Under consideration for publication in Knowledge and Information Systems

Recommending Packages with Validity Constraints to Groups of Users

Shuyao Qi[1], Nikos Mamoulis[2], Evaggelia Pitoura[2] and Panayiotis Tsaparas[2]

[1]Department of Computer Science, University of Hong Kong, Pokfulam Road, Hong Kong SAR, China.
Email: qisy@connect.hku.hk;
[2]Department of Computer Science and Engineering, University of Ioannina, Ioannina, Greece.
Email: {nikos,pitoura,tsap}@cs.uoi.gr

Abstract. The success of recommender systems has made them the focus of a massive research effort in both industry and academia. Recent work has generalized recommendations to suggest packages of items to single users, or single items to groups of users. However, to the best of our knowledge, the interesting problem of recommending a *package to a group* of users (P2G) has not been studied to date. This is a problem with several practical applications, such as recommending vacation packages to tourist groups, entertainment packages to groups of friends, or sets of courses to groups of students. In this paper, we formulate the P2G problem, and we propose probabilistic models that capture the preference of a group towards a package, incorporating factors such as user impact and package viability. We also investigate the issue of recommendation *fairness*. This is a novel consideration that arises in our setting, where we require that no user is consistently slighted by the item selection in the package. In addition, we study a special case of the P2G problem, where the recommended items are places and the recommendation should consider the current locations of the users in the group. We present aggregation algorithms for finding the best packages and compare our suggested models with baseline approaches stemming from previous work. The results show that our models find packages of high quality which consider all special requirements of P2G recommendation.

Keywords: Package to group recommendation; Recommender systems; Probabilistic models; Recommendation fairness

1. Introduction

Consider a group of people who would like to dine at a restaurant and then have drinks at a nearby bar. Given the potentially numerous options, the group would favor a rec-

Received 15 March, 2017
Revised 12 May, 2017
Accepted 20 June, 2017

ommendation of a (restaurant, bar) pair, which is consistent with the preferences of its members and does not make a member unhappy with respect to the rest of the group.

Despite the vast amount of work on recommender systems, to the best of our knowledge, this *package-to-group (P2G)* recommendation problem has not been studied before, although there is work on recommending a package of items to a single user (e.g., (Deng et al. 2013, Xie et al. 2014, Interdonato et al. 2013)) and recommending a single item to a group of users (e.g., (Amer-Yahia et al. 2009, Roy et al. 2014)). In addition, there are studies on helping a group of users to select a bundle of items (e.g., (Stettinger 2014, Stettinger et al. 2015)). However, they assume that the users are given a set of items and together they decide the items to select, which is a different problem from P2G recommendation.

Specifically, given a group of users U, the goal of P2G recommendation is to suggest one or more packages of items to U, which are suitable for U's members. This problem has several applications beyond the night-out scenario. For example: (i) A tour operator wants to create a package for a group of tourists, consisting of hotels, restaurants, attractions and activities; (ii) An academic institution that organizes a summer school wants to create a curriculum that meets the interests of a group of students; (iii) A movie channel wants to package together a set of movies to offer to a group of movie-goers, or a large family.

In line with previous work on package recommendation (Xie et al. 2010, Parameswaran et al. 2011), we assume the existence of constraints limiting possible item combinations that can be included in a package. Constraints may be either *hard* or *soft*. Hard constraints should definitely be satisfied by a set of items in order for it to form a valid package. Soft constraints express desirable, but not necessary properties for an item set. In terms of hard constraints, without loss of generality, we focus on the special, but practical case, where the items are divided into categories, and a valid package is formed by selecting one item from each category. For instance, in the night-out example, the group may be interested in a package which includes a restaurant and a bar (i.e., one item from category "restaurants" and one item from category "bars"); the tourist group visit a city and they are interested in visiting a museum, dining at a restaurant, and finally resting at a good hotel; the summer school may consist of courses covering different areas (e.g., Theory, Systems); the movie-goers may want to watch a thriller and a comedy. The soft constraints we consider in the paper are defined based on the relationships between the items in a package. For example, in a venue-package recommendation problem, a set of places far from each other is less likely to be selected by a group, compared to a package of nearby places. In this case, we say that the package is less *viable*. The assumption of constraints is not compulsory and is independent of our proposals, as we show in Sections 3.4 and 7.5. For instance, the category constraint can be easily replaced with selecting a number of items regardless of their categories, by virtually assuming that all items belong to a single large category.

Based on the above, we present two probabilistic models for P2G recommendation: one that first computes the probabilities that the group of users likes individual items, before deriving the probability that the group would select a package of items, and one which first forms item packages that are favored by the individual group members before identifying those that have high likelihood to be selected by the group. Our experimental results show that the first model is superior because it seamlessly takes into consideration all special factors of P2G recommendation (e.g., user impact, package viability). In addition, we design and implement algorithms for the models on a database of individual user ratings on items. The algorithms efficiently combine items into candidate packages for recommendation, while avoiding the exploration of the entire search space with the help of pruning bounds.

A unique and novel characteristic of P2G recommendation is that it raises the issue of *fairness*. User groups may be heterogeneous, consisting of people with potentially dissimilar tastes. Thus, for a package I that is overall good for the group (i.e., the average group member preferences on its items are high), there could be one or more members that do not like any of the items in I; these users would be frustrated if I is selected by the group. In this case, we consider the package to be *unfair*. On the other hand, if each group member finds at least one item in the package that she likes, we consider such a package to be *fair*. We formalize fairness for P2G recommendation, inspired by the corresponding concept in fair division of resources (Steinhaus 1948) and adapt our models accordingly. Note that our fairness definition is very different from that in group decision making (Stettinger 2014), where it is assumed that the group would do item selection for multiple times and to be fair, the unsatisfied users will have higher priority in the next decision. On the contrary, the fairness problem in P2G recommendation is one time and defined on item basis.

This paper is an extended version of (Qi et al. 2016). In addition to the general P2G problem, here, we also investigate a special case of it, where the items for recommendation are related to places (e.g., points of interest, venues of events, etc.) and the users of the group should *travel* to these places; hence, the recommendation should consider the locations of items and the current locations of the users in the target group U. For example, the users in U may currently ve at their workplaces which may be different for each individual group member. If the group U would like to dine at a restaurant and then have drinks at a bar, the recommended (restaurant, bar) packages to U should include places that are not too far, otherwise the package is likely to be rejected. We adapt our P2G recommendation models to comply to this application scenario and test these adaptations experimentally. Moreover, in this paper, we present extended versions of the algorithms that implement P2G and prove their correctness.

The contributions of this paper can be summarized as follows:

- This is the first work that formulates and studies P2G recommendation.
- We propose probabilistic models that incorporate factors such as user impact, package viability, and fairness.
- We consider the application scenario where the the group users have to travel to the recommended items and include the travel distance in the probabilistic models.
- We design efficient P2G recommendation algorithms that scale for large data.
- We evaluate the effectiveness of our models via experiments on two real datasets.

The rest of the paper is organized as follows. We first formally define P2G recommendation in Section 2. Section 3 presents our two probabilistic models and introduces package viability. In Section 4, we define fairness and show how it can be integrated into the models. Section 5 presents the location-based P2G problem and our suggested solutions for it. In Section 6, we propose algorithms that efficiently implement the proposed P2G recommendation models. Section 7 presents our experimental evaluation. Related work is reviewed in Section 8 and the paper concludes in Section 9.

2. Problem Statement

We assume a collection $\mathcal{I}$ of items and a collection $\mathcal{U}$ of users, who express their preferences to items from $\mathcal{I}$ through ratings. A rating $r(u, i)$ of user u for item i may be *explicit*, i.e., u has used and evaluated item i, or *implicit*, i.e., predicted by a classic recommender (e.g., collaborative filtering (Sarwar et al. 2001)).

