# GCAR: A Group Composite Alternatives Recommender Based on Multi-Criteria Optimization and Voting

Hanan Mengash
George Mason University
4400 University Drive 4A4
Fairfax, VA 22030
hmengash@gmu.edu

Alexander Brodsky
George Mason University
4400 University Drive 4A4
Fairfax, VA 22030
brodsky@gmu.edu

## Abstract

*This paper proposes a Group Composite Alternatives Recommender (GCAR) framework, which provides recommendations on dynamically defined composite bundles of products and services. 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 the Instant Runoff Voting (IRV) method, from social choice theories, to refine the recommendations. A preliminary experimental study is conducted which shows that the proposed framework significantly outperforms three popular aggregation strategies normally used for group recommendations.*

## 1. Introduction

Recommender systems are increasingly used to help users make effective product and service choices over the Internet. There has been extensive work on recommender systems (e.g. [2, 9, 17]). However, most of this work focuses on atomic (single) products or services, and on individual users. In this paper, we focus 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 [14]. 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.

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

Addressing the above challenges is exactly the focus of this paper. More specifically, the contributions of this paper are two-fold. First, we develop and propose the Group Composite Alternatives Recommender (GCAR) framework based on multi-criteria decision optimization and voting to address the outlined limitations. The GCAR framework works on a very large, or even infinite, recommendation space, which is implicitly defined by a constraint representation of the CARD framework [8]. The idea of the GCAR framework is to elicit individual users' utility functions and to use them to estimate a group utility function. However, using the group utility function directly may limit the flexibility of decision makers to refine their choices. Therefore, in the framework we use the estimated group utility function to come up with a small set of diverse recommendations that are optimal, or near optimal, in terms of the estimated group utility function, yet optimized by individual decision makers' utility functions. Then, the framework uses the Instant Runoff Voting (IRV) method to refine the ranking of this small set by the group.

Second, we conduct a preliminary experimental study to evaluate the proposed framework by

978-1-4799-2504-9/14 $31.00 © 2014 IEEE
DOI 10.1109/HICSS.2014.144

comparing precision and recall of ranked recommendations of our proposed aggregation strategy vs. three well-known state-of-the-art aggregation strategies normally used for group recommender systems, as mentioned in [18]. These strategies are: (1) Average strategy, which takes the average of individuals' rating; (2) Least Misery strategy, which takes the minimum of individual ratings to avoid "misery" for members (intuitively, group's happiness is the happiness of the least happy member since members are all miserable if one of them is unhappy); and (3) Average without "misery" strategy, which takes the average without the minimum rating.

The study showed that the proposed framework's strategy significantly outperforms all of the three popular strategies in terms of recall and precision.

This paper is organized as follows: Section 2 details the related work and research gap. Section 3 gives a high level description of the proposed group recommender framework. Section 4 presents a preliminary experimental study for the purpose of evaluating the framework. Section 5 is the conclusion and avenues for future work.

## 2. Related Work and Research Gap

A number of group recommenders were proposed in the past two decades (e.g. [4, 10, 13, 14, 19-23, 25]), which used different strategies to aggregate individual preferences into a group model [18]. For example, PolyLens [21] 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. MusicFx [19] 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 [4] recommends tourist attractions to groups of users by using the Weighted Average strategy and taking the preferences of relatively homogeneous subgroups, e.g. children, into account. Yu's TV Recommender [25] selects a TV program for a group of users depending on the average of individuals' rating of program features. Travel Decision Forum [14] 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. The Collaborative Advisory Travel System (CATS) [20] 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 [13] 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 [22] 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 [23], is another group recommender application on Facebook that recommends a movie to groups.

However, none of the above group recommender systems were designed for composite product and services, which makes the recommendation space very large, or even infinite, and implicitly, rather than explicitly, defined. In addition, most of them require specific group characteristics rather than provide a general framework for the development of group recommender systems. For instance, the aggregation method in [25] is applicable when the group is quite homogenous, while [21] worked well only for small size groups. Furthermore, the majority of these group recommender systems assume that individual preferences are already known [18]. However, [20] 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. In addition, many group recommender systems are intrusive and require significant feedback from users. For example, Travel Decision Forum [14] and CATS [20] 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.

