives, and yet lost the election and is eliminated in the first round. Which means that applying the standard IRV method on the ranked set of *l* recommendation to refine the final top-k recommendation may result in a Condorcet winner alternative being excluded from the choice set. In order to avoid this issue, a Hybrid Condorcet-IRV method can be applied instead, as follows:

Hybrid Condorcet-IRV Method

This method makes a use of both Condorcet's pairwise comparisons principle and the IRV method, similar to the Benham method mentioned in work [65] but with some differences. The method checks if an alternative exists that beats all other alternatives by one-to-one comparison (Condorcet winner), it will be moved to the first place in a winner list (*W*), otherwise, the IRV method, described above, will be applied and any eliminated alternative is moved to an eliminated list (*E*). This process is repeated on the remaining alternatives until no more alternatives remain. Finally, the method ends with two lists: list *W* of alternatives in descending order by the decision makers' preferences, and list *E* of alternatives ordered in the opposite way. By reversing the order of list *E* alternatives, and merging them below the alternatives in list *W*, we end with a list of alternatives in descending order from which we can select the top-*k* recommendations.

To illustrate how the hybrid Condorcet-IRV method works, see Fig. 2.

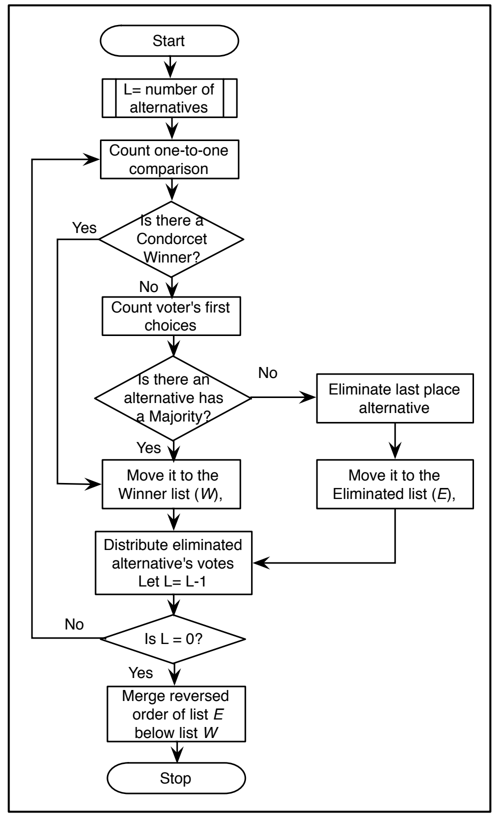

Fig. 2 Flowchart illustrates the Hybrid Condorcet-IRV method

For example, suppose that we have a group of 9 decision makers who initially ranked 5 alternatives as shown on Table 4. By counting the one-to-one comparisons among the 5 alternatives we found that A_2 beats all other alternatives. Even though A_2 had the smallest number of first-place votes over A_1 and A_5, it is the Condorcet Winner and is moved to list (*W*) shifting everyone's options to fill the gaps (see Table 5).

Table 4. Initial votes

Total number of voters	4	2	3
1st choice	A_1	A_2	A_5
2nd choice	A_2	A_3	A_4
3rd choice	A_3	A_5	A_2
4th choice	A_4	A_1	A_1
5th choice	A_5	A_4	A_3

In Round 2, there is no Condorcet winner, so the method looks for an alternative that has a majority (> 50%) of first-place votes. Since no alternative has a majority, the alternative with the least first-choice votes, A_4, is eliminated and moved to the eliminated list (E) (see Table 6).

Table 5. Round 1

Total number of voters	4	2	3
1st choice	A_1	A_3	A_5
2nd choice	A_3	A_5	A_4
3rd choice	A_4	A_1	A_1
4th choice	A_5	A_4	A_3

Table 6. Round 2

Total number of voters	4	2	3
1st choice	A_1	A_3	A_5
2nd choice	A_3	A_5	A_1
3rd choice	A_5	A_1	A_3

In Round 3, there is neither a Condorcet winner nor a majority winner, so A_3 is eliminated and moved to list (E), since it has the least first-choice votes (see Table 7).

Table 7. Round 3

Total number of voters	4	2	3
1st choice	A_1	A_5	A_5
2nd choice	A_5	A_1	A_1

In Round 4, A_5 is the Condorcet winner and is moved to list (W), then, only A_1 remains and is moved to list (W) in the last round. Finally, The method ends with list W ordered as A_2, A_5, A_1, and list E ordered as A_4, A_3. By reversing the order of the alternatives in list E and merging them with the alternatives in list W, we get an ordered recommendation list $A_2 \succ A_5 \succ A_1 \succ A_3 \succ A_4$ from which we select the top-k recommendations.
As we can see, one of the advantages of the hybrid Condorcet-IRV method is that it makes the Condorcet winner alternative impervious to elimination.
In the following sub-section we propose a new group aggregation strategy, which we call Structurally-Adjusted Average.

4.3. *The Proposed Structurally-Adjusted Average*

When aggregating individuals' ratings into a group rating, the proposed aggregation strategy takes into account two main factors in group recommender systems: a) the influence of individuals within the group; and b) the dissimilarity of opinion among group members.

To consider the influence of individuals within the group, we use different users' weights when we aggregate the users' ratings to compute the group rating. These weights reflect the expertise degrees of the group members. Since experts in a group often attempt to persuade other group members, their opinions may be weighted more highly than those of other group members. Based on the domain of the group recommender, expertise can be divided into different levels quantitatively in which each member will be assigned, e.g., in the movies domain these levels will be divided based on the number of movies that a group's member has watched from a list of n popular movies [24], while in the traveling domain they will be divided based on the average yearly travelling of each member within the group, as on Table 8.

Table 8. Categorization of expertize levels based on average yearly travel

The average yearly travelling	<= 1	2 - 3	4 - 5	> 5
Expertise level	I	II	III	IV

The normalized expertize level of a group member j is defined as:

$$E(j,G) = \frac{e_j}{\sum_{u=1}^{m} e_u} \qquad (3)$$

where e_u is the absolute expertise level of each group member u and the sum of the relative expertise levels of a group = 1.

