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

Tailoring Group Package Recommendations to Large Heterogeneous Groups Based on Multi-Criteria Optimization

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

Technical Report GMU-CS-TR-2015-7

Abstract

A group package recommender framework is proposed to provide recommendations on dynamically defined packages of products and services to large heterogeneous groups based on multi-criteria optimization. The framework is based on: (1) sampling the entire large group; (2) eliciting the utility function for each member; (3) clustering the sample heterogeneous group into a number of relatively small homogeneous subgroups; (4) extracting the representative utility function for each subgroup; (5) estimating the utility function of the entire group, and use it to find an optimal recommendation alternative; (6) diversify recommendations across those subgroups; (7) applying a group decision-making method, to refine the recommendations. A preliminary experimental study is conducted, which shows that the proposed framework is able to produce a small set of ranked recommendations that retains close to optimal precision and recall, as compared to the baseline method applied directly to original large groups.

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 a package of, rather than atomic, recommendations; (2) to deal with multiple, rather than single, criteria associated with recommendations; and, most importantly, (3) to support a possibly large heterogeneous group of diverse users /decision makers who may have different, or even strongly conflicting, views on weights for different criteria. Complex group recommender systems having these features are important in areas such as: budget allocation recommendations, company-wide health care plan selection recommendations, public infrastructure investment recommendations, and travel package recommendations for a large group.

In applications like these, recommendations are given as a package, e.g., a travel package recommendation would include interrelated components of air reservation, accommodation, activities, car rental, etc. Also recommendations associated with multiple criteria, e.g., a travel package would have associated cost, benefit, enjoyment, satisfaction, risk, etc. In many areas, recommendations affect a large number of users/stakeholders. For example, a large number of employees of a company would like to go to a conference, or a large number of people in a county or a city, would like to influence infrastructure investment recommendations and outcomes.

There has been extensive work on recommender systems mostly focused on single-users rather than groups. More recently, researchers have proposed group recommenders in different domains and applications that used different strategies to aggregate individual preferences into a group model [4-11].

However, most of these 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 [12, 13]. However, they do not consider and/or use dynamic preference learning and decision optimization. Work [14, 15] provides package recommendations based on dynamic preference learning and decision optimization. However, they focus on individuals rather than groups.

More recently, it has been proposed in [16, 17] techniques to address the outlined limitations, and provide a diverse set of group package recommendations based on multi-criteria decision optimization. However, their techniques are designed to work for small groups of users, but not flexible enough to support the case where the group can be highly heterogeneous with many possible conflicting views on the weights of different criteria, especially when very large groups are considered, like the case in recommendations for public infrastructure investments. We further detail the related work and research gap in Section 2.

Extending the work of [16, 17] with the ability to support large heterogeneous groups is exactly the focus of this paper. More specifically, the contributions of this paper are two-fold. First, we develop and propose a framework, based on work [16, 17], to support a possibly large heterogeneous group of diverse users who may have different, or even strongly conflicting, views on weights for different criteria. This framework is based on multi-criteria decision optimization and voting. In addition, it works on a very large, or even infinite, recommendation space, which is implicitly defined. The idea of this framework is to randomly select a representative sample from the entire large heterogeneous group, and elicit the utility function of each member in the sample. Then, clustering these utility functions into a number of relatively small homogeneous clusters (subgroups) of users with similar utilities, and using the representative utility function for each cluster to find the optimal recommendation alternative for the subgroup. These subgroup utility functions are then combined using a weighted average to estimate the utility function of the entire group's sample (U). However, using U directly may limit the flexibility of users to refine their choices. Therefore, in the proposed framework, we use the estimated U to come up with a small set of diverse recommendations that are optimal, or near optimal, in terms of the estimated U, yet optimized by the subgroups' utility functions. Finally, the framework uses a group decision-making method to refine the ranking of this small set by each user in the group sample.

Second, we conduct a preliminary experimentation to evaluate the proposed framework by comparing the accuracy of the ranked recommendations using our proposed framework vs. the baseline technique, which apply the system directly on the entire large group as in work [16, 17]. The study shows that the proposed framework is able to produce a small set of ranked recommendations that retains close to optimal precision and recall.