Given a *group* (set) U of users in $\mathcal{U}$, we consider recommending to U a *package* (set) I of items in $\mathcal{I}$. Recommended packages must be *valid*, i.e., have specific properties. In this paper, we study the case where items belong to categories taken from a set $\mathcal{C}$ (e.g., $\mathcal{C}$ = {restaurant, bar, theater, museum}). Without loss of generality, we assume that each $i \in \mathcal{I}$ belongs to a single category $c_i \in \mathcal{C}$. The group U inputs a query specifying the set of categories $C \subseteq \mathcal{C}$ where the items of the package should be drawn from (e.g., $C = \{$bar, restaurant$\}$). For the ease of discussion, we assume that each item belongs to only one category and a feasible package must contain one item per category (e.g., the users want to visit one bar and one restaurant). More general problem instances will be elaborated in Section 3.4.

Formally, a P2G recommendation task takes as input a group of users $U \subseteq \mathcal{U}$, a set of ratings, and a set of user-specified item categories $C \subseteq \mathcal{C}$, and recommends to U the k most preferable among all feasible packages. We now present generic probabilistic models which define the preference of a group U over a package I.

3. Probabilistic Models

Given a target group U and a query input by U specifying category set C, the objective is to derive the probability distribution $\Pr(I|U, C)$ of the group U to select the package I over C. The probability $\Pr(I|U, C)$ obviously depends on the preference of each user $u \in U$ for the individual items $i \in I$. Given a $u \in U$ and an item i from a specific category $c_i \in C$, the probability of u independently selecting i over other items in c_i can be defined as

$$\Pr(i|u, c_i) = \frac{r(u, i)}{\sum_{i' \in c_i} r(u, i')} \tag{1}$$

Here $r(u, i)$ is u's (explicit or implicit) rating on i. Note that a $\Pr(i|u, c_i)$ is defined for every category c_i. Intuitively, u is more likely to accept a recommendation $i \in c_i$ with higher $r(u, i)$ compared to $r(u, i')$ for other items $i' \in c_i$. Next, we present two models for computing $\Pr(I|U, C)$ based on $\Pr(i|u, c_i)$ and other factors, such as the influence between users in the group, and the likelihood that a set of items are appealing together as a package.

3.1. Group Rating (GR) Model

In the *group rating* (GR) model, we first define the probability that the group U will select an item i. Then we combine the probabilities of individual items, to derive the likelihood of a package.

Item to Group (I2G) Probability

Given group U and a category c_i, the probability of U selecting $i \in c_i$ is $\Pr(i|U, c_i)$. Here $\Pr(i|U, C) = \Pr(i|U, c_i)$, that is, the probability of item i being selected depends only on its own category and not in the full set of categories C. The above probability can be computed based solely on the probabilities of the users in U selecting the item (e.g., see (Gorla et al. 2013)). In our model we adopt the approach in (Liu et al. 2012, Yuan et al. 2014), where it is assumed that different group members may have different impact on the group's decision. In simple words, one or more members of the group, who could be considered as experts on a category, may influence the group in selecting an item in this category. For example, the preference of a group member who is a "foodie" will count more in selecting a specific restaurant.

Following this intuition, we model the group selection as a stochastic process where a user u is first selected as the representative of the group with probability $\Pr(u|U, c_i)$, and then the group selects an item according to u's item distribution. Therefore, we have:

$$\Pr(i|U, c_i) = \sum_{u \in U} \Pr(u|U, c_i) \Pr(i|u, c_i) \tag{2}$$

In this work, we assume that the probability $\Pr(u|U, c_i)$ of a user $u \in U$ is proportional to the activity of the user in category c_i, relative to the other members in the group. This captures the relative *expertise* of the user in the group for this category, which determines her influence in the group. Specifically, let η_{u,c_i} denote the number of explicit ratings user u has given for items in category c_i. We have that

$$\Pr(u|U, c_i) = \frac{\eta_{u,c_i}}{\sum_{u' \in U} \eta_{u',c_i}}$$

Note that Equation (2) is general enough to model different scenarios, depending on the definition of the probability $\Pr(u|U, c_i)$. For example, we can set $\Pr(u|U, c_i)$ to the uniform distribution, where all users influence equally the final selection. Or, we may assume that user influence is independent of the category as $\Pr(u|U)$. In fact, we also considered an approach similar to that in (Yuan et al. 2014), where we used topic-modeling to extract the user-topic and item-topic distributions, and then defined the user influence probability based on the user-item distribution. Experimentally, this approach gave us similar recommendation results on our test data, because the categorization of items is correlated to their underlying topics.

Package to Group (P2G) Probability
To derive the probability $\Pr(I|U, C)$ of the package I to be selected by the group U, for the moment we assume that items are selected independently. Therefore, given $\Pr(i|U, c_i)$, we have:

$$\Pr(I|U, C) = \prod_{i \in I} \Pr(i|U, c_i) \tag{3}$$

3.2. User Package (UP) Model

The GR model assumes that items are selected independently, according to the preferences of a representative user, who is chosen according to her expertise and influence in the item category. The *user package* (UP) model reverses the above generative process. In UP, the group first chooses a representative user u with probability $\Pr(u|U, C)$. The representative user will decide for the *whole package*. We assume that the representative user selects each item independently for now, according to her own preferences. Formally:

$$\begin{aligned} \Pr(I|U, C) &= \sum_{u \in U} \Pr(u|U, C) \Pr(I|u, C) \\ &= \sum_{u \in U} \left\{ \Pr(u|U, C) \prod_{i \in I} \Pr(i|u, c_i) \right\} \end{aligned} \tag{4}$$

Accordingly, once the group has selected a representative u, the selection probability for the package depends only on u, i.e. $\Pr(I|u, U) = \Pr(I|u)$. Also, $\Pr(i|u, C) =$

Table 1. Comparison of GR and UP

		u_1	u_2
X	x_1	1	0
	x_2	0	1
Y	y_1	1	0
	y_2	0	1

(a) $\Pr(i|u, c_i)$

Packages	GR	UP
$I_1(x_1, y_1)$	$1/4$	$1/2$
$I_2(x_1, y_2)$	$1/4$	0
$I_3(x_2, y_1)$	$1/4$	0
$I_4(x_2, y_2)$	$1/4$	$1/2$

(b) $\Pr(I|U, C)$

$\Pr(i|u, c_i)$ similar to GR. We can again adjust the probability $\Pr(u|U, C)$ to model different scenarios. Different from GR, however, UP considers the user impact on packages instead of items. Therefore, $\Pr(u|U, C)$ is defined based on the influence of u on all target categories C collectively.

$$\Pr(u|U, C) = \frac{\sum_{c \in C} \eta_{u,c}}{\sum_{u' \in U} \sum_{c \in C} \eta_{u',c}}$$

where as before $\eta_{u,c}$ is the user u's overall influence on category c.

Essentially, the selection is a two-step process: The package to user (P2U) phase computes the probability $\Pr(I|u, C)$ that user u selects package I, for all users in U; the package to group (P2G) phase computes the overall preference probability $\Pr(I|U, C)$ of the group by taking the combination of the user preferences weighted by the user impact probabilities.

Note that the UP model gives more power to the user selected as representative, since the package selection disregards other group members. As a result, GR and UP may produce very different package selection probabilities, even in the case of uniform user impact probabilities. Consider the example in Table 1(a), where a group of two users $U = \{u_1, u_2\}$ wants to select a package over two categories $C = \{X, Y\}$, each having two items, (x_1, x_2) and (y_1, y_2) respectively, with the preference probabilities shown in the table. Assume that $\Pr(u_1|U) = \Pr(u_2|U) = 1/2$, in all categories. Table 1(b) shows the probabilities of GR and UP for each possible package.

The example shows that, for the UP model, a package that no user likes as a whole (I_2 and I_3) will have very low (zero) probability, while the packages with high probability are actually favored by a single user (I_1 by u_1 and I_4 by u_2). On the other hand, in the GR model, a package becomes acceptable as long as there is at least one user that likes some item in the package (e.g., u_1 likes x_1 and u_2 likes y_2), balancing the preferences of the users better.

Overall, the UP model has the following drawbacks: (1) It assumes each user selects the package as a whole, so that the users' impact on different categories cannot be evaluated; (2) For the same reason, a user will never select a low rated item by her, that is, a user will never compromise for the sake of the group; (3) The top packages for different users may not overlap, especially for dissimilar users, leaving the group dissatisfied as a whole. We therefore expect UP to produce worse packages than GR in practice.