To take the expertise factor into account, we compute the weighted average of the individual ratings for alternative i as follows:

$$\mathrm{EGR}_i = \frac{1}{m}\left(\sum_{j=1}^{m} \mathrm{E}_J \, . \, r_{ij}\right) \qquad (4)$$

where E_j is the expertise of each decision maker j, m = |G|, and r_{ij} is the rating of user j for alternative i , and $0 \leq r_{ij} \leq 1$

As suggested by work [24, 25], the overall group rating value of an alternative needs to reflect the degree of consensus in the ratings for such alternative among group members. Therefore, if the weighted average of the individual ratings is the same for any two alternatives, the one that obtains more similarity of opinion among the group members should have higher group rating value than the one with a lower overall group similarity of opinion in order to avoid the misery of some members. Suppose that there are two different alternatives $a1$ and $a2$, and both obtain the same weighted average of the individual ratings, which is computed as in Eq. (4), but the similarity of opinion among the group members over $a1$ is higher than the one over $a2$; and we like to choose only one of these two alternatives to be included in the small set of the top optimal alternatives. Intuitively, we will choose $a1$ to avoid the misery of the members who may extremely dislike $a2$. This dissimilarity of opinion among group members over an alternative tends to be more significant the larger the group is.

To describe the group dissimilarity over an alternative i, we use the *Standard Deviation* i.e.,

$$\sigma(r_{\mathrm{i1}}, \dots, r_{\mathrm{im}}) = \sqrt{\frac{1}{m-1}\sum_{\mathrm{j=1}}^{\mathrm{m}}(r_{\mathrm{ij}} - avg_{\mathrm{i}})^2} \qquad (5)$$

where avg_{i} is the average of all individual ratings for alternative i.

Finally, to reflect both the influence of individuals within the group and the dissimilarity of opinion among them, we compute the group rating for alternative i as:

$$\mathrm{GR_i} = \mathrm{EGR_i} \ . (1 - \delta) \qquad (6)$$

where $\mathrm{EGR_i}$ is the weighted average of the individual ratings for alternative i defined in Eq. (4), and δ represents the dissimilarity penalty that defined as:

$$\delta = \alpha \ . \frac{\sigma}{\sigma_{\max}} \qquad (7)$$

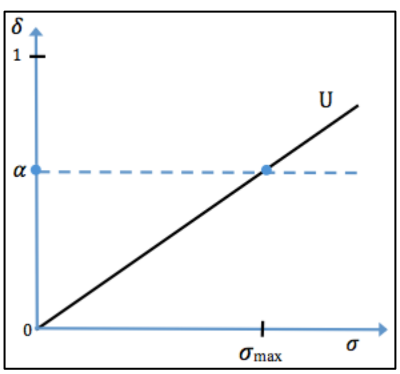

Fig. 3 The Adjusted Group Utility taking into account the dissimilarity of opinion among the group members

where α, $0 \leq \alpha \leq 1$, is a parameter that represents an upper bound for the dissimilarity penalty, (see Fig.3), σ is the *Standard Deviation* as in Eq. (5), and $\sigma_{\max}$ is the maximum possible σ, i.e.,

$$\sigma_{\max} = \max \sigma(r_{\mathrm{i1}}, \dots, r_{\mathrm{im}}) = \frac{1}{2}\sqrt{\frac{m}{m-1}} \qquad (8)$$

where $0 \leq r_{\mathrm{i1}}, \dots, r_{\mathrm{im}} \leq 1$, and clearly, $0 \leq \sigma \leq \sigma_{\max}$

After giving an overview of the group decision-making methods, which are used in the last step of the proposed framework, we now explain the first step, which is eliciting user utility functions, in the following section.

5. Eliciting User Utility Functions

We start by adopting the COD method [17] for eliciting the utility function of each decision maker. This method, as mentioned in [17], starts by viewing a number of distinguishable recommendations in terms of utility vectors to each decision maker. Each recommendation returned stretches the dimension it represents (e.g. Saving), and relaxes on the other dimensions (e.g. Enjoyment, Location attractiveness, etc.). The process continues iteratively updating the utility vector every time, based on the feedback of the decision maker until an exit point is reached (e.g., indicating "no difference" between recommendations presented). Upon exit, the recommendation space will be constructed according to the utility vector learned.

Components of a utility vector $\vec{u} = (u_1, u_2, \cdots, u_n)$, are associated with criteria such as Enjoyment, Saving, Location attractiveness, etc., which are previously defined.

The relative importance the user places in each dimension is modeled by a vector of weights $\vec{w} = (w_1, w_2, \cdots, w_n)$, where $\sum_{i=1}^{n} w_i = 1$.

Each component w_i captures the weight of the i-th dimension according to a decision maker j. So for each decision maker j, the total utility of a recommendation alternative a_{k} w.r.t. the vector $\vec{w}_j$ is defined as:

$$U_j(\vec{\mathrm{u}}) = w_{j1}\, u_1 + w_{j2}\, u_2 + \cdots + w_{jn}\, u_n \qquad (9)$$

where $\sum_{i=1}^{n} u_i = 1$.

6. Estimating the group utility function

In this paper, we assume that all individuals in a group have already agreed on the overall set of criteria but not on their weights. The problem of how to come with unified set of features (criteria) for multiple decision makers has been studied (e.g. [39, 53, 57]), and is outside the scope of this paper.

Estimating the group utility function is the second step in Fig.1. Recall, that a group utility function $\mathrm{U}:[0,1]^n \rightarrow [0,1]$ maps a vector of criteria $u_1 \dots, u_n \in [0,1]$ into a combined group utility $\mathrm{U}(u_1 \dots, u_n) \in [0,1]$. This group utility estimation is parameterized based on the final target group decision-making method that is applied in the last step, as explained in Section 4.

We now discuss in detail how we estimate the group utility using each of these methods starting with the Average Strategy.

Average Strategy

We use the additive utility function, which have been used in many early works (e.g. [58, 59]), as well as some other recent works (e.g. [17, 56]). We estimate the group utility of a recommendation alternative a_k as follows: for each i-th dimension, the individual weights of importance of this dimension is aggregated into the group weights w_i by calculating the algebraic mean of the individual weights as:

$$w_i = \frac{1}{m}\left(\sum_{j=1}^{m} w_{ij}\right) \qquad (10)$$

where $j = 1, \dots, m$, and m is the number of decision makers in the group. The group utility of a recommendation alternative a w.r.t. axis $\vec{\mathrm{w}}_i$ is defined as:

$$\mathrm{U}(\vec{\mathrm{u}}) = \mathrm{w}_1 \mathrm{u}_1 + \mathrm{w}_2 \mathrm{u}_2 + \ldots + \mathrm{w}_n \mathrm{u}_n \qquad (11)$$

where $\sum_{i=1}^{n} w_i = 1$, and $\sum_{i=1}^{n} u_i = 1$.