This paper is organized as follows: Section 2 details the related work and its limitations. Section 3 gives a high level description of the proposed framework. Section 4 gives an overview of group decision-making methods that are used in the paper. Section 5 explains the user utility functions' extraction. Section 6 explains the clustering phase. Section 7 explains the group utility estimation. Section 8 presents the optimization and diversity layering. Section 9 discusses the initial experimentation for the purpose of evaluating the framework. Finally, Section 10 concludes the paper and discusses some of our future work.

2. Related Work and Its Limitations

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 include: recommending TV programs and movies [5, 6]; finding songs to play at a shared public space [7]; or finding tourist attraction for a group of tourists [8, 18]. Work [10], is a family-based recipe recommender, which 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 did not address some important 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. By using rank aggregation techniques, work [5] addressed the affect of the group's size on the group recommender system. In addition, work [9, 11] proposed a group consensus function that captured some of the key characteristics of groups, such as social, expertise, and interest dissimilarity among multiple group members.

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

More recently, there has been a host of research that supports packages recommendations [12, 13], however, they do not consider and/or use dynamic preference learning and decision optimization. The CARD Framework [14] and the COD framework [15] 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 [19, 20] 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 [21-26]. 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 that need to be taken into account. Recently, few existing multi-criteria recommender systems have roots in multi-criteria optimization techniques (e.g., [27, 28]; however, these systems focus on atomic (single) products, rather than composite products, and on individual users, rather than groups of users.

Furthermore, few group recommenders applied algorithms to improve group recommendations by creating homogeneous subgroups from the whole group (e.g., [8, 29-31], however, none of them were designed to support package recommendations that are implicitly defined.

The frameworks proposed in [16, 17] support a group package recommender, which based on multi-criteria decision optimization, however, both techniques are designed to work for small groups of users, and they are not flexible enough to support the case where the group can be highly heterogeneous with many possible conflicting views on the weights of different criteria.

3. Overview of The Proposed Framework

In this section, we first describe the recommendation space, then, we explain the recommendation process implemented by the proposed 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: $\forall_i, 1 \leq i \leq n$, $u_i: R \to [0,1]$. 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, $1 \leq i \leq n$, and each domain D_i has a total ordering "better than" denoted $\succcurlyeq_{Di}$. For example, for domain *Saving*, $a_1 \succcurlyeq_{\text{Saving}} a_2 \Leftrightarrow a_1 \geq a_2$.

For a given group of m users, the utility of each user j, denoted by: $\forall_j, 1 \leq j \leq m$, $U_j: [0,1]^n \to [0,1]$, maps a vector of criteria $u_1 \dots, u_n \in [0,1]$ into a user utility $U_j(u_1 \dots, u_n) \in [0,1]$. Similarly the utility of each subgroup z, is denoted by:

$\forall_z, 1 \leq z \leq k$, $U_z: [0,1]^n \to [0,1]$, where k is the total number of subgroups, In addition, the entire group sample utility is denoted by: $U: [0,1]^n \to [0,1]$.

U_z and U define a utility associated with each alternative $a \in R$. Therefore, the subgroup recommendation alternative utility for recommendation a is defined by: $RU_z: R \to [0,1]$, where $RU_z(a) = U_z(u_1(a), \dots, u_n(a))$, and the entire group sample 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 technique is depicted in Fig. 1.

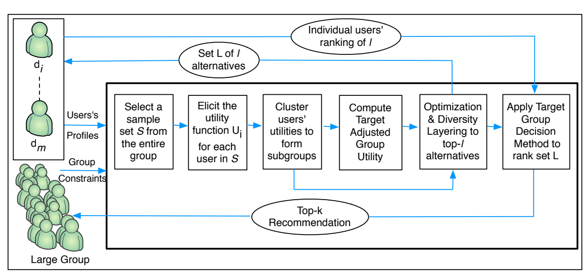

Fig. 1 The Proposed Framework

As shown in the diagram, the process starts when a group of users submits a request to the group recommender. This request specifies the group's decision constraints on recommendation alternatives. To generate top-k recommendations for this large heterogeneous group, the recommender follows seven steps: (1) randomly selecting a representative sample from the entire large heterogeneous group; (2) eliciting the utility function for each member in the sample; (3) clustering these utility functions into a number of relatively small homogeneous clusters (subgroups) of users with similar utilities; (4) estimating the utility function (U) for the group sample by combining the subgroups' representative utility functions; (5) using U to find an optimal recommendation alternative; (6) diversity layering to generate a diverse set of l recommendations which contains the optimal group recommendation; (7) 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 a group decision-making method to make the final recommendations for a group of users. Diverse group decision-making methods are used in practice, which depend on the domain, groups' characteristics, and desirable properties to be satisfied. No single method is considered to be generally superior to all others and fully fair [32]. Work [33], considered six group decision-making methods, any one of which can be used to instantiate their framework. These methods include three of which are based on known and commonly used aggregation strategies, namely, Average, Least Misery, and Average Without Misery strategies; two are existing voting methods based on individuals' ranking, namely, Instant Runoff Voting (IRV) method, and Hybrid Condorcet-IRV method; in addition to a new aggregation strategy called Structurally-Adjusted Average, in which both the influence of decision makers within the group and the dissimilarity of opinions among them, are taken into account.