3.3. Package Viability

So far, we have considered only the preferences of the users over individual items, assuming independence between items. However, in real-life, some items are more likely to be selected together than others. For example, a restaurant and a movie theater have higher chances to form a preferable package if they are spatially close. Motivated by this fact, we define the probability $\Pr(V|I)$ that a package I is *viable as a whole*. One possible evaluation of $\Pr(V|I)$ is to consider the pairwise *relevance* between items in I.[1] Here, V denotes a random variable, which is 1 if the package is viable and 0 otherwise. The relevance between two items can be derived by a function on their features (e.g., their spatial distance), or by recorded statistics (e.g., joint probability). Take the case of recommending a package of places as an example. If we regard the relevance between any pair of items to be inversely proportional to their distance (measured by a function $dist(.)$), we can define $\Pr(V|I)$ as:

$$\Pr(V|I) \propto e^{-\max_{i,i' \in I}\{dist(i,i')\}} \tag{5}$$

Intuitively, if the maximum distance between any pair of places in a package is large, the package has low probability to be appealing. There can be other measurements of $\Pr(V|I)$ as well. For example, we can consider the visting order of the items in I, or relate the viability to traveling time cost instead of distance between items; we can also define $\Pr(V|I)$ based on the historical probability where items in I are selected together (Zhu et al. 2014). Our models are independent of the specific definition of $\Pr(V|I)$ and for ease of discussion, we use Equation (5) as an exemplary viability definition in this paper.

Let us now formalize the probability that a group U will select a package I and the package is viable. Assuming that viability depends only on the package, we have:

$$\begin{aligned}\Pr(I, V|U, C) &= \Pr(V|I, U, C)\Pr(I|U, C)\\ &= \Pr(V|I)\Pr(I|U, C)\end{aligned} \tag{6}$$

In the rest of the paper, both the GR and UP models are augmented with package viability according to Equation 6.

3.4. Generality

So far we have assumed that (1) each item i belongs to a single category and (2) only one item is recommended from each category. In a real-life scenario, these assumptions may be too restrictive. Our models can be easily adapted to apply to more general cases.

Firstly, suppose that an item may belong to multiple categories, e.g., a place is regarded both a restaurant and a bar. If the group U accepts a duplicate item serving different purposes, then the models do not require any adaptation; an item may appear in the recommended package multiple times, from different categories. If, on the contrary, U would not accept any duplicate item in a package, the models can still work with a minor adaptation that filters out packages containing duplicate items. For example, given a package I, its viability probability $\Pr(V|I)$ is set to 0 if an item appears more than once in I.

Next, suppose that the group U is looking for multiple items in one category, or simply looking for items without any category constraint. Without loss of generality, we assume that U wants to find n items in category c_i (or C if U sets no category constraints). In this case, we virtually replicate c_i n times and apply the same models on categories set $C' = \{c'_{i1}, ..., c'_{in}\}$. As a result, n items will be selected from category

[1] In general, for a set of n items, the viability can be defined by aggregating their pairwise relevance or by defining an n-ary function. In this paper, for simplicity and due to the application domain of our case studies in the experiments, we follow the first approach.

c_i. However, since it becomes possible to select an item from c_i multiple times, the aforementioned filtering method should be applied to avoid selecting duplicate items. The above strategy also extends to the generic case of recommending arbitrary number (0 to $|c_i|$) of items from multiple categories. In Section 7.5, we show experimentally the performance of our models without category constraint.

4. Fairness in Recommendations

Both GR and UP find the top packages without considering which users are the most or least happy with the items in the packages. For a selected package I, it is possible that a given user $u \in U$ does not like *any* of the items in I, or that u is the least satisfied user in U for *all* items in the package. Therefore, although U as a whole may like package I, the package selection is not *fair* to user u. In a real-life scenario, where the users in the group care for each other's preferences, we should recommend a package which is both attractive and fair to the majority of the group members.

For a user u and a package I, we say that I is fair to u, if there exists at least one item $i \in I$, such that u's rating on i is ranked in the top-$\Delta\%$ of u's ratings on all items. The rationale is that the existence of at least one item in the package for which u has high rating would make the user tolerant to the existence of other items that she may not prefer, considering that there are other members in the group who may like these items. Given the group U and a package I, we denote by $U_f \subseteq U$ the users to whom I is fair. A *fairness* measure $fair(U, I)$ is accordingly defined:

$$fair(U, I) = \frac{|U_f|}{|U|}, \tag{7}$$

meaning that the more users I is fair to, the better I is for U.

Lastly, we define the fairness-aware score of a package as

$$score_{fair}(U, I) = \Pr(I, V|U, C) \cdot fair(U, I), \tag{8}$$

i.e., we look for packages that are both highly preferable and fair. Note that the above equation is applicable to both GR and UP models. It scores a package I based on both its relevance to the group members U (according to GR or UP), and its fairness to U. In the rest of the paper, we denote the GR and UP package selection models augmented with fairness as GR-Fair and UP-Fair, respectively.

Fairness is inspired by the classic fair division problem in Economics (Steinhaus 1948). Fair division splits one or more heterogeneous resources to a number of people who have different preferences to different parts of the resources, such that everybody believes that they have a fair share. Our P2G selection problem is reminiscent to fair division, because every user in the group has different preferences in the items. However, in P2G, the group members share the items in the suggested package, instead of the items being divided.

5. Location-based Recommendation

In this section, we study a special case of the P2G problem, where the items for recommendation correspond to places (e.g., points of interest, venues of events, etc.) and the P2G recommendation should consider the locations of items and the current locations of the users in U. When making recommendations, we consider the fact that the users in

U must travel to the locations of the items in I. For example, let U be a set of users who would like to dine at a restaurant and then have drinks at a bar. Assume that the users are currently at their workplaces which may be different for each $u \in U$. The recommended (restaurant, bar) packages to U should not include items that are too far from the current locations of the users in U, otherwise there is high chance that the package is rejected.

Formally, we assume that each $u \in U$ has a *current location* (we overload symbol u to denote u's location). In addition, we assume that each candidate item i has a location (also denoted by the same symbol). We use $dist()$ to denote the spatial distance between two locations (e.g., $dist(u, i)$ denotes the distance between the locations of user u and item i). We explore and compare two definitions of package viability with respect to the current user locations and the item locations. In the first, we change Equation (5) to model the viability of a package with respect to the locations of U and I as follows:

$$\Pr(V|U, I) \propto e^{-\max_{u \in U, i \in I}\{dist(u,i)\}} \tag{9}$$

The intuition behind Equation (9) is that the probability that a package is appealing to a group of users is inversely proportional to the maximum distance between any user in the group and any item in the package. That is, the dissatisfaction of the group would be low if all items in the package are relatively close to the users of the group.

However, the above definition sets the probability to be relevant only to the maximum distance between users and items, disregarding the remaining distances. We now explore an alternative definition of viability, which considers the fact that the users would first meet at one item (i.e., place) of the package and then visit the remaining ones as a group. This definition decouples the quality of the package as a function of the distances between the items in it with the distance between the users and the best meeting point inside the package. Therefore, Equation (10) extends Equation (5) to also account for the minimum *meeting point distance* for any item in I.

$$\Pr(V|U, I) \propto e^{-\max_{i,i' \in I}\{dist(i,i')\} - \min_{i \in I}\{\max_{u \in U}\{dist(u,i)\}\}} \tag{10}$$

The distance between the group U and an item i, assuming that the group members are going to meet at i is $\max_{u \in U}\{dist(u, i)\}$, because the farthest distance from i to any member in U determines the time to be spent until all members meet. This definition also encapsulates the fairness between group members in choosing i. By choosing the best meeting point from the items in I, i.e., the one that minimizes $\max_{u \in U}\{dist(u, i)\}$, we get a quantity proportional to the probability that U likes I, based on the best meeting point for U in I. $\Pr(V|I)$ (as in Equation (5)) is the probability that U likes I based on only the distances between the items in I (given that the group will travel between these items). By multiplying the two factors, we get the location-based viability $\Pr(V|U, I)$ of package I for the group U of users. Equation (6) then becomes:

$$\Pr(I, V|U, C) = \Pr(V|U, I)\Pr(I|U, C) \tag{11}$$

For the generation of fair P2G recommendations, Equation (11) is used in the scoring Formula 8 for the definition of $\Pr(I, V|U, C)$.