Least Misery Strategy

In this strategy, the group utility is computed as the minimum utility value for any alternative among group members as follows:

$$\mathrm{U}(\vec{\mathrm{u}}) = \min_j(\mathrm{U}_J(\vec{\mathrm{u}})) \qquad (12)$$

where U_j is the utility of the decision maker j, $1 \le j \le m$, for an alternative a, defined in Eq. (9)

Average Without Misery Strategy

In this strategy, the group utility is computed as in the Average strategy, explained above, but those alternatives with any individual utility below a certain threshold are not considered in the group recommendations, more formally:

$\mathrm{U}(\vec{\mathrm{u}}) = \frac{1}{m}\sum_{j}^{m} \mathrm{U}_J(\vec{\mathrm{u}})$, such that $\forall_j, 1 \le j \le m$, $min_j(\mathrm{U}_j(\vec{\mathrm{u}})) \ge t$.

Structurally-Adjusted Average Strategy

First, to estimate the group utility taking the expertise factor into account, we compute the algebraic mean of the individual criteria-weights as:

$$w'_i = \frac{1}{m}\left(\sum_{j=1}^{m} \mathrm{E}_j \, . \, w_{ij}\right) \qquad (13)$$

where E_j is the expertise of each decision maker j, m = |G|, and w_{ji} is the weight of i=th criterion by the individual decision maker j. Then, for an alternative a, we define the weighted group utility that takes the expertise factor into account, denoted by (EU), as:

$$EU(\vec{\mathrm{u}}) = \mathrm{w}_1' \mathrm{u}_1 + \mathrm{w}_2' \mathrm{u}_2 + \ldots + \mathrm{w}_n' \mathrm{u}_n \qquad (14)$$

where $\sum_{i=1}^{n} w_i = 1$, and $\sum_{i=1}^{n} u_i = 1$.

Second, to describe the dissimilarity of opinion among group members over an alternative, we use the *Standard Deviation* i.e.,

$$\sigma(U_1, \dots, U_m) = \sqrt{\frac{1}{m-1}\sum_{j=1}^{m}(U_j - AU)^2} \qquad (15)$$

where m = |G|, U_j is the decision maker j utility for an alternative a, defined in Eq. (9), and AU is the average utility for an alternative a, as defined in Eq. (11).

Finally, to reflect both the influence of individuals within the group and the dissimilarity of opinion among them, we compute the adjusted group utility as:

$$U = EU \, . (1 - \delta) \qquad (16)$$

where EU is the weighted group utility defined in Eq. (14), and δ represents the dissimilarity penalty that defined in Eq. (7).

Instant Runoff Voting (IRV) Method

First, for each decision maker, we rank the set of alternatives in descending order by her extracted utility U_j. Second, we apply the IRV method, as explained in Section 4, to obtain the group ranked list of alternatives. Finally, we estimate the group utility of each alternative $a \in R$ as:

$$RU(a) = \frac{n-i}{n-1} \qquad (17)$$

where RU(a) = U($u_1(a)$,….,$u_n(a)$), n is the number of the ranked alternatives, and i is the position of an alternative a in the ranked set resulted from IRV method.

Hybrid Condorcet-IRV Method

We estimate the group utility of each alternative $a \in R$ similarly to the estimation process used in IRV method, except that here we applied the Hybrid Condorcet-IRV method instead.

7. Optimization and Diversity Layering

Since it is not practical for decision makers to consider and focus on more than a very small set of recommendation alternatives, the goal of this step is to come up with this small set. On one hand, it is important that these alternatives be optimal, or near optimal, in terms of the estimated group utility function. On the other hand, since the group utility is only an estimate, it is also important to have alternatives that are sufficiently diverse in terms of individual decision makers' preferences.

Note that optimal choices according to the estimated utility may limit the flexibility to diversify recommendations. Hence, there is tradeoff to be made between the two competing goals: optimization and diversity. To find the right "balance", we follow two steps: First, for optimization, we find the optimal choice a_1 by maximizing the estimated group utility, i.e., a_1 = argmax U ($\vec{u}$ (a)), where $a \in R$, $\vec{u}(a)$ is the utility vector, and U ($\vec{u}$ (a)) is the estimated group utility corresponding to vector $\vec{u}$ (a), and is computed by using any of the six group decision-making methods explained above.

Second, for diversification, we adapted the diversity layering method from CARD [16]. However, the dimensions of the utility space in [16] are the original criteria, whereas, we are advocating of using the space of extracted utilities of individual decision makers instead.

The motivation behind this choice is because individuals may not be satisfied if the options presented to them for voting do not include options that are closely related to their preferences. We would like to mimic, as close as possible, the popular group decision-making mechanisms when the alternatives are proposed by individual group members and, therefor, reflect their preferences.

However, the diversity layering method, described below, will still have alternatives that are optimal or within bounded distance from the optimal group utility.

The key idea is to create a subset of divers recommendations that correspond to different individuals' utility functions, while preserving a bounded distance from the optimal group utility score in order to provide the right balance between optimality and diversity. We partition the recommendation space into q layers starting from the layer that includes the optimal recommendation, which maximizes the group utility U. The second layer includes the recommendations that are close to the optimal recommendation having a total utility value no less than the maximum group utility minus ε, where ε corresponds to a percentage of the maximum group utility score. The third layer includes the recommendations indicating a total utility value no less than the maximum group utility minus 2ε. Recommendations in the i-th layer have a utility value no less than the maximum group utility function minus (i-1)ε. Within each layer, we select n recommendations to maximize each dimension of the recommendation space in turn.

To illustrate the diversity layering method, consider the example depicted in Fig. 4.

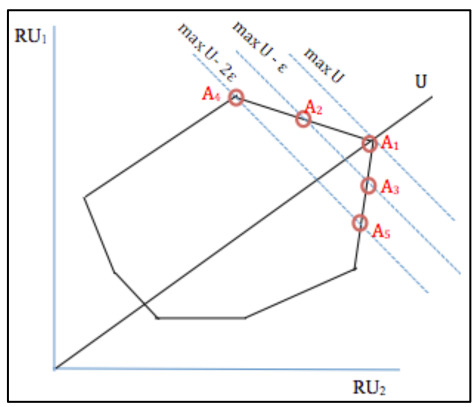

Fig. 4 Diversity Layering

Here, RU_1 and RU_2 are two individual decision maker's utilities, and U is the group utility, which is defined as a linear combination of RU_1 and RU_2. The two-dimensional polyhedron set in the figure depicts all possible utility vectors of recommendations. Among these vectors, A_1 is the optimal recommendation that maximizes U.