Furthermore, most of the above group recommender systems aggregate preferences without using the fairness criteria. For instance, in [14] and [25], 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 [10] does not help avoid the fairness issues.

Finally, 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 [1, 7, 24]. 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 [8], 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.

There are several approaches to aggregate individuals' utility functions. Some earlier Multi-Attribute Utility Theory (MAUT) methods of group decision are reviewed by [6], such as the use of weighted algebraic means proposed in [12], and the use of the sample additive theory to aggregate the individuals' utility functions proposed on [5]. The aggregated utility function, however, is only an approximation, and using it directly may limit the flexibility of decision makers to refine their choices.

The CARD Framework [8] supports composite product and service definitions, and provides recommendations based on dynamic preference learning and decision optimization. Composite services in CARD are characterized by a set of sub-services, which, in turn, can be composite or atomic. CARD uses a decision-guidance query language (DGQL) to define recommendation views, which specify multiple utility metrics, as well as the weighted utility function. The COD framework [3] 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.

# 3. GCAR Framework

The recommendation process implemented by the proposed GCAR framework is depicted in Figure 1. 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.

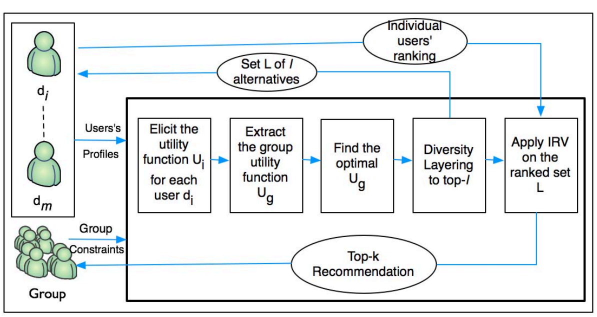


**Figure 1. GCAR framework**

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 the set of $l$ recommendations by each individual in order of her preference; (6) applying the IRV method to refine the final top-k diverse recommendations. We now discuss each of these steps.

## 3.1. Eliciting user utility functions

We start by adopting the COD method [3] for eliciting the utility function of each decision maker. This method, as mentioned in [3], 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.

Recommendation space R, consists of composite products and services; each recommendation is mapped to a utility vector $\vec{u}$, from an $n$ dimensional utility space U, which is presented as $R_+^n$, this mapping is denoted by: $\vec{U} : R \rightarrow R_+^n$. 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 $\succcurlyeq_{D_i}$. For example, for domain Saving, $a_1 \succcurlyeq_{\mathrm{Saving}} a_2 \Leftrightarrow a_1 \geq a_2$.

The relative importance the user places in each dimension is modeled by means of a vector of weights $\vec{w} = (w_1, w_2, \cdots, w_n)$, where $|\vec{w}| = \sqrt{\sum_{i=1}^{n} w_i^2} = 1$, which is called an axis. Each component $w_i$ captures the weight of the $i$-th dimension according to a decision maker $d_j$. So for each decision maker $d_j$, the total utility of a recommendation alternative $a_k$ w.r.t. axis $\vec{w}_j$ is defined as $U_{dj}(\vec{u}) = w_1u_1+w_2u_2+\cdots+w_nu_n$

## 3.2. Estimating the group utility function

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 will be aggregated into the group weights $w_{gi}$ by calculating the algebraic mean of the individual weights:

$w_{gi} = \frac{1}{m}\left(\sum_{j=1}^{m} w_i\right)$, where $j$ = 1, …., $m$, and $m$ is the number of decision makers in the group. The group utility $U_g$ of a recommendation alternative $a_k$ w.r.t. axis $\vec{w}_g$ is defined as: $U_g(\vec{u}) = w_{g1}\, u_1 + w_{g2}\, u_2 + \ldots + w_{gn}\, u_n$. The optimal choice $a_1$ is the one that maximizes the group utility function $U_g$, i.e., $a_1 = \mathrm{argmax}\ U_g(\vec{u}\,(a))$.