Borrowing from [33], we also consider the same six group decision-making methods, in the last step of 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 package 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 up with a small set of recommendations that are close to optimal, and sufficiently diverse, so that the group members would have enough flexibility. This explains the second last step. However, to perform optimization and diversification, we need to be able to estimate the entire group utility function that captures the whole group's preferences, this estimation can be derived from the subgroups utility functions, and this explains the third last step in Fig. 1. Note that this group utility function is parameterized based on the final target group decision-making method. We formed the subgroups by clustering a large number of individual heterogeneous utility functions into a number of relatively small homogeneous subgroups of similar utilities. This is done in the third step.

However, the utility functions of the individual users are also not known to the system and need to be extracted from individuals, and this is the second step. Since, in some situation in the real world, it is impractical to elicit the utility function of every member in the entire group, the framework added a sampling phase to support this case by considering only a representative sample of users, whish is done in the first step of the process. We now discuss each of these steps in detail starting with an overview of the six group decision-making methods that mentioned above.

4. Overview of Group Decision Methods

4.1. Average Aggregation Strategy

This strategy is the most straightforward one, which averages the individual utilities for each alternative, and then ranks the one with the highest average first in the group's ranked list. In this strategy, weighted could be assigned to individual preferences based on their influence on the whole group preferences.

4.2. Least Misery Strategy

This strategy is applicable when the group recommender system needs to avoid "misery" for members, which may occur by recommending alternatives that are strongly disliked by any of the group members. It computes the group utility for an alternative as the lowest utility assigned for that alternative by any of the group members.

4.3. 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 utility below a certain threshold are not considered in the group recommendations.

4.4. Structurally-Adjusted Average

This aggregation strategy is developed by work [33]. It computes the group utility by taking 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 (see Section 7).

4.5. Instant Runoff Voting (IRV) Method

IRV method is relatively a strong resistance to strategic manipulation [34], which we believe is a critical feature. It states that: 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 [35]. The method ends with a total order of alternatives from which the recommender system selects the top-k recommendations to the group of decision makers. This total order is a list of eliminated items ordered by which round they are eliminated in.

4.5. Hybrid Condorcet-IRV Method

Applying the IRV method, as described above, on the ranked set of l recommendation to refine the final top-k recommendation may result in a Condorcet winner alternative (which preferred in every one-to-one comparison with the other alternatives) being excluded from the choice set. In order to avoid this issue, a Hybrid Condorcet-IRV method can be applied instead [34]. The method checks if an alternative exists that beats all other alternatives by one-to-one comparison, it will be chosen as the winner, otherwise, the IRV method, described above, will be applied.

After giving an overview of the group decision-making methods that are possible to apply in the last step of the proposed framework, we now explain the first step.

5. Eliciting User Utility Functions

In real world, there are some situations where resources have limitations, which make it impractical to elicit the utility function of every user in the large entire group. For these situations, the framework starts by randomly selecting a representative sample of users to represent the entire group. However, this sampling phase could be skipped for other situations with complete resources.

We start by adopting the COD method [15] for eliciting the utility function of each user. This method starts by viewing a number of distinguishable recommendations in terms of utility vectors to each user. Each recommendation returned stretches the dimension it represents (e.g. Saving), and relaxes on the other dimensions (e.g. Enjoyment, Location, etc.). The process continues iteratively updating the utility vector every time, based on the feedback of the user 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.