The second layer includes recommendations for which U >= max{U} – ε, where ε corresponds to a percentage of max{U}, say 2%. The selected recommendations in this layer are A_2 and A_3 because they maximize RU_1 and RU_2 in turn, which provides diversity while restricting the group utility within its layer preserves the distance from the optimal recommendation. The third layer includes recommendations for which $U >= \max\{U\} - 2\varepsilon$, and the selected recommendations in this layer are A_4 and A_5 which have the maximum RU_1 and RU_2 in turn.

As explained, the diversity layering method generates a set of diverse alternatives by optimizing each user utility function in each layer. Note here that this method may not scale well for large number of decision makers. This issue may occur in large size groups, i.e., if the total number of individuals is greater than the number of the diversity recommendations needed. In this case, the system may reach the needed number of recommendations early, before completing optimization over all the members' utility functions, even if it is still in the first layer. This issue can be solved by clustering large groups into number of homogenous subgroups then diversify recommendations across those subgroups of decision makers, rather than individuals. However this is outside the scope of this paper.

After generating the diversity set of *l* recommendations by using the estimated group utility function and optimizing each user utility function, such recommendations are presented to each individual decision maker in descending order of the group utility according to the estimated group utility function, and each individual decision maker is asked to rank (or rate, based on which group decision-making method is used) the set of *l* recommendations in a way that truly reflects her preferences. The benefit of allowing each member to rank/rate the pre-final results by herself is to avoid the effect of an incorrect estimation of the individual decision maker's utility function in the first step.

This individuals' ranking/rating of the optimal and divers set of recommendations is the input of the group decision-making method, which is applied in the final step of the proposed framework to refine the final top-k divers recommendations, as explained in Section 4.

8. Initial Experimental Evaluation

Experimental Setting

We conducted a preliminary experimental study to evaluate the proposed GPR framework performance, under each applied group decision-making method, in terms of precision and recall metrics. Precision and recall metrics are widely used on information retrieving scenario, recall is the proportion of truly good recommendations that appear in top recommendations, and the precision is the proportion of recommendations that are truly good recommendations [1].

The experimental study involved a total of 67 users, all were graduate students, in 13 groups of different sizes, as follows: 1 group of 2 users; 2 groups of 3 users; 2 groups of 4 users; 4 groups of 5 users; and 2 group of 6 users; 1 group of 9 users; and 1 group of 10 users. The hypothesis of this study was: The proposed GPR framework is able to produce a small set of recommendations that retains near optimal recommendations in terms of precision and recall.

The data of the experimental study was real data about vacation packages, which were extracted from a popular commercial travel website, by submitting a request for a two week vacation in Los Angeles, California, which included a non-stop round-trip airfare from Washington Dulles Airport. All of the returned packages from this website, were extracted keeping only the cost and number of stars (enjoyment) of each package.

Experimental Methodology

In this experimental study, we decided to study our framework's performance under applying different group decision-making methods, but we are not trying to make a judgment of which method is the best one. There has been extensive work on group recommender systems, which explained that some methods are better than others in different situations (e.g., [18, 68]). For this paper, in which the group decision-making method that best fits the type of situation is externally chosen, we are trying to study the accuracy of our system, in spite of all the approximations in the very large space, compared to the ideal accuracy, which is obtained under the assumption of complete knowledge (without approximations).

For the evaluation purpose, we consider for each group only the packages that have been evaluated by all group's members in our previous experimental of work [19, 20]. These evaluations were based on ratings on a scale of 1 to 5, where 5 means "strongly agree", 4 means "Agree", 3 means "neutral", 2 means "disagree", and 1 means "strongly disagree".

For each group, we applied the GPR framework using alternative group decision-making methods, as explained in Section 4 and 6, on the same data set to generate the top-4 recommendations to each group under each method. Finally, to generate the ground truth for each group, we aggregate the actual individual preferences into the group actual overall preferences, using alternative group decision-making methods, in order to fit the choice of the method used in the framework.

To calculate the estimated recall for GPR framework at a given rank (k), we gathered all the packages rated 4 or above from the group ground truth in a set called "Good". Then, for each group, we calculated the estimated recall as:

$$Recall\ (k) = \frac{|\{r \in Good | rank(r) \leq k\}|}{|\{\text{Good}\}|} \tag{18}$$

We then computed the average recall at each rank k, for each group decision-making method, by taking the average of recall (k) among all the 13 groups. The results are shown in Fig. 5 through Fig. 10.

Similarly, we calculated the estimated precision for GPR framework at a given rank (k), for each group, as:

$$Precision\ (k) = \frac{|\{r \in Good | rank(r) \leq k\}|}{k} \tag{19}$$

We then computed the average precision at each rank k, for each group decision-making method, by taking the average of precision (k) among all the 13 groups. The results are also shown in Fig. 5 through Fig. 10.

Experimental Results

The experimental study shows that for top-1 recommendations, the average recall and precision achieved by the proposed GPR framework, under each target group decision-making method, were exactly the same as the ideal. For top-2 recommendations, GPR framework, under each method, obtained recall and precision within 90 % of the ideal, except under the Least Misery method, in which they were within 85 % of the ideal. For top-3 recommendations, GPR framework's recall and precision were within 80 % of the ideal under all methods except the Least Misery and the Average Without Misery methods, in which the framework's accuracy was between 20% to 27% off from the ideal, in terms of recall and precision. However, GPR framework's recall and precision were 23% to 34% off from the ideal for the top-4 recommendations under all methods.