### 3.3. 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 are 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 achieve these two competing goals, we adapt the diversity layering method from CARD [8]. However, the dimensions of the utility space in [8] deal with criteria, whereas, in this case, the dimensions of the utility space deal with utilities of the decision makers.

The key idea is to create a subset of diversity recommendations where each of them is based on a different individual's utility function, while preserving a bounded distance from the optimal group utility score 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 $U_g$. 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 Figure 2.

Here, $U_{d1}$ and $U_{d2}$ are two individual decision maker's utilities, and $U_g$ is the group utility, which is defined as a linear combination of $U_{d1}$ and $U_{d2}$. The two-dimensional polyhedron set in the figure depicts all possible utility vectors of recommendations.

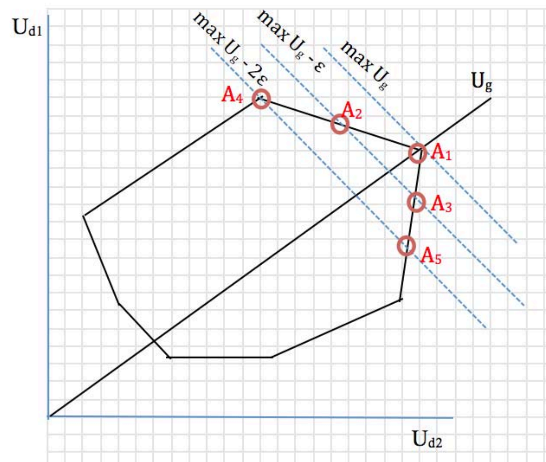


**Figure 2. Diversity Layering**

Among these vectors, $A_1$ is the optimal recommendation that maximizes $U_g$. The second layer includes recommendations for which $U_g >= \max\{U_g\} - \varepsilon$, where ε corresponds to a percentage of $\max\{U_g\}$, say 2%. The selected recommendations in this layer are $A_2$ and $A_3$ because they maximize $U_{d1}$ and $U_{d2}$ 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 $U_g >= \max\{U_g\} - 2\varepsilon$, and the selected recommendations in this layer are $A_4$ and $A_5$ which have the maximum $U_{d1}$ and $U_{d2}$ in turn.

As explained, the diversity layering method generates a set of diverse alternatives by optimizing each user utility function in turn. However, in order to limit the allowable degradation in the 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 $l$.

### 3.4. Ranking the recommendations by individuals

After generating the diversity set of $l$ recommendations by using the estimated group utility function and optimizing each user utility function in turn, 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 the set of $l$ recommendations in a way that truly reflects her preferences. The benefit of allowing each member to rank 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.

### 3.5. Applying the IRV method

Finally, the IRV method is applied on the ranked set of $l$ recommendations to refine the final top-k diverse recommendations.

Instant Runoff Voting (IRV), also referred to as Alternative Vote (AV), 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 [11, 15]. If there is a tie for last place in numbers of votes, special tie-breaking rules are applied to select which alternative to eliminate [15, 16].

IRV 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 order to end with a total order of eliminated alternatives from which the final top-k recommendations are selected and displayed to the group, GCAR framework uses 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 5 decision makers, $d_1$, $d_2$, $d_3$, $d_4$, and $d_5$, who initially ranked the generated diversity set of $l$ recommendations as shown on Table 1. Both alternatives, $A_2$ and $A_4$ have the least first-choice votes, which = 0. Based on rule 1, the alternative to eliminate in the first round is selected randomly. Suppose $A_4$ is eliminated first, shifting everyone's options to fill the gaps (see Table 2).