6. Algorithms

Given a group U of users in $\mathcal{U}$, a set of categories C, a database of items $\mathcal{I}$ and the user ratings over the items, our goal is to find the top-k packages that maximize $score(U, I)$

according to Equation (8), following either model GR or UP. An efficient implementation is critical because the number of candidate packages is exponential to the number of categories. We now present efficient branch-and-bound algorithms for ranking packages based on GR and UP. We also discuss how the algorithms can be adapted for the location-based P2G recommendation problem presented in Section 5.

6.1. Algorithms for GR

Recall that GR includes two phases: the I2G phase which finds in each category the probability $\Pr(i|U, c_i)$ of each item being selected, and the P2G phase which combines items into packages. The final scoring function (Equation (8)) considers three factors in the P2G phase, (1) the group preference $\Pr(i|U, c_i)$ (Equation (2)), (2) the package viability $\Pr(V|I)$ (Equation (6)), and (3) fairness (Equation (8)). As a result, combining the best items found in the I2G phase into packages does not necessarily lead to the best packages. We now present two algorithms for GR.

6.1.1. Baseline Algorithm for GR

To implement the GR model, a baseline algorithm (GR-BA) is to firstly calculate the I2G probability for each item relevant to U, then consider each possible package by calculating its P2G probability and finally select the top-k packages. One optimization is that once there are at least k package candidates, a lower bound θ of the current k-th maximum probability can be calculated, so that for any package I, it can be directly pruned unless $Pr(I, V|U, C) \geq \theta$. Based on similar idea, we can further optimize the algorithm by pruning at the item level even before the package is formed. Assume there is a partial package $I_p = \{i_1, i_2, ...\}, i_j \in c_j$. For any package I that can potentially be formed by expanding I_p, the maximum possible probability of I being selected satisfies

$$\overline{Pr(I, V|U, C)} \leq \overline{Pr(I_p, V|U, C)}$$

where

$$\overline{Pr(I_p, V|U, C)} \leq \prod_{i \in I_p} Pr(i|U, c_i)$$

Therefore, if $\overline{Pr(I_p, V|U, C)} < \theta$, any packages that contain I_p can be directly pruned before a complete package is constructed. The pseudo-code of GR-BA is presented in Algorithm 1. However, even with the optimizations, GR-BA does not prioritize differently rated items and various packages, and thus considers most of the possible packages.

6.1.2. An Incremental Algorithm for GR

As an alternative to GR-BA, we propose a 2-level incremental algorithm GR-INC, which prioritizes items and packages with respect to their potential probability of being selected and computes the I2G and P2G phases concurrently. In particular, the I2G phase is implemented as an (incremental) top-k selection query (Fagin et al. 2003), which generates for each category a list of its items in decreasing probability order of being selected by the group U, according to Equation (2). The I2G phase takes as input $|U|$ sorted lists of item ratings, one per user in U; each list includes only the items in one of the input categories c_i. The P2G phase is implemented as an (incremental) top-k

ALGORITHM 1: Baseline Algorithm for GR (GR-BA)

Input : U, C, k
Output: R
min-heap $R \leftarrow \emptyset$, table $P_U \leftarrow \emptyset$, $\theta = \infty$
for *each item* i *rated by* $u \in U$ **do**
    $P_U[i] = Pr(i|U, c_i)$
for *(partial) package* I_p **do**
    calculate $\overline{Pr(I_p, V|U, C)}$ with P_U
    if $\overline{Pr(I_p, V|U, C)} < \theta$ **then**
        skip any package containing I_p
    if I_p *is a full package* **then**
        update R and θ if $\overline{Pr(I_p, V|U, C)} \geq \theta$
    $\theta = k$th largest score in R
return R

join query (Ilyas et al. 2003) where viability is considered in the aggregation score of the joined item combinations. P2G takes as input the items output by the I2G phase on each category and combines them. Algorithm 2 is a pseudo code of GR-INC, using Procedure 3 as a module, which implements the I2G phase.

Procedure 3 takes as inputs the group U, a category c, the number of requested items k, and a reusable max-heap H_c, and returns the next top-k items in c that are most likely to be selected by U. Here H_c is used to store the recommended items ranked by their probability of being selected. The bounds θ_i and T_i (Line 1) are used to terminate the procedure while guaranteeing the next top-k items are found, and ub_u (Line 2) records the upper bound of u selecting an unseen item $i \in c$. In each round (Lines 4-10), GR-INC-I2G accesses from each u the next rated item $i \in c$ in decreasing order (Line 5) and performs random accesses to retrieve the ratings of i by the other users, in order to calculate $Pr(i|U, c)$ (Line 6) and update H_c (Line 7). Because the items are accessed in decreasing order w.r.t. u's ratings, the last seen item i must have larger probability than any unseen ones to be selected by u. Therefore, the procedure updates ub_u to be $Pr(i|u, c)$ (Line 8). After the accesses, GR-INC-I2G updates θ_i (Line 9) and T_i (Line 10). Note that T_i is calculated based on Equation (2) but instead uses ub_u from each user. Lastly, the procedure terminates when $T_i \leq \theta_i$, returning returns the top-k items as results while removing them from H_c. The following lemma shows that the termination condition is correct.

Lemma 6.1. GR-Inc-I2G correctly finds the next top-k items when $T_i \leq \theta_i$.

Proof. An unseen item i better than the current top-k items must satisfy $Pr(i|U, c_i) > \theta_i$. For any unseen item i, based on Equation (2), $Pr(i|U, c_i) = \sum_{u \in U}\{Pr(u|U, c_i)Pr(i|u, c_i)\}$. Because $ub_u \geq Pr(i|u, c_i)$, we derive that $Pr(i|U, c_i) \leq \sum_{u \in U}\{Pr(u|U, c_i) \times ub_u\} = T_i \leq \theta_i$. Therefore, it is impossible to find better items and the lemma holds. □

Algorithm 2 takes as input the group U, the category set C, the number of requested packages k, and returns the top-k package recommendations R for U. The result set R is initialized as a min-heap ranked by the score defined in Equation (8) (Line 1). For each category $c \in C$, it initializes a max-heap H_c to be used by GR-Inc-I2G, a buffer B_c to record the accessed items in c, and an upper bound ub_c of the maximum possible

I2G probability that U selects an unseen item i (Lines 3-5). Similar to Procedure 3, two bounds θ_I and T_I (Line 6) are used for terminating the algorithm. The algorithm then follows a rank-join procedure (Line 7-17). In each round, it selects the next category c to be accessed (Line 8), based on the corner bound (Ilyas et al. 2003) $ub_c \geq ub_{c'}, c' \neq c$. GR-INC then calls Procedure 3 to get the next k group recommendation items L_c (Line 9). For each item $i \in L_c$, i is inserted into the buffer B_c (Line 11) and joined with buffered items in other categories to form packages (Lines 12-15). Afterwards, ub_c is updated as the I2G probability of the last seen item from c, i.e., minimum one in L_c (Line 18), θ_I is updated based on R (Line 15), and T_I is calculated based on Equation (3) using ub_c from each $c \in C$ (Line 17). Finally, the algorithm terminates when $T_I \leq \theta_I$ and returns R as the results. In the following, we show that the termination condition of GR-INC is correct.

Lemma 6.2. GR-INC correctly finds the top-k packages when $T_I \leq \theta_I$.

Proof. For an unseen package I, in order to be better than the current top-k results, it must satisfy $score(U, I) > \theta_I$. For any unseen package I, based on Equation 8, $score(U, I) = Pr(I, V|U, C) \times fair(U, I)$. Based on Equation 6 and Equation 3, $Pr(I, V|U, C) \leq Pr(I|U, C) = \prod_{i\in I} Pr(i|U, c_i)$, thus $score(U, I) \leq \prod_{i\in I} Pr(i|U, c_i) \times fair(U, I)$. Because $ub_c \geq Pr(i|U, c_i)$ and $ub_{fair} \geq fair(U, I)$, we derive that $\Pi_{i\in I} Pr(i|U, c_i) \times fair(U, I) \leq \Pi_{c\in C} ub_c \times ub_{fair} = T_I \leq \theta_I$, so that $score(U, I) \leq \theta_I$. Therefore, it is impossible to find better packages and the lemma holds. □

Figure 1 illustrates an example where there is a group U with three users u_1-u_3 looking for recommendations from categories bar $c_b(b_1, ..., b_5)$ and restaurant $c_r(r_1, ..., r_5)$, assuming that a package $I(b_i, r_j)$ is always viable (i.e., $\Pr(V|I) = 1$).