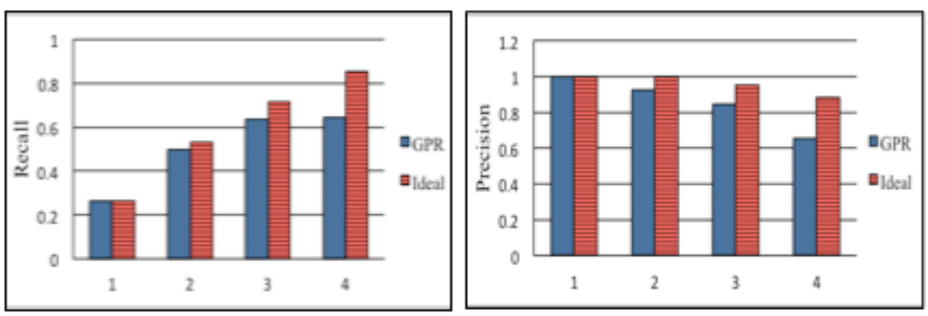

Fig. 5 Average Recall and Precision for the Proposed GPR Framework vs. Ideal, using Average Strategy

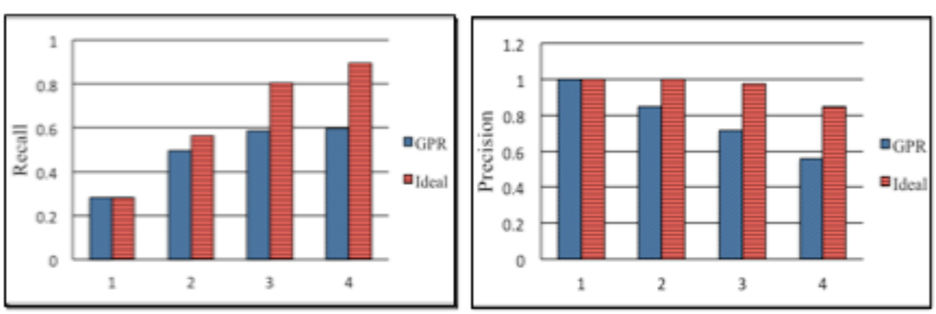

Fig. 6 Average Recall and Precision for the Proposed GPR Framework vs. Ideal using Least Misery Strategy

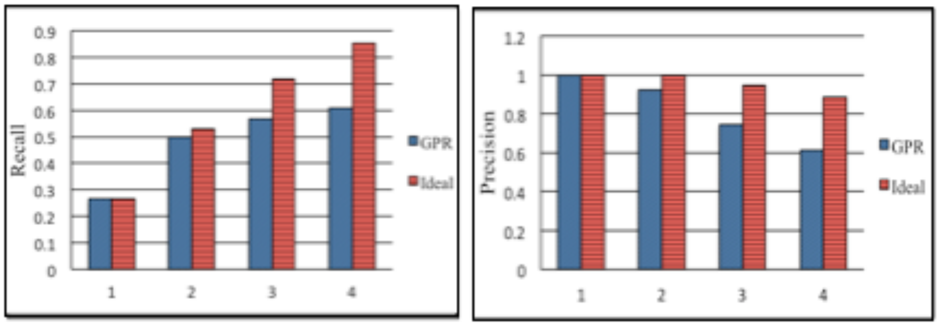

Fig. 7 Average Recall and Precision for the Proposed GPR Framework vs. Ideal, using Average Without Misery Strategy

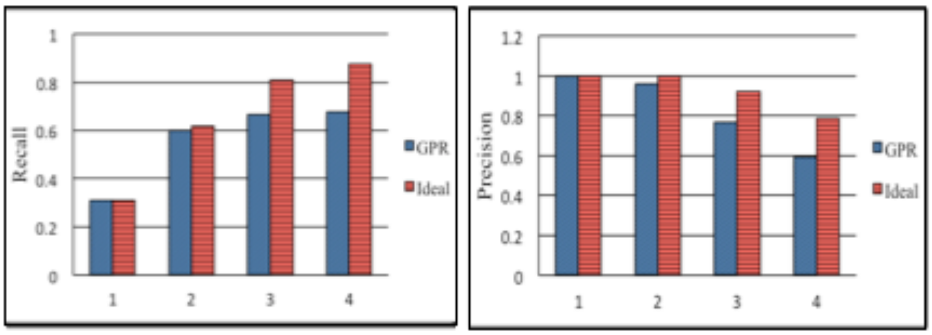

Fig. 8 Average Recall and Precision for the Proposed GPR Framework vs. Ideal, using Structurally-Adjusted Average

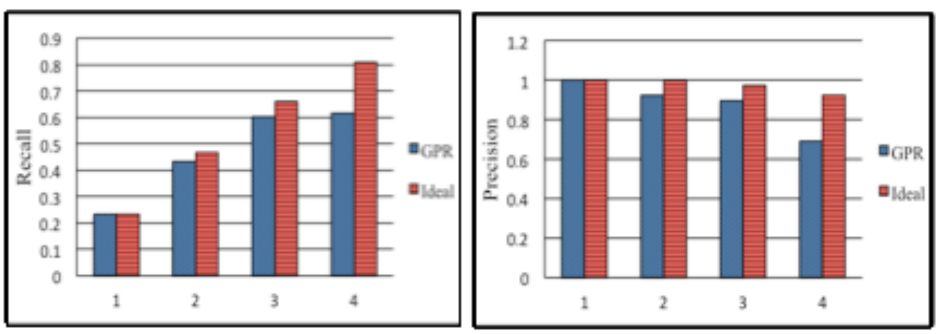

Fig. 9 Average Recall and Precision for the Proposed GPR Framework vs. Ideal, using IRV Method

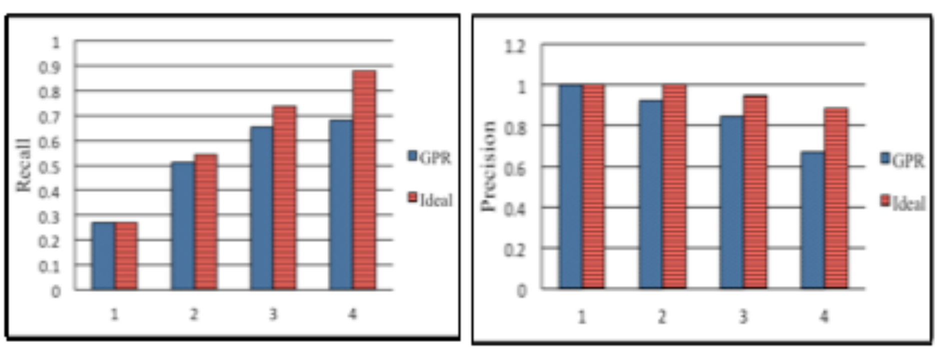

Fig. 10 Average Recall and Precision for the Proposed GPR Framework vs. Ideal, using Condorcet-IRV Method

Statistical Analysis

As we explained above, in this experimentation we are not trying to compare the group decision-making methods against each other, and make a judgment of which method is the best one. Instead, we are trying to study how well our framework did, in spite of all the approximations in the very large space, comparing to the ideal.