**Table 1. Initial votes**

| | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ |
|---|---|---|---|---|---|
| 1st choice | $A_1$ | $A_3$ | $A_5$ | $A_1$ | $A_3$ |
| 2nd choice | $A_2$ | $A_1$ | $A_2$ | $A_3$ | $A_1$ |
| 3rd choice | $A_3$ | $A_5$ | $A_1$ | $A_2$ | $A_4$ |
| 4th choice | $A_4$ | $A_2$ | $A_3$ | $A_5$ | $A_5$ |
| 5th choice | $A_5$ | $A_4$ | $A_4$ | $A_4$ | $A_2$ |

Then, in round 2, $A_2$ is eliminated and again everyone's options are shifted to fill the gaps (see Table 3). In round 3, $A_5$ is eliminated since it has the least first-choice votes (see Table 4). Finally, $A_1$ has the majority votes, and wins the election.

According to the previous example, we end with the following total order of alternatives: $A_1 \succ A_3 \succ A_5 \succ A_2 \succ A_4$, from which the recommender system selects the top-k recommendations to the group of decision makers.

**Table 2. Round 1**

| | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ |
|---|---|---|---|---|---|
| 1st choice | $A_1$ | $A_3$ | $A_5$ | $A_1$ | $A_3$ |
| 2nd choice | $A_2$ | $A_1$ | $A_2$ | $A_3$ | $A_1$ |
| 3rd choice | $A_3$ | $A_5$ | $A_1$ | $A_2$ | $A_5$ |
| 4th choice | $A_5$ | $A_2$ | $A_3$ | $A_5$ | $A_2$ |

**Table 3. Round 2**

| | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ |
|---|---|---|---|---|---|
| 1st choice | $A_1$ | $A_3$ | $A_5$ | $A_1$ | $A_3$ |
| 2nd choice | $A_3$ | $A_1$ | $A_1$ | $A_3$ | $A_1$ |
| 3rd choice | $A_5$ | $A_5$ | $A_3$ | $A_5$ | $A_5$ |

**Table 4. Round 3**

| | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ |
|---|---|---|---|---|---|
| 1st choice | $A_1$ | $A_3$ | $A_1$ | $A_1$ | $A_3$ |
| 2nd choice | $A_3$ | $A_1$ | $A_3$ | $A_3$ | $A_1$ |

## 4. Initial Experimental Evaluation

In order to evaluate the proposed GCAR framework, we conducted a preliminary experimental study which involved a total of 32 users, all were graduate students, in 7 groups: 1 group of 3 users; 2 groups of 4 users; 3 groups of 5 users; and 1 group of 6 users. We compared the performance of the proposed strategy with the performance of three aggregation strategies normally used for group recommender systems [18]: (1) the Average rating strategy, (e.g., used in [4, 25]), which takes the average of individual ratings as the rating of the whole group; (2) the Least misery strategy, (e.g., used in [21]), which takes the minimum of individual ratings (i.e., minimum utility value) as the rating of the whole group, to avoid "misery" for members; and (3) the Average without misery strategy, (e.g., used in [19]), which takes the average of individual ratings, after excluding items with minimum ratings, as the rating of the whole group.

The hypothesis of the study was: The proposed system achieves the best recall and precision against the other three aggregation strategies. Precision and recall metrics are widely used on information retrieving scenario [2], 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 [2].

First, we extracted real data about vacation packages from a popular commercial travel website, by submitting a request for a two week vacation in Los Angeles, California starting from June 15, 2013, which included a non-stop round-trip airfare from Washington Dulles Airport. Then, we extracted all the returned packages from this website, keeping only the cost and number of stars (enjoyment) of each package. Second, we adopted COD's method [3], for eliciting the utility function of each individual user in each group. Third, based on each of the three aggregation strategies mentioned above, and in addition to the proposed GCAR strategy, we generated four different sets of the top-4 recommendation packages and presented them in descending order of their utility, as follows:

(1) In the proposed GCAR system, and for each group, we estimated the group utility function as described in Section 3.2, and used the estimated group utility function to find the optimal recommendation. Then, we computed seven diverse recommendations using the diversity layering method described in Section 3.3. Then, we asked each individual user to rank them in order of his/her preferences. Finally, we applied the IRV method to refine the top-4 recommendation packages.