The components of a utility vector $\vec{u} = (u_1, u_2, \cdot\cdot\cdot, u_n)$, are associated with criteria which are previously defined, where $\sum_{i=1}^{n} u_i = 1$. The relative importance the user places in each criterion 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 user j. So for each user j, the total utility of a recommendation alternative a w.r.t. the vector $\vec{w}_j$ is defined as:

$$U_j(\vec{u}) = w_{j1}u_1 + w_{j2}u_2 + \cdots + w_{jn} \quad (1)$$

6. Clustering Users' Utilities

Our goal is to cluster the large number of individual heterogeneous utility functions of the group sample into a number of relatively small homogeneous subgroups, where utility functions in a specific subgroup are more similar to each other than to those in other subgroups. Formally, we aim to partition the m utility functions into a set of k clusters $C = \{c_1, c_2, \ldots, c_k\}$ in order to minimize the within-cluster sum of squares, which defined as:

$$\operatorname{argmin}_C \sum_{Z=1}^{k} \sum_{U_j \in C_Z} \left\| U_j - \mu_Z \right\|^2 \quad (2)$$

where, μ_z is the mean of cluster C_Z. For clustering, we choose to use k-means algorithm for its simplicity and popularity [36]. k is defined as: $k = l - 1$, where l is the number of alternatives needed from the Optimization and Diversity layering step. To compute the distance between the input vector and the clustering center, the algorithm uses the Euclidean distance [37].

Recall that it is normal in some situations, where groups are very large, that each individual member may either represents only himself, or represents number of users. For example, in the public infrastructure investments situation, we may have a member who represents the government sector, the private sector, the expertise decision makers, or the environmental protection people.

Therefore, we introduce a representation factor, r_j, $\forall_j, 1 \leq j \leq m$, which is the number of users each member represented. And we take this factor into account when estimating the utility functions for both, subgroups, and the entire group. Consequently, the utility function of a given subgroup C_Z is defined as:

$$U_Z(\vec{u}) = \frac{1}{|C_Z|} \sum_{j=1}^{p} (U_j(\vec{u}) \, . \, r_j) \tag{3}$$

where p is the number of users in subgroup C_Z, and $|C_Z|$ is the normalized size of subgroup C_Z (the total number of users represented by this subgroup), which defined as: $|C_Z| = \sum_{j=1}^{p} r_j$.

7. Estimating The Group Utility Function

This group utility estimation is parameterized based on the final target group decision-making method that is applied in the last step, which 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.

7.1. Average Strategy

For a given subgroup C_Z, the degree of its influence on the whole group preferences (Inf_Z), is computed as:

$$Inf_Z = \frac{|C_Z|}{|G|} \tag{4}$$

where $|C_Z|$ is the normalized size of subgroup C_Z, and $|G|$ is the normalized size of the entire group sample, which defined as: $|G| = \sum_{j=1}^{m} r_j$; and $(\sum_{Z=1}^{k} Inf_Z) = 1$.

Taking (Inf_Z) of each subgroup into account, the group utility is defined as:

$$U(\vec{u}) = \sum_{Z=1}^{k} (U_Z(\vec{u}) \, . \, Inf_z) \tag{5}$$

7.2. Least Misery Strategy

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

$$\mathrm{U}(\vec{\mathrm{u}}) = \min_Z(\mathrm{U}_Z(\vec{\mathrm{u}})) \tag{6}$$

where U_Z is defined in Eq. (3).

7.3. Average Without Misery Strategy

In this strategy, the group utility is computed as in the Average strategy, explained above, but those alternatives with any subgroup utility below a certain threshold are not considered in the group recommendations, more formally:
$U(\vec{u}) = \sum_{Z=1}^{k} (U_Z(\vec{u}) \, . \, Inf_z)$, such that $\forall_Z, 1 \leq Z \leq k$, $min_Z(\mathrm{U}_Z(\vec{\mathrm{u}})) \geq t$.

7.4. Structurally-Adjusted Average Strategy

As suggested by work [9, 11], the overall group utility of an alternative needs to reflect the degree of consensus on its utility value among group members. Suppose that there are two different alternatives $a1$ and $a2$, and both obtain the same weighted average of the subgroups utilities, but the similarity of opinion among the subgroups 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 subgroups over an alternative tends to be more significant the larger the group is.
To describe the dissimilarity of opinion among subgroups over an alternative, we use the *Standard Deviation* i.e.,

$$\sigma(U_1, \dots, U_\mathrm{k}) = \sqrt{\frac{1}{Z-1} \sum_{\mathrm{Z}=1}^{\mathrm{k}} (U_\mathrm{Z} - MU)^2} \tag{7}$$