For simplicity, in Algorithm 2, we assume that the packages are ranked and selected in decreasing order of $\Pr(I, V|U, C)$ (not $score(U, I)$). Its adaptation to a GR-Fair algorithm (i.e., find the top packages considering fairness) is straightforward. GR-Fair ranks the packages by $score_{fair}(U, I)$ based on Equation (8) (Line 13) and sets θ_I as the kth maximum score in R (Line 16). Based on Equation (8), we can use a tigher bound $T_{I,fair} = \Pi_{c\in C} ub_c \cdot ub_{fair}$ (replacing Line 17) where ub_{fair} is the maximum fairness degree of unseen packages. ub_{fair} is initially 1, and is decreased by $1/|U|$ if a user u exhausts all her top-$\Delta\%$ items, as in this case none of the unseen items could be fair to u. Finally, to implement the location-based P2G problem, we define $\Pr(I, V|U, C)$ according to Equation (11), which is based on Equation (9) or (10).

6.2. Algorithms for UP

We can design algorithms for the UP (and UP-Fair) model in a similar manner as for the GR model.

6.2.1. Baseline Algorithm for UP

The baseline UP-BA algorithm, in the first step, finds for each user $u \in U$ the relevant packages $\mathcal{I}_u$ and their probabilities of being selected by u, i.e., $\Pr(I|u, C)$. Next, UP-BA considers all packages from $\cup_{u\in U}\mathcal{I}_u$ and ranks them using Equation 6.

6.2.2. Incremental Algorithm for UP

Similar to GR-INC, the UP-INC algorithm also follows a 2-level procedure prioritizing items and packages w.r.t. their potential probability of being selected by a user or the

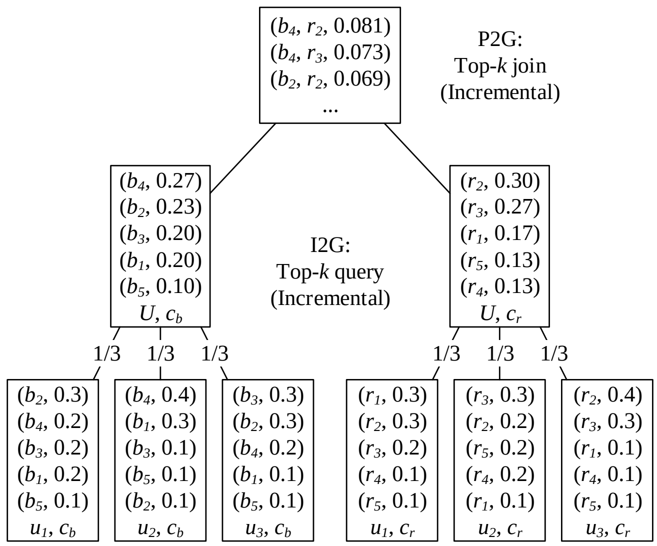


Fig. 1. GR-INC

ALGORITHM 2: Incremental Algorithm for GR (GR-INC)

Input : U, C, k
Output: R

min-heap $R \leftarrow \emptyset$
for *each* $c \in C$ **do**
    initialize a max-heap $H_c \leftarrow \emptyset$
    initialize a buffer $B_c \leftarrow \emptyset$
    $ub_c = \infty$
$\theta_I = -\infty, T_I = \infty$
while $T_I > \theta_I$ **do**
    $c =$ select the next category
    $L_c =$ GR-INC-I2G(U, c, k, H_c)
    for *each item* $i \in L_c$ **do**
        insert i into B_c
        for *each package* I *with* i *and items from* $B_{c'}, c' \neq c$ **do**
            calculate $\Pr(I, V|U, C)$ // Eq. 6
            insert I into R and pop from R if $|R| > k$
    $ub_c = \min_{i \in L_c} \Pr(i|U, c_i)$
    $\theta_I = k$th largest probability in R
    $T_I = \Pi_{c \in C} ub_c$
return R

PROCEDURE 3: GR-INC-I2G

Input : U, c, k, H_c
Output: L_c, H_c

```
θ_i = −∞, T_i = ∞
ub_u = ∞ for each u ∈ U
while T_i > θ_i do
    for each u ∈ U do
        access the next item i ∈ c rated by u
        calculate Pr(i|U, c)                                // Eq. 2
        insert i into H_c
        ub_u = Pr(i|u, c)
    θ_i = kth largest I2G probability in H_c
    T_i = Σ_{u∈U} {Pr(u|U, c) · ub_u}                        // Eq. 2
move the top-k items in H_c to L_c
return L_c
```

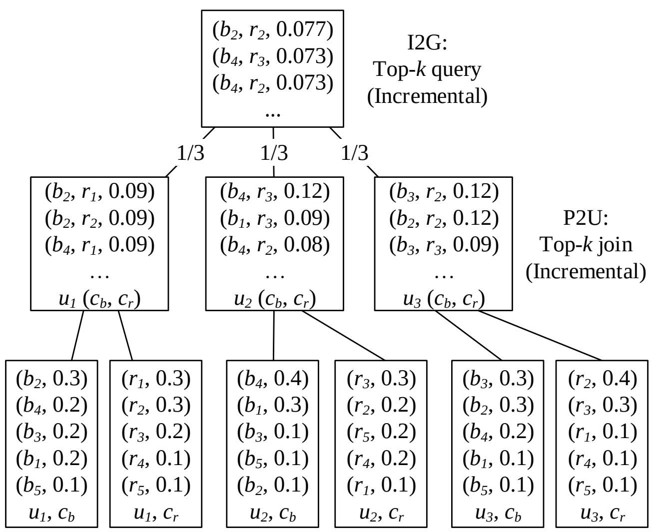


Fig. 2. UP-INC

group. Hence, the P2U phase of UP is implemented as an (incremental) viability aggregated top-k combination query to gradually access packages being liked by each user. On top of that, the P2G phase is implemented as an (incremental) top-k selection query where the packages being liked by the group as a whole are incrementally selected.

Figure 2 presents an example for UP with the same data and setup as in Figure 1. UP-INC performs top-k join at the bottom P2U level. Take the package recommendation to u_1 as an example, the algorithm incrementally accesses u_1's items from c_b and c_r, respectively. It combines the accessed items and first returns the package (b_2, r_1) as the top-1 package for u_1. The upper I2G level operates as a top-k query over the packages retrieved from each user. Note that UP-INC returns a different top-1 package (b_2, r_2) than GR-INC, due to the difference between the two models.

GR-INC and UP-INC, reduce the complexity in practice by only examining only

Table 2. Parameters in experiments (default values in bold)

description	parameter	values
Group size	$\|U\|$	$1, 2, \mathbf{3}, 4, 5$
Number of categories	$\|C\|$	$1, 2, \mathbf{3}, 4, 5$
Fairness threshold (%)	Δ	$1, 5, \mathbf{10}, 50, 100$

a small percentage of the packages for recommendation. However, in the worst case where the package qualities have very small differences, the algorithms examine the majority of packages; hence, the worst-case time complexity of the algorithms is the same as that of an exhaustive algorithm that iterates through all item combinations, i.e., O($n_c^{|C|}$), where n_c is the average number of items per category and $|C|$ is the number of categories in the P2G recommendation problem.

7. Experimental Evaluation

This section evaluates our P2G models and algorithms. Section 7.1 details the setup of our analysis. Section 7.2 studies the effectiveness of the proposed models. Section 7.3 evaluates the effect of considering fairness in the models and presents a user study. Finally, Section 7.4 tests the efficiency of our algorithms.