(2) In the Average strategy, and for each group, we computed all package's utility values according to the individuals' utility functions. Then, we took the package's mean utility value as the group utility value, (i.e., the group rating for this package). Finally, we selected the top-4 packages with the highest mean.

(3) In the Least misery strategy, and for each group, we took the minimum package's utility value as the package's group utility value, then we selected the top-4 packages with the highest group utility value.

(4) In the Average without misery strategy, we computed each package's mean utility value after excluding the minimum utility value, and then selected the top-4 packages with the highest mean.

We applied each of the four systems mentioned above on the same data set to generate the top-4 recommendations to each group under each system. Since we wanted to evaluate the quality of the top recommendations resulting from the proposed GCAR system against those resulting from the other three systems, we limited the number of results shown to each group from each system to 4.

Finally, we surveyed all the seven groups, and asked them to rate each of the 16 resulting recommendations from the four systems 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". None of the groups knew which recommendation set resulting from which system.

To calculate the estimated recall at a given rank ($k$), we gathered all the packages rated 4 or 5 by any group in a set called "Good". Then, for each system, we calculated the estimated recall by counting how many of these good recommendations appeared in the top recommendations resulting from the system, as shown in equation (1).

$$Recall\ (k) = \frac{|\{r \in Good | rank(r) \leq k\}|}{|Good|} \quad (1)$$

We then computed the average recall at each rank $k$ for each system by taking the average of recall ($k$) among all the seven groups. The results are shown in Figure 3.

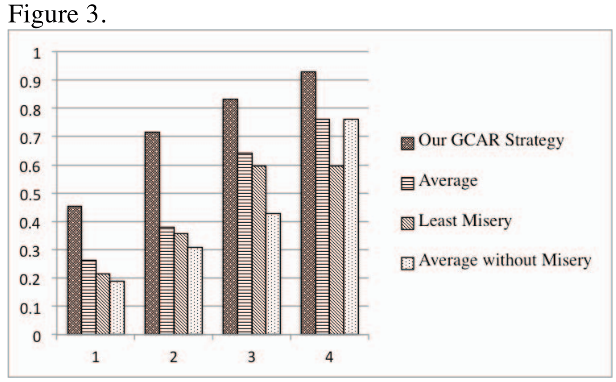


**Figure 3. Average recall vs. rank($k$)**

As shown, at rank 3, our system returned more than 80% of the relevant packages compared to 64% for the Average strategy, 59% for the Least misery strategy, and 42% for the Average without misery strategy. Moreover, our system outperforms all of the three strategies at all ranks in term of recall metric.

To calculate the estimated precision for each system, we counted how many of the returned top-k recommendations were actually in the set "Good", as shown in equation (2).

$$Precision\ (k) = \frac{|\{r \in \text{Good} | \text{rank}(r) \leq k\}|}{k} \quad (2)$$

We then computed the average precision at each rank $k$ for each system by taking the average of precision ($k$) among all the seven groups. The results are shown in Figure 4. As shown, at rank 1, all of the recommendations resulting from our system as the top-1 recommendations were actually relevant, compared to 57% for the Average strategy, and 43% for both the Least misery strategy and the Average without misery strategy. Moreover, our system outperforms all of the three strategies at all ranks in term of precision metric.