Therefore, for the statistical analysis, we calculated the Confidence Interval (CI) for the estimated mean of the percentage differences between the GPR framework's accuracy and the ideal, in terms of recall and precision, using alternative group decision methods. For this calculation, we applied the following formula:

$$Estimated\ mean = sample\ mean \pm (se\ .\ t_{crit}) \qquad (20)$$

where, *se* is the standard error of the mean; and t_{crit} = the two-tailed critical value of t for the 0.05 level of significance. The results are illustrated in Fig. 11.

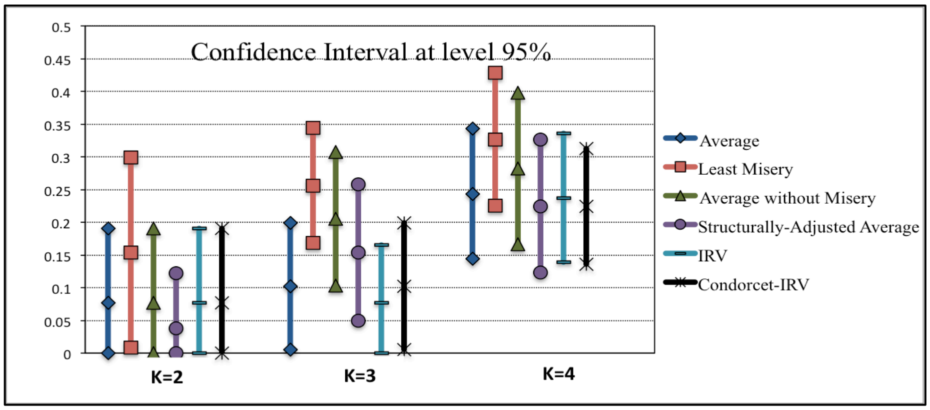

Fig. 11 The Confidence Interval for the estimated mean of the percentage differences between the GPR framework's accuracy and ideal, in terms of recall and precision, using alternative group decision methods.

Note that the CI for the top-1 recommendation (i.e., when k=1) is not shown in Fig.11, because it is equal to 0, i.e., we are 95% confident that the proposed framework's accuracy, using all of the six explained methods, will achieve the ideal accuracy for top-1 recommendations, in terms of recall and precision.

In Fig. 11, markers represent the mean of the percentage differences between the GPR framework's accuracy and the ideal, in terms of recall and precision, using alternative group decision methods. Vertical lines across markers indicate the confidence interval, for instance, Fig.11 reveals that under the Average strategy, for top-2 recommendations, the mean percentage differences between the proposed framework's accuracy and the ideal, will be between 0 to 20%, in terms of recall and precision. In addition, it will not be off more than 30%, and 40% from the ideal, for top3, and top-4 respectively, under all methods except the Least Misery method, in which the accuracy may decrease.

As we mentioned above, there has been extensive work on group recommender systems, which explained that some methods are better than others in different situations. We think that the accuracy of the proposed framework decreases when using the Least Misery strategy, because, as explained in work [5], this method works well only for small size groups, whereas, in our experimental study some groups are relatively large.

9. Conclusions

In this paper, we proposed the Group Package Recommender (GPR) framework that provides a diverse set of recommendations on packages of products and services to a group of users. This framework extended the existing recommender systems in many ways: (1) it considered composite, rather than atomic, recommendations; (2) it dealt with multiple, rather than single, criteria associated with recommendations; and (3) it supported a group of diverse decision makers who may have different, or even strongly conflicting, views on weights for different criteria.

We also conducted a preliminary experimental real-world user study to evaluate the proposed framework performance under each of the six explained group decision-making method. In this study we use alternative methods to model the “actual” group preferences in order to fit the choice of the group decision-making method used in the framework. By calculating the average precision and recall of ranked recommendations resulted from our proposed GPR framework, under each applied group decision-making method, it shows that GPR is able to produce a small set of recommendations that retains near optimal recommendations.

More specifically, the experimental study shows that for each target group decision-making method, the average precision and recall achieved by the proposed GPR framework for the top-1 recommendations were exactly the same as the ideal precision and recall (which are obtained under the assumption of complete knowledge), and that they were between (0 to 15%) off from the ideal for the top-2 recommendations, and between (8 to 27%), and (23% to 34%) for the top-3, and the top-4 recommendations respectively.

Although our framework is designed to be high scalable in terms of alternatives through utility optimization, it may not scale well for large number of decision makers. One reason for this challenge is the fact that eliciting the utility function for each user may not be practical for groups with large number of users. In addition, aggregation methods for large groups may loose accuracy, as indicated in many works (e.g. [25, 26]).

Furthermore, although our initial user study helps us to learn more about the group decision-making process, the problem of obtaining ground truth data of actual group preferences remains.

Many research questions remain open, including: (1) combining a model for automatically inferring the expertise and social dissimilarity among the group members with our recommender system; (2) improving the quality of the framework when considering large heterogeneous groups by splitting large groups into number of homogenous subgroups, then diversify recommendations across those subgroups of decision makers, rather than individuals; (3) Studying the framework’s performance using additional group decision-making methods; (4) Conducting a large-scale evaluation.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, pp. 734-749, 2005.

[2] F. Cacheda, #237, c. Carneiro, D. Fern, #225, ndez, *et al.*, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Trans. Web,* vol. 5, pp. 1-33, 2011.

[3] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges," *ACM Comput. Surv.*, vol. 47, pp. 1-45, 2014.

[4] A. Jameson, "More than the sum of its members: challenges for group recommender systems," presented at the Proceedings of the working conference on Advanced visual interfaces, Gallipoli, Italy, 2004.

[5] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl, "PolyLens: a recommender system for groups of users," presented at the Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work, Bonn, Germany, 2001.

[6] L. Quijano-Sanchez, J. A. Recio-Garcia, and B. Diaz-Agudo, "HappyMovie: A Facebook Application for Recommending Movies to Groups," presented at the Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, 2011.