7.1. Setup

We use two real datasets: Yelp[2] and MovieLens[3] in our evaluation. For Yelp, we use as items venues from the city of Phoenix and consider five categories (restaurants, shopping, beauty & spa, health & medicine, nightlife) with the most venues. Yelp originally contains about 100K users, 17K places and 476K reviews with a numerical rating. Because the number of reviews is small, we employ collaborative filtering (CF) (Sarwar et al. 2001) to get additional review ratings for each user. In particular, we use Mahout[4] to build an item-based CF recommender and retrieve for each user u all item ratings that are not present in the dataset. For the items that are neither explicitly rated by u in the dataset nor recommended by CF, we set zero as u's rating. Finally, we end up having 53M non-zero ratings in total. For MovieLens, we use movies as items from the five most popular genres (drama, comedy, thriller, romance, action), which contain about 138K users, 33K movies and 31M reviews. The same CF recommendation process results in 51M ratings in total. To prevent bias toward any user, in both datasets, we normalize the ratings of every user to $[0, 1]$. All algorithms were implemented in C++ and the tests ran on a machine with Intel Core i7-3770 3.40GHz and 16GB main memory, running Ubuntu Linux.

Table 2 summarizes all parameters involved in our study. On each test, we vary one parameter, while keeping the others to their default values. Each test computes the top-10 recommended packages to a random group U of users. We consider two classes of

[2] http://www.yelp.com/dataset_challenge
[3] http://grouplens.org/datasets/movielens/
[4] http://mahout.apache.org

user groups. Groups in the SIM class consist of users that have similar preferences to items. Each SIM group is generated by randomly selecting a user and then iteratively picking the next user as the one for which the item preference vector has the maximum cosine similarity to the selected users so far. DSIM user groups are generated in the same way, however, using minimum instead of maximum similarity when selecting the next user to add to the group.

7.2. Model Evaluation

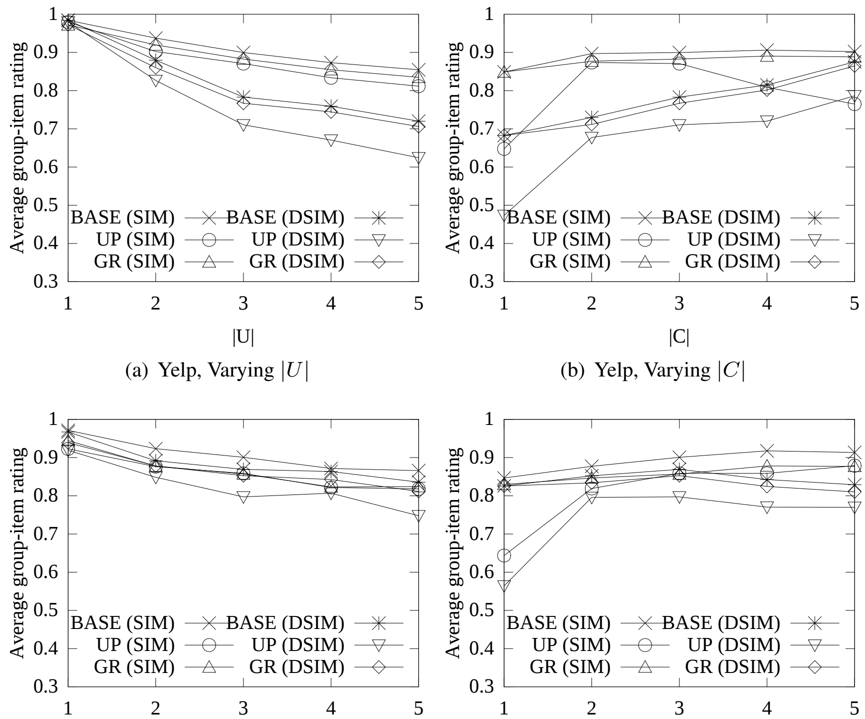
We study the effectiveness of our proposed GR and UP models, by first focusing on the basic models where fairness is not considered. In the evaluation, we include a baseline approach (BASE) which is based on the state-of-the-art group recommendation technique (COM (Yuan et al. 2014)). For each category $c \in C$, COM is used to select the best item for U. These items are then combined to form the top package. The 2nd-best item of each category is then combined with the best items from the other ones to form additional packages and so on until k packages are computed. BASE aims at maximizing the preferences of the group to the individual items in the suggested packages. Note that the original COM model is designed for the scenario where the topics (categories in our case) are not specified by the group of users and thus need to be inferred from group or user-topic distributions. BASE adapts the COM model for our problem by limiting to one topic for recommendation in each category. Still, BASE ignores the possible relationships between items (see Section 3.3); thus, the top items per category selected by BASE do not necessarily form good packages.

We compare BASE, GR, and UP in terms of package quality using two metrics: the average group-item rating $R(U, I)$ and the average item distance $dist(I)$. $R(U, I)$, indicating how much the members of U like the individual items in I, is the average of group rating $\rho(U, i)$ to each item $i \in I$, weighted by the user impacts:

$$R(U, I) = avg\sum_{i \in I} \rho(U, i) = avg\sum_{i \in I} \sum_{u \in U} \Pr(u|U, c_i) r(u, i)$$

The average distance $dist(I) = avg\sum_{i,i' \in I} dist(i, i')$ between the items in the package I indicates how viable it is for them to be chosen together (i.e., items far from each other could be a bad choice). For Yelp, $dist(i, i')$ is the Euclidean distance between the items (venues). For MovieLens, we define $dist(i, i') = 1 - sim(i, i')$, where sim is the similarity between movies i and i', calculated via the Movie-Topic matrix extracted using Latent Dirichlet Allocation (LDA). In this LDA model, we use movie items as *documents* and users who have rated a movie as its *words*. Note that $R(U, I)$ and $dist(I)$ are two indicators of package quality, in terms of group rating on items and package viability, respectively. BASE is expected to generate packages with the best $R(U, I)$, because it is designed to combine items most liked by the groups regardless of the relationship among them. A desirable model should have similar $R(U, I)$ to BASE and at the same time find packages with small $dist(I)$ (i.e., high viability).

Figure 3 shows the average $R(U, I)$ over the packages recommended by BASE, GR and UP, respectively, on Yelp and MovieLens. Since each model recommends a set of top-10 packages, we average $R(U, I)$ (and $dist(I)$) over all these packages. BASE performs the best because of its design goal, however GR finds packages of nearly the same group-item rating. UP, on the other hand, always performs the worst because it only considers user impact at the package level, failing to address cases where different users have different impact on the various categories in C. As expected, for SIM groups,



(a) Yelp, Varying $|U|$

(b) Yelp, Varying $|C|$

(c) MovieLens, Varying $|U|$

(d) MovieLens, Varying $|C|$

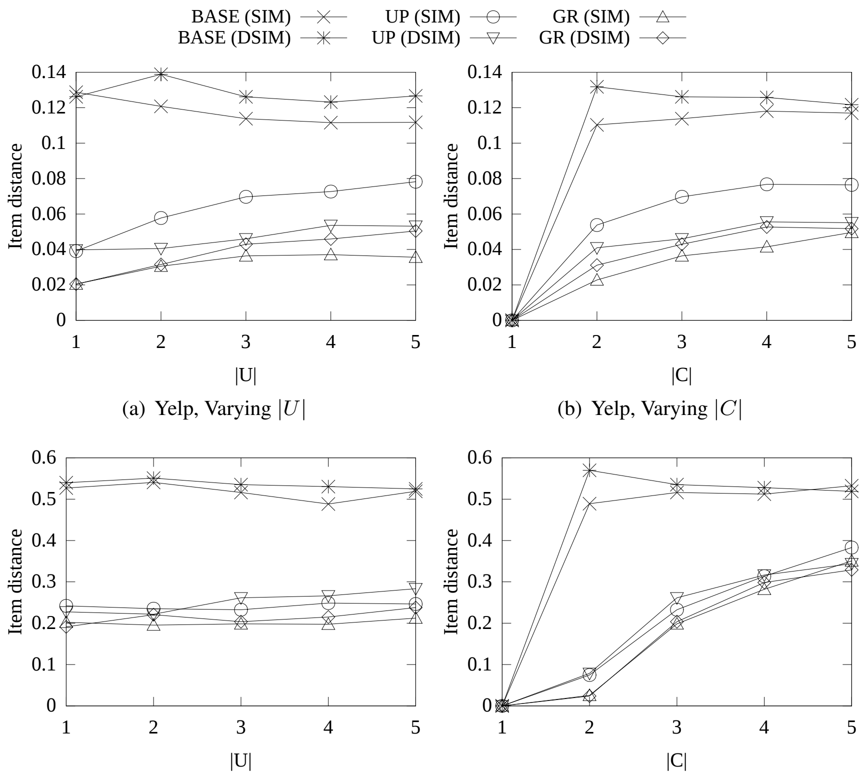
Fig. 3. Group Rating

the models perform similarly, as it is easy to find packages where all items satisfy all group members.