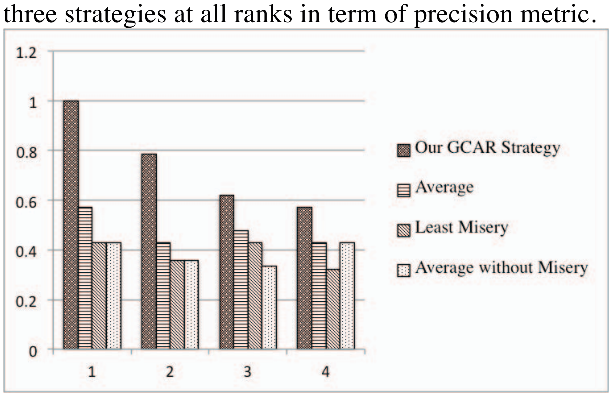


**Figure 4. Average precision vs. rank (*k*)**

In order to determine the statistical significance of our results, we assumed a uniform distribution of ratings over the available packages, which means that each randomly selected package has an equal chance to receive any of the 5 group ratings. Under this assumption, the probability of rating a randomly selected package as "Good" (rating 4 or 5) is 2/5. We obtained 17 selected packages rated "Good" out of 28 trials (4 recommendations for 7 groups) by the proposed system, which can occur by chance with a probability of 2.15%, (i.e., p-value of 0.0215). Therefore, our hypothesis is confirmed with adequate statistical significance.

We conjectured that the proposed GCAR framework outperformed the three other strategies because of: (1) applying diversity layering to generate a diverse set of recommendations; (2) allowing each individual user to rank these diverse recommendations by himself to avoid the effect of an incorrect estimation of the individual user's utility function in the first step; and (3) using the IRV mechanism to aggregate the users preferences in a collective preference that fairly satisfied all of the group members.

## 5. Conclusions

In this paper, we proposed the GCAR framework that provides a diverse set of recommendations on composite bundle of products and services to a group of users. This framework was based on: (1) eliciting the utility function for each individual decision maker; (2) estimating the group utility function; (3) using the group utility function to find an optimal recommendation alternative; (4) constructing a set of diverse recommendations which contains the optimal recommendation alternative; and (5) applying the Instant Runoff Voting (IRV) method, from social choice theories, on the generated set of recommendations to refine the final top-k group recommendations. We also conducted a preliminary experimental study which showed that the proposed framework's strategy significantly outperforms three popular aggregation strategies normally used for group recommendations.

GCAR extended the existing recommender systems in three ways: (1) it considered composite, rather than atomic, recommendations, e.g. a travel recommendation may involve interrelated air reservation, accommodation, activities, car rental, etc.; (2) it dealt with multiple, rather than single, criteria associated with recommendations such as cost, benefit, enjoyment, satisfaction, risk, etc.; and, most importantly, (3) it supported a group of diverse users/decision makers who may have different, or even strongly conflicting, views on weights for different criteria.

Although group recommendations require users to give up some of their privacy in order to improve the recommender's transparency, it was less of a problem in this study since the groups were small and probably consisting of close friends. However, future work is needed to understand how to balance privacy with transparency for larger groups (over 100 users). Furthermore, although GCAR worked well for small groups of users, it is a challenge to scale it to groups of a very large number of decision makers, as well as to establish appropriate aggregation strategies for such groups. In particular, large-scale evaluations and investigations on the affect of group size are needed.

Many research questions remain open, including: (1) studying the system behavior across various group size and similarity; (2) scaling the GCAR framework to groups of a very large number of decision makers; (3) developing efficient algorithms for diversity layering; (4) proposing new aggregation strategies; (5) conducting a large-scale evaluation.

## References


[1] Adomavicius, G., and Kwon, Y., "New Recommendation Techniques for Multicriteria Rating Systems", IEEE Intelligent Systems, 22(3), 2007, pp. 48-55.

[2] Adomavicius, G., and Tuzhilin, A., "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., 17(6), 2005, pp. 734-749.

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

[4] Ardissono, L., Goy, A., Petrone, G., Segnan, M., and Torasso, P., "Tailoring the Recommendation of Tourist Information to Heterogeneous User Groups", Revised Papers from the nternational Workshops OHS-7, SC-3, and AH-3 on Hypermedia: Openness, Structural Awareness, and Adaptivity, 2002, pp. 280-295.