[7] J. F. McCarthy and T. D. Anagnost, "MusicFX: an arbiter of group preferences for computer supported collaborative workouts," presented at the Proceedings of the 1998 ACM conference on Computer supported cooperative work, Seattle, Washington, USA, 1998.

[8] G. Popescu and P. Pu, "What's the best music you have?: designing music recommendation for group enjoyment in groupfun," presented at the Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts, Austin, Texas, USA, 2012.

[9] Z. Yu, X. Zhou, Y. Hao, and J. Gu, "TV Program Recommendation for Multiple Viewers Based on user Profile Merging," *User Modeling and User-Adapted Interaction,* vol. 16, pp. 63-82, 2006.

[10] L. N. Dery, M. Kalech, L. Rokach, and B. Shapira, "Iterative voting under uncertainty for group recommender systems," presented at the Proceedings of the fourth ACM conference on Recommender systems, Barcelona, Spain, 2010.

[11] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso, "Tailoring the Recommendation of Tourist Information to Heterogeneous User Groups," presented at the Revised Papers from the nternational Workshops OHS-7, SC-3, and AH-3 on Hypermedia: Openness, Structural Awareness, and Adaptivity, 2002.

[12] I. Garcia, L. Sebastia, E. Onaindia, and C. Guzman, "A Group Recommender System for Tourist Activities," presented at the Proceedings of the 10th International Conference on E-Commerce and Web Technologies, Linz, Austria, 2009.

[13] F. Lorenzi, S. Loh, and M. Abel, "PersonalTour: A Recommender System for Travel Packages," presented at the Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 02, 2011.

[14] T. Deng, W. Fan, and F. Geerts, "On the complexity of package recommendation problems," presented at the Proceedings of the 31st symposium on Principles of Database Systems, Scottsdale, Arizona, USA, 2012.

[15] R. Interdonato, S. Romeo, A. Tagarelli, and G. Karypis, "A Versatile Graph-based Approach to Package Recommendation," 2013.

[16] A. Brodsky, S. M. Henshaw, and J. Whittle, "CARD: a decision-guidance framework and application for recommending composite alternatives," presented at the Proceedings of the 2008 ACM conference on Recommender systems, Lausanne, Switzerland, 2008.

[17] K. Alodhaibi, A. Brodsky, and G. A. Mihaila, "COD: Iterative Utility Elicitation for Diversified Composite Recommendations," in *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, 2010, pp. 1-10.

[18] J. Masthoff, "Group Recommender Systems: Combining Individual Models," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., ed: Springer US, 2011, pp. 677-702.

[19] H. Mengash and A. Brodsky, "GCAR: A Group Composite Alternatives Recommender Based on Multi-Criteria Optimization and Voting," in *47th Hawaii International Conference in System Sciences (HICSS)*. 2014.

[20] H. Mengash and A. Brodsky, "DG-GPR: A decision-guided group package recommender with hybrid condorcet-instant runoff voting," in *DSS 2.0 - Supporting Decision Making with New Technologies*, Paris, France, 2014, pp. 317 - 328.

[21] K. McCarthy, L. McGinty, B. Smyth, M. Salam, and #243, "The needs of the many: a case-based group recommender system," presented at the Proceedings of the 8th European conference on Advances in Case-Based Reasoning, Fethiye, Turkey, 2006.

[22] L. M. Campos, J. M. Fern\, \#225, ndez-Luna, J. F. Huete, and M. A. Rueda-Morales, "Managing uncertainty in group recommending processes," *User Modeling and User-Adapted Interaction,* vol. 19, pp. 207-242, 2009.

[23] S. Berkovsky and J. Freyne, "Group-based recipe recommendations: analysis of data aggregation strategies," presented at the Proceedings of the fourth ACM conference on Recommender systems, Barcelona, Spain, 2010.

[24] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra*, et al.*, "Enhancing group recommendation by incorporating social relationship interactions," presented at the Proceedings of the 16th ACM international conference on Supporting group work, Sanibel Island, Florida, USA, 2010.

[25] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu, "Group recommendation: semantics and efficiency," *Proc. VLDB Endow.,* vol. 2, pp. 754-765, 2009.

[26] L. Baltrunas, T. Makcinskas, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering," presented at the Proceedings of the fourth ACM conference on Recommender systems, Barcelona, Spain, 2010.

[27] A. Brodsky, N. Egge, and X. Wang, "Supporting Agile Organizations with a Decision Guidance Query Language," *J. Manage. Inf. Syst.,* vol. 28, pp. 39-68, 2012.

[28] A. Brodsky, M. M. Bhot, M. Chandrashekar, N. E. Egge, and X. S. Wang, "A decisions query language (DQL): high-level abstraction for mathematical programming over databases," presented at the Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, Providence, Rhode Island, USA, 2009.

[29] A. Brodsky, M. Al-Nory, and H. Nash, "Service Composition Language to Unify Simulation and Optimization of Supply Chains," in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, 2008, pp. 74-74.

[30] A. Brodsky, L. Kerschberg, and S. Varas, "Resource Management in Agent-based Distributed Environments," in *Cooperative Information Agents III*. vol. 1652, M. Klusch, O. Shehory, and G. Weiss, Eds., ed: Springer Berlin Heidelberg, 1999, pp. 61-85.

[31] A. Brodsky and H. Nash, "CoJava: A Unified Language for Simulation and Optimization," in *Principles and Practice of Constraint Programming - CP 2005*. vol. 3709, P. van Beek, Ed., ed: Springer Berlin Heidelberg, 2005, pp. 877-877.

[32] A. Brodsky and H. Nash, "CoJava: Optimization Modeling by Nondeterministic Simulation," in

Principles and Practice of Constraint Programming - CP 2006. vol. 4204, F. Benhamou, Ed., ed: Springer Berlin Heidelberg, 2006, pp. 91-106.

[33] A. Brodsky, "Constraint Databases: Promising Technology or Just Intellectual Exercise?," *Constraints,* vol. 2, pp. 35-44, 1997/04/01 1997.

[34] A. Brodsky and V. E. Segal, "The C3 Constraint Object-Oriented Database System: An Overview," presented at the Second International Workshop on Constraint Database Systems, Constraint Databases and Their Applications, 1997.

[35] K. Bradley and B. Smyth, "Improving Recommendation Diversity," ed, 2001.

[36] B. Smyth and P. McClave, "Similarity vs. Diversity," presented at the Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development, 2001.