Figure 4 compares the models based on the average distance $dist$ between items. Since BASE ignores relationships between the items, the packages it selects may contain items that are far from each other and have high $dist(I)$ values. UP fails to find packages with items close to each other, which are liked by the group as a whole, but not that much by individual group members (i.e., representatives); hence, its performance w.r.t. $dist(I)$ is worse than that of GR. On MovieLens, $dist(I)$ tends to be larger than on Yelp, because it is harder for two movies to be very similar to each other, compared to finding venues in Yelp that are spatially close. In addition, we observe that the relative performance among the models is the same regardless of the similarity between group members (SIM/DSIM). Overall, GR performs the best considering $dist(I)$, while being only marginally inferior to BASE w.r.t. $R(U, I)$. In the rest of the experiments, we only show results for the more interesting case of DSIM groups.

7.3. Fairness Evaluation

In this section, we compare the basic GR and UP models presented in Section 3 with the variations GR-Fair and UP-Fair that consider fairness (see Eq. 8). Our goal is to understand the tradeoff between quality of recommendation and fairness. Figure 5 shows



(a) Yelp, Varying $|U|$ (b) Yelp, Varying $|C|$

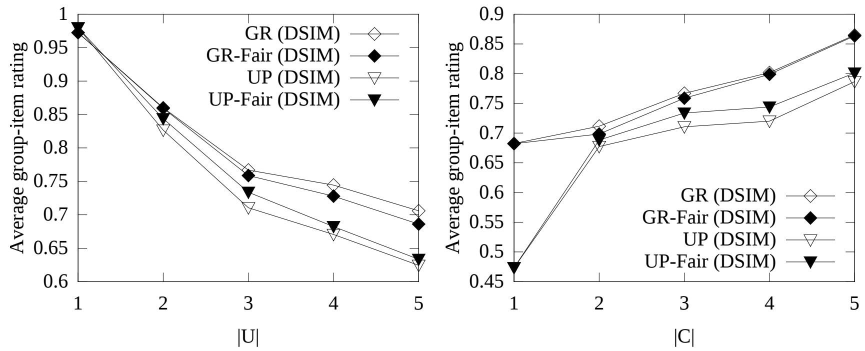
(c) MovieLens, Varying $|U|$ (d) MovieLens, Varying $|C|$

Fig. 4. Item Distance

the package quality in terms of R, for all three versions of GR and UP. Figure 6 shows the average fairness degree of the packages; $fair(U, I)$, defined in Eq. 7, with $\Delta = 10$. In order to consider a metric of fairness independent of the ones optimized in our algorithms, we also compute the mean highest rank of an item $i \in I$ for a user u. Formally, $hrank(U, I) = avg_{u \in U} \min_{i \in I} rank(u, i)$ where $rank(u, i)$ is defined as the rank of i among all items rated by u in category c_i (normalized to $(0, 1]$, the lower the better). Intuitively, if a user u is happy in at least one category, at least one item will have high rank. Figure 7 shows $hrank$ for our algorithms.

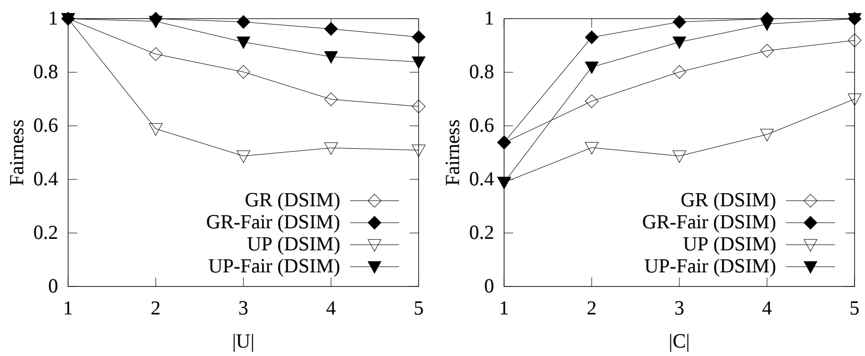
The first observation from these plots is that introducing fairness to GR reduces the quality, as it prevents the model from selecting packages of higher quality which are not fair to some users. Nevertheless, the loss in quality is relatively small. On the other hand, the gains in fairness are significant: GR-Fair improves both $hrank$ (the lower the better) and $fair$. Surprisingly, we observe that the addition of fairness *improves* the quality of UP (i.e., UP-Fair performs better than UP). Note that UP is inherently unfair (it has the worst performance in all fairness metrics – Figures 6 and 7), since it bases the selection on the preferences of a single user. The introduction of fairness counter-balances the drawbacks of UP, and forces the selection process to consider better packages.

Figure 8 evaluates the effect of the fairness threshold Δ, which controls the tradeoff between package quality and fairness. The figure shows the $hrank$ and quality values (based on Eq. 6) against Δ. For small Δ, an item must be ranked very high by u to make



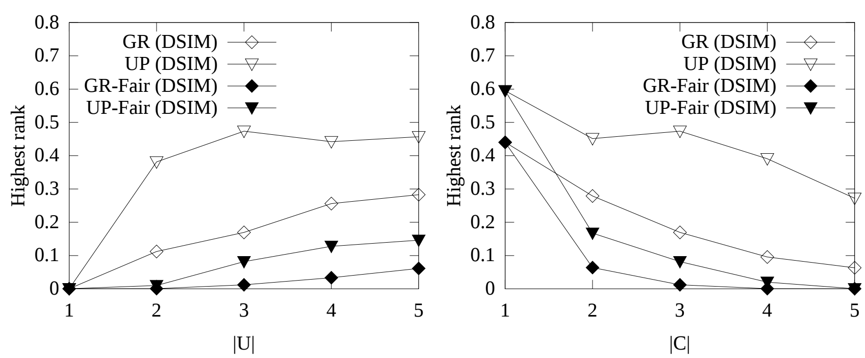
(a) Yelp/DSIM, Varying $|U|$ (b) Yelp/DSIM, Varying $|C|$

Fig. 5. Fair Models: Group Rating



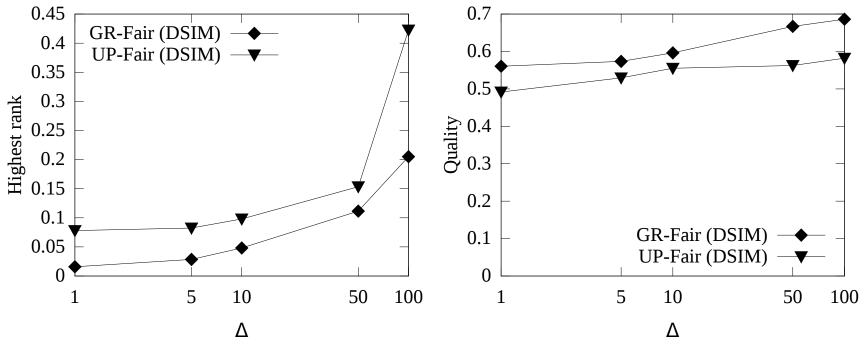
(a) Yelp/DSIM, Varying $|U|$ (b) Yelp/DSIM, Varying $|C|$

Fig. 6. Fair Models: Fairness Degree



(a) Yelp/DSIM, Varying $|U|$ (b) Yelp/DSIM, Varying $|C|$

Fig. 7. Fair Models: Highest Rank



(a) Yelp/DSIM, Highest Rank

(b) Yelp/DSIM, Quality

Fig. 8. GR-Fair Varying Δ

u happy; on the other hand, if Δ is large, fairness becomes looser. As expected, with Δ increasing, quality improves and fairness deteriorates. $\Delta = 10$ gives a good tradeoff between the two.