where MU is the mean of subgroups utilities for an alternative a. Finally, to reflect both the influence of subgroups within the entire group and the dissimilarity of opinion among them, we compute the adjusted group utility as:

$$U = WU \, . \, (1 - \delta) \tag{8}$$

where WU is the weighted average group utility defined in Eq. (5), and δ represents the dissimilarity penalty that defined as:

$$\delta = \alpha \cdot \frac{\sigma}{\sigma_{\max}} \quad (9)$$

where α, $0 \leq \alpha \leq 1$, is a parameter that represents an upper bound for the dissimilarity penalty, (see Fig. 2), and $\sigma_{\max}$ is the maximum possible σ, i.e.,

$$\sigma_{\max} = \max \sigma(U_1, \ldots, U_k) = \frac{1}{2}\sqrt{\frac{k}{k-1}} \quad (10)$$

where $0 \leq U_1, \ldots, U_k \leq 1$, and clearly, $0 \leq \sigma \leq \sigma_{\max}$

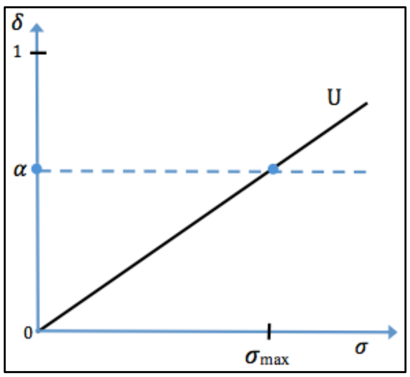

Fig. 2. The Adjusted Group Utility

7.5. Instant Runoff Voting (IRV) Method

First, for each subgroup, the set of alternatives are ranked in descending order by its extracted utility U_Z. Second, the IRV method is applied, as explained in Section 4, to obtain the entire group ranked list of alternatives. Finally, the group utility of each alternative $a \in R$ is estimated as:

$$RU(a) = \frac{n-i}{n-1} \quad (11)$$

where $RU(a) = U(u_1(a), \ldots, u_n(a))$, 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.

7.6. 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, which explained in Section 4.

8. 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 the subgroups' 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 as explained in Section 4. Second, for diversification, we adapted the diversity layering method from CARD [14]. However, the dimensions of the utility space in [14] are the original criteria, whereas, we are advocating of using the space of extracted utilities of the homogeneous subgroups instead.

The key idea is to create a subset of divers recommendations that correspond to different subgroups' 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 $(i\text{-}1)\varepsilon$. 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. 3.

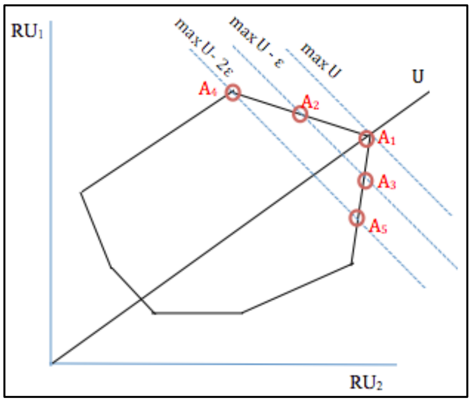

Fig. 3 Diversity Layering

Here, RU_1 and RU_2 are two subgroup's utilities, and U is the entire 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 $U >= \max\{U\} - \varepsilon$, 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 subgroup utility function in turn. However, in order to limit the allowable degradation in the entire group utility of recommendations, the maximum incremental decrease in utility is bounded by ε, which is computed in such a way that gives a total number of recommendations equal to a specific number l.

After generating the diversity set of l recommendations, each individual user is asked to rank (or rate, based on which group decision-making method is used) the set 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 user'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.

9. Initial Experimental Evaluation

Experimental Setting

As explained in Section3, the purpose of the first three steps of the proposed framework is to approximate group utility, because working with all group members individually may not be practical for very large groups.

An important question arises: how much accuracy (in terms of precision and recall) we lose due to approximations in the framework.

We conducted an initial experimentation trying to answer this question. More specifically, in the experimentation, we compared two scenarios: 1) the proposed framework, as explained in Section 3; and 2) the baseline, which is without approximations, namely, the basic framework, which skips the sampling and clustering steps, and elicits the utility for each member in the entire large group, then computes the group utility and uses it to find the optimal recommendation, then creates a subset of divers recommendations corresponding to all individuals' utilities, and finally applying a voting method on the whole large group of individuals to refine the recommendations.