[5] Arora, N., and Allenby, G., "Measuring the Influence of Individual Preference Structures in Group Decision Making", Journal of Marketing Research, 36(4), 1999,

[6] Bose, U., Davey, A.M., and Olson, D.L., "Multi-Attribute Utility Methods in Group Decision Making: Past Applications and Potential for Inclusion in Gdss", Omega, 25(6), 1997, pp. 691--706.

[7] Bradley, K., and Smyth, B., "Improving Recommendation Diversity", in Book Improving Recommendation Diversity, 2001

[8] Brodsky, A., Henshaw, S.M., and Whittle, J., "Card: A Decision-Guidance Framework and Application for Recommending Composite Alternatives", Proceedings of the 2008 ACM conference on Recommender systems, 2008, pp. 171-178.

[9] Burke, R., "Hybrid Recommender Systems: Survey and Experiments", User Modeling and User-Adapted Interaction, 12(4), 2002, pp. 331-370.

[10] Campos, L.M., Fern\, J.M., \#225, Ndez-Luna, Huete, J.F., and Rueda-Morales, M.A., "Managing Uncertainty in Group Recommending Processes", User Modeling and User-Adapted Interaction, 19(3), 2009, pp. 207-242.

[11] Cary, D., "Estimating the Margin of Victory for Instant-Runoff Voting", Proceedings of the 2011 conference on Electronic voting technology/workshop on trustworthy elections, 2011, pp. 3-3.

[12] Csáki, P., Rapcsák, T., Turchányi, P., and Vermes, M., Research and Development for Group Decision Aid in Hungary by Wingdss, a Microsoft Windows Based Group Decision Support System. (Working Paper of the Laboratory of Operations Research and Decision Systems (Lords) Wp 93-9.), MTA SZTAKI, 1993.

[13] Dery, L.N., Kalech, M., Rokach, L., and Shapira, B., "Iterative Voting under Uncertainty for Group Recommender Systems", Proceedings of the fourth ACM conference on Recommender systems, 2010, pp. 265-268.

[14] Jameson, A., "More Than the Sum of Its Members: Challenges for Group Recommender Systems", Proceedings of the working conference on Advanced visual interfaces, 2004, pp. 48-54.

[15] Lippman, and David, Math in Society, CreatSpace Independent Publishing Platform, 2012.

[16] Magrino, T.R., Rivest, R.L., Shen, E., and Wagner, D., "Computing the Margin of Victory in Irv Elections", Proceedings of the 2011 conference on Electronic voting technology/workshop on trustworthy elections, 2011, pp. 4-4.

[17] Manouselis, N., and Costopoulou, C., "Analysis and Classification of Multi-Criteria Recommender Systems", World Wide Web, 10(4), 2007, pp. 415-441.

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

[19] Mccarthy, J.F., and Anagnost, T.D., "Musicfx: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts", Proceedings of the 1998 ACM conference on Computer supported cooperative work, 1998, pp. 363-372.

[20] Mccarthy, K., Mcginty, L., Smyth, B., Salam, M., and #243, "The Needs of the Many: A Case-Based Group Recommender System", Proceedings of the 8th European conference on Advances in Case-Based Reasoning, 2006, pp. 196-210.

[21] O'connor, M., Cosley, D., Konstan, J.A., and Riedl, J., "Polylens: A Recommender System for Groups of Users", Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work, 2001, pp. 199-218.

[22] Popescu, G., and Pu, P., "What's the Best Music You Have?: Designing Music Recommendation for Group Enjoyment in Groupfun", Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts, 2012, pp. 1673-1678.

[23] Quijano-Sanchez, L., Recio-Garcia, J.A., and Diaz-Agudo, B., "Happymovie: A Facebook Application for Recommending Movies to Groups", Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, 2011, pp. 239-244.

[24] Smyth, B., and Mcclave, P., "Similarity Vs. Diversity", Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development, 2001, pp. 347-361.

[25] Yu, Z., Zhou, X., Hao, Y., and Gu, J., "Tv Program Recommendation for Multiple Viewers Based on User Profile Merging", User Modeling and User-Adapted Interaction, 16(1), 2006, pp. 63-82