[37] G. Adomavicius and Y. Kwon, "New Recommendation Techniques for Multicriteria Rating Systems," *IEEE Intelligent Systems,* vol. 22, pp. 48-55, 2007.

[38] P. C. Fishburn, *Utility Theory for Decision Making*: John Wiley & Sons, New York, 1970.

[39] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*: John Wiley & Sons, New York, 1976.

[40] B. Roy, "Classement et choix en présence de points de vue multiples," *RAIRO - Operations Research - Recherche Opérationnelle,* vol. 2, pp. 57-75, 1968.

[41] B. Roy, "ELECTRE III: un algorithme de classement fondé sur une représentation floue des préférences en présence de critères multiples," *Cahiers du CERO,* vol. 20, pp. 3-24, 1978.

[42] W. Edwards, "How to Use Multiattribute Utility Measurement for Social Decisionmaking," *Systems, Man and Cybernetics, IEEE Transactions on,* vol. 7, pp. 326-340, 1977.

[43] W. Edwards and F. H. Barron, "SMARTS and SMARTER: Improved Simple Methods for Multiattribute Utility Measurement," *Organizational Behavior and Human Decision Processes,* vol. 60, pp. 306-325, 12// 1994.

[44] C.-L. Hwang and K. Yoon, *Multiple Attribute Decision Making Methods and Applications: A State-of-the-Art Survey*: Springer Berlin Heidelberg, NewYork, 1981.

[45] T. L. Saaty, *The Analytic Hierarchy Process*: McGraw-Hill, New York, 1980.

[46] J. P. Brans and P. Vincke, "A preference ranking organization method," *Management Science,* vol. 31, pp. 647-656, 1985.

[47] J. P. Brans, P. Vincke, and B. Mareschal, "How to select and how to rank projects: the PROMETHEE method," *European Journal of Operational Research,* vol. 24, pp. 228-238, 1986.

[48] K. Lakiotaki, S. Tsafarakis, and N. Matsatsinis, "UTA-Rec: a recommender system based on multiple criteria analysis," presented at the Proceedings of the 2008 ACM conference on Recommender systems, Lausanne, Switzerland, 2008.

[49] Q. Li, C. Wang, and G. Geng, "Improving personalized services in mobile commerce by a novel multicriteria rating approach," presented at the Proceedings of the 17th international conference on World Wide Web, Beijing, China, 2008.

[50] L. Hsin-Hsien and T. Wei-Guang, "Incorporating Multi-Criteria Ratings in Recommendation Systems," in *Information Reuse and Integration, 2007. IRI 2007. IEEE International Conference on*, 2007, pp. 273-278.

[51] N. Manouselis and C. Costopoulou, "Experimental Analysis of Design Choices in multiattribute Utility Collaborative Filtering," *International Journal of Pattern Recognition and Artificial Intelligence - IJPRAI,* vol. 21(2), pp. 311-332, 2007.

[52] E. Yano, E. Sueyoshi, I. Shinohara, and T. Kato, "Development of a recommendation system with multiple subjective evaluation process models," in *Cyberworlds, 2003. Proceedings. 2003 International Conference on*, 2003, pp. 344-351.

[53] U. Bose, A. M. Davey, and D. L. Olson, "Multi-attribute utility methods in group decision making: Past applications and potential for inclusion in GDSS," *Omega,* vol. 25, pp. 691--706, 1997.

[54] N. Arora and G. Allenby, "Measuring the Influence of Individual Preference Structures in Group Decision Making," *Journal of Marketing Research,* vol. 36, 1999.

[55] C. W. Kirkwood and J. L. Corner, "The Effectiveness of Partial Information about Attribute Weights for Ranking Alternatives in Multiattribute Decision Making," *Organizational Behavior and Human Decision Processes,* vol. 54, pp. 456-476, 4// 1993.

[56] Y.-S. Huang, W.-C. Chang, W.-H. Li, and Z.-L. Lin, "Aggregation of utility-based individual preferences for group decision-making," *European Journal of Operational Research,* vol. 229, pp. 462-469, 9/1/ 2013.

[57] P. Csáki, T. Rapcsák, P. Turchányi, and M. Vermes, "Research and development for group decision aid in Hungary by WINGDSS, a Microsoft Windows based group decision support system," *Decision Support Systems 14,* pp. 205-221, 1995.

[58] R. L. Keeney, "A Group Preference Axiomatization with Cardinal Utility," *Management Science,* vol. 23, pp. 140-145, 1976/10/01 1976.

[59] J. S. Dyer and R. K. Sarin, "Group Preference Aggregation Rules Based on Strength of Preference," *Management Science,* vol. 25, pp. 822-832, 1979.

[60] R. F. Dyer and E. H. Forman, "Group decision support with the Analytic Hierarchy Process," *Decision Support Systems,* vol. 8, pp. 99-124, 4// 1992.

[61] C. Macharis, J. P. Brans, and B. Mareschal, "The GDSS PROMETHEE procedure -- A PROMETHEE-GAIA based procedure for group decision support.." *Journal of Decision Systems,* vol. 7, pp. 283-307, 1998.

[62] J. C. Leyva-López and E. Fernández-González, "A new method for group decision support based on ELECTRE III methodology," *European Journal of Operational Research,* vol. 148, pp. 14-27, 7/1/ 2003.

[63] K. Arrow, "A Difficulty in the Concept of Social Welfare," *Journal of Political Economics,* vol. 58, pp. 328-346, 1950.

[64] Lippman and David, *Math in Society*: CreatSpace Independent Publishing Platform, 2012.

[65] J. Green Armytage, "Four Condorcet-Hare Hybrid Methods for Single-Winner Elections," *Voting Matters,* vol. 29, pp. 1-14, 2011.

[66] D. Cary, "Estimating the margin of victory for instant-runoff voting," presented at the Proceedings of the 2011 conference on Electronic voting technology/workshop on trustworthy elections, San Francisco, CA, 2011.

[67] T. R. Magrino, R. L. Rivest, E. Shen, and D. Wagner, "Computing the margin of victory in IRV elections," presented at the Proceedings of the 2011 conference on Electronic voting technology/workshop on trustworthy elections, San Francisco, CA, 2011.

[68] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier, "Analysis of Strategies for Building Group Profiles," in *User Modeling, Adaptation, and Personalization*. vol. 6075, P. De Bra, A. Kobsa, and D. Chin, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 40-51.