In the experimentation, we set the group size (m) to three different numbers of users: 1000, 10,000, and 100,000, 20 groups of each, and we assumed that each alternative i, $1 \le i \le N$, is associated with only two utilities, u_1 and u_2. For the recommendation space, the number of alternatives N is set to 1000, and we generated these alternatives by assuming that recommendations, in terms of u_1 and u_2, are uniformly split on a quarter circle, as shown in Fig. 4, where $\forall i$, $u_1 = \cos\frac{\pi \,.\, i}{2 \,.\, N}$, and $u_2 = \sin\frac{\pi \,.\, i}{2 \,.\, N}$.

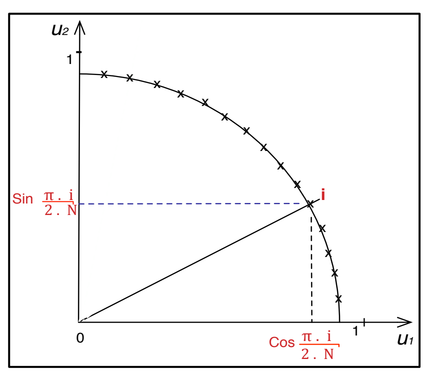

Fig. 4. Alternatives Generation

Note that, the alternatives on the quarter circle are corresponding to the pareto-optimal selection, and, in reality, if there are additional alternatives located inside the circle, they will not be considered because they will be dominated.

In the setting of this study, whereas very large groups of people are considered, it is impractical to consider real user utility functions. Therefore, we assumed that for each user j, $1 \leq j \leq m$, the proposed framework extracted her correct utility function denoted as U_j , and that each user j represented only herself for simplicity. These user utilities were simulated as shown in Fig. 5. We do not have enough space to explain how, exactly, these utility functions are constructed, but essentially, we formulated them so that they are geometrically converged on a quarter circle, see Fig. 5.

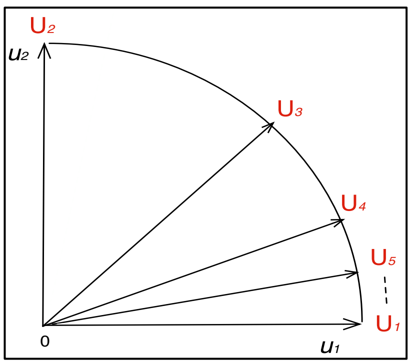

Fig. 5. User utilities Generation

Experimental Methodology

For both systems, the baseline and the proposed framework, we used a particular group decision-making method, namely, the Weighted Average, as explained in Section 4 and 8, to generate the top-5 recommendations for each group. In order to answer the question mentioned above, we calculated the accuracy, in terms of precision and recall metrics, of the top-5 recommendations returned from the proposed framework against the top-5 recommendations returned from the baseline. Precision and recall metrics are widely used on information retrieving scenario, recall is the proportion of relevant recommendations that appear in top recommendations, and the precision is the proportion of recommendations that are relevant recommendations [1]. In this study, a recommendation in the set of the framework's top recommendations (F) is considered as a relevant one if it belongs to the set of the baseline system's top recommendations (B).

For each group, we calculated the estimated recall for the proposed framework at a given rank (k) as:

$$Recall\ (k) = \frac{|F_k \cap B_k|}{|B|} \quad (12)$$

We then computed the average recall at each rank k for the proposed framework by taking the average of recall (k) among all groups. The result is shown in Fig. 6.

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

$$\text{Precision (k)} = \frac{|F_k \cap B_k|}{k} \quad (13)$$

We then computed the average precision at each rank k for the proposed framework by taking the average of precision (k) among all groups. The result is shown in Fig. 6.

Experimental Results

As shown in Fig. 6, at rank 1, almost all of the recommendations returned from the proposed framework in the top-1 recommendations were relevant. This indicates that by applying the proposed framework, we almost did not lose accuracy for top-1 recommendations, even with the approximations. For top-2 recommendations, 86% of the returned recommendations are relevant, however, for top-3, 4, and 5 recommendations, only about 20% of the returned recommendations were irrelevant.

For the statistical analysis, we calculated the Confidence Interval, at level 95%, for the estimated mean of the proposed framework's accuracy, in terms of recall and precision. The results are illustrated in Table 1.

As shown in Table 1, we are 95% confident that we will not lose more than 25% of outcomes' accuracy, in terms of recall and precision, for all top-5 recommendations by applying the proposed framework that approximated the original large groups.

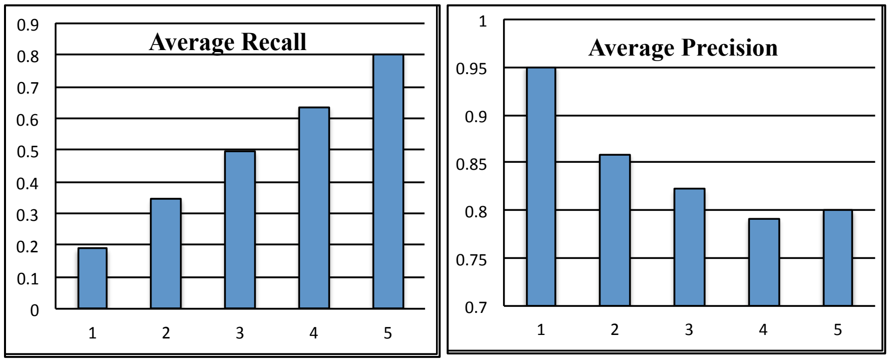

Fig.6 Average Recall and Precision for the Proposed Framework

Table 1. The Confidence Interval for the estimated mean of the proposed framework's accuracy, in terms of recall and precision.

	k = 1	k = 2	k = 3	k = 4	k = 5
Recall	0.19 ±0.0114	0.3467 ±0.023	0.4967 ±0.0292	0.6367 ±0.0386	0.8033 ±0.046
Precision	0.95 ±0.0568	0.8583 ±0.0586	0.8222 ±0.0488	0.7917 ±0.0492	0.8 ±0.0466

10. Conclusions and Future Work

In this paper, we proposed a framework that provides a diverse set of recommendations on packages of products and services to a very large 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 very large group of diverse decision makers who may have different, or even strongly conflicting, views on weights for different criteria.

We also conducted a preliminary experimentation to evaluate the proposed framework's outcomes accuracy in terms of precision and recall against the baseline method. The study shows that our proposed framework, which approximated the original large groups, is able to produce a small set of recommendations that retains near optimal recommendations. More specifically, the experimentation shows that by applying the proposed framework, we did not lose more than 21% of the top-5 recommendation accuracy, even with the approximations.

Although our initial experimentation helps us to learn more about the group decision-making process, the need of extensive validation studies remains.

Many research questions remain open, including: studying the framework's performance using additional group decision-making methods, and demonstrating how this framework applies to a real problem by considering a realistic case study.

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] 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.

[3] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, pp. 331-370, 2002.

[4] 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.

[5] 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.

[6] 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.

[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] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso, "INTRIGUE: Personalized recommendation of tourist attractions for desktop and handset devices," *Applied Artificial Intelligence,* vol. 17, pp. 687-714, 2003.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

[29] M. Komkhao, J. Lu, Z. Li, and W. Halang, "Improving Group Recommendations by Identifying Homogenous Subgroups," in *Knowledge Engineering and Management*. vol. 214, F. Sun, T. Li, and H. Li, Eds., ed: Springer Berlin Heidelberg, 2014, pp. 453-462.

[30] L. Boratto and S. Carta, "Modeling the Preferences of a Group of Users Detected by Clustering: a Group Recommendation Case-Study," presented at the Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), Thessaloniki, Greece, 2014.

[31] E. Ntoutsi, K. Stefanidis, K. Nørvag, and H.-P. Kriegel, "Fast group recommendations by applying user clustering," presented at the Proceedings of the 31st international conference on Conceptual Modeling, Florence, Italy, 2012.

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

[33] H. Mengash and A. Brodsky, "A Group Package Recommender Based on Learning Group Preferences, Multi-Criteria Decision Optimization, and Voting," Technical Report. GMU-CS-TR-2015-6, Dept. CS, George Mason University, VA, USA GMU-CS-TR-2015-6, 2015.

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

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

[36] K.-j. Kim and H. Ahn, "A recommender system using GA K-means clustering in an online shopping market," *Expert Systems with Applications,* vol. 34, pp. 1200-1209, 2// 2008.

[37] X. Wang, Y. Wang, and Y. Zhan, "Optimization of K-means Clustering by Weight Learning," *Journal of Computer Research and Development,* vol. 40, pp. 869-873, 2003.