We repeated the above tests on MovieLens; the results are consistent with those on Yelp. In sum, GR-Fair finds packages such that users are more likely to be happy by at least one item, while not compromising quality compared to GR.

User Study. We also conducted a user study with 30 participants (students) to test the effectiveness of our models and the importance of fairness. First, we asked each participant to rate 70 popular movies belonging to 5 different genres (action, animation, comedy, romance, thriller). The participants were divided into 10 groups of 2-4 users each. For each group, movie packages with 3-4 genres were generated using BASE, GR-Fair, and UP-Fair. We also used a RAND model which selects movies randomly and a least-misery (LM) model that minimizes the maximum compromise a member makes for the group. We asked each group to assess the created packages by providing (1) an overall *rating* (PR) of the package and (2) a characterization of its *fairness* (PF). We did not provide any information on how the packages were generated and presented them to the groups in random order. Figure 9 depicts the average of the PR and PF values (0–1) given by the users. We also report the $R(U, I)$ and $dist(I)$ values of the packages as defined in Section 7.2. In terms of PR, the GR-Fair model outperforms all other models, i.e., it generated the packages that the groups liked the most. This is consistent with the fairness (PF), where GR-Fair also gives the best result, indicating that group satisfaction is correlated with fairness. Lastly, the relative values of $R(U, I)$ and $dist(I)$ are consistent with our experiments on Yelp and MovieLens.

7.4. Efficiency Evaluation

Finally, we evaluate the efficiency of algorithms GR-BA, GR-INC, UP-BA and UP-INC that implement GR and UP models (Section 6). In terms of CPU cost, as Figures 10(a) and 10(b) show, GR-INC outperforms GR-BA by up to an order of magnitude, especially for large values of $|U|$ or $|C|$. As opposed to GR-BA, GR-INC accesses and calculates items/packages in an incremental fashion only when necessary, and stops once the bounding condition is satisfied. Similarly, UP-INC outperforms UP-BA. Note that the UP model is more expensive than GR to compute, because UP prioritizes packages favored by a single user, however, most of these items/packages are not favored

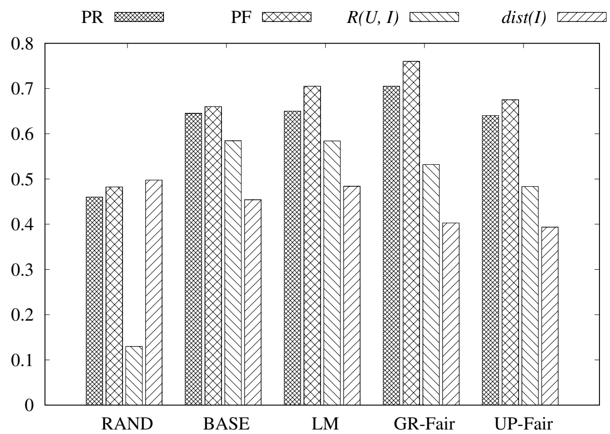


Fig. 9. User Study

by the other users and do not participate in the results. Package recommendations are more costly on MovieLens (Figure 10(c)) compared to Yelp. This is again due to the different item distance distribution between Yelp and MovieLens; on MovieLens, it is more likely that packages have larger distance and thus lower viability, rendering the termination condition during package formation harder to hold. Finally, GR-INC and UP-INC outperform GR-BA and UP-BA, respectively, in terms of accesses to item ratings (Figure 10(d)). Summing up, (1) GR-INC and UP-INC greatly outperform baseline implementations of GR and UP and (2) GR is not only better than UP in terms of quality of suggested packages, but also it is much faster to compute.

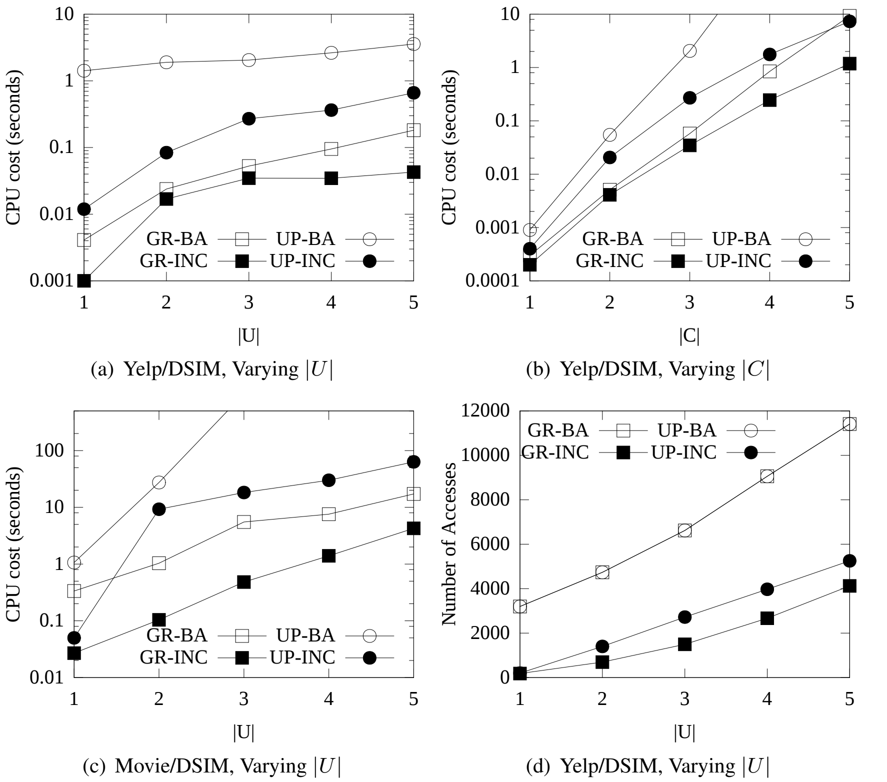
7.5. The Case of No Category Constraints

As discussed in Section 3.4, our definitions and models are also applicable in the more general case, where there are no category constraints. Figures 11–13 show the performance of our models on the Yelp dataset, for the case where the users specify the desired number of items $|I|$ to be drawn from the general pool of items regardless of their categories. Observe that the relative performance of the models is similar to that of the category-constrained case. Specifically, GR-Fair is best at finding packages that are more likely to satisfy each user by at least one item (i.e., being fair) without compromising quality compared to GR.

7.6. Location-based P2G Recommendation

Lastly, we test the effectiveness of our GR-Fair on location-based package to group recommendation. For this purpose, we conducted a series of experiments, in which we assume that the users in the target group U are currently in different locations. Using the Yelp dataset, we set as the default location of each user, the geometric centroid of the items the user has already rated.[5]

[5] Since the user has visited all these restaurants, it is reasonable to assume that his usual location would be the one that minimizes the average distance to all of them.



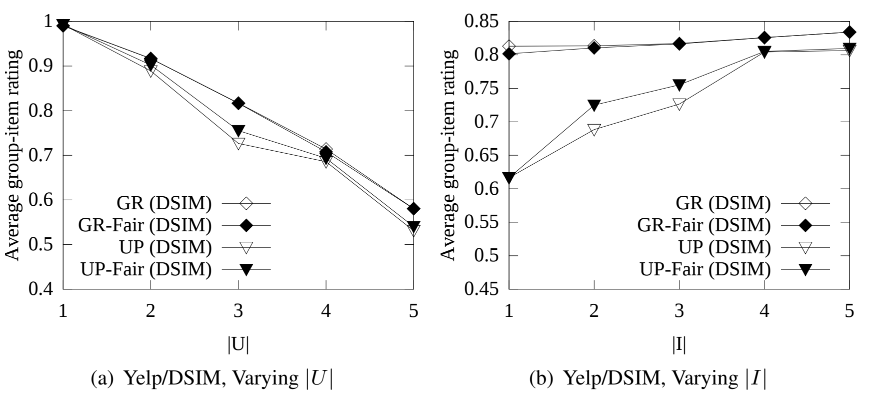
(a) Yelp/DSIM, Varying $|U|$

(b) Yelp/DSIM, Varying $|C|$

(c) Movie/DSIM, Varying $|U|$

(d) Yelp/DSIM, Varying $|U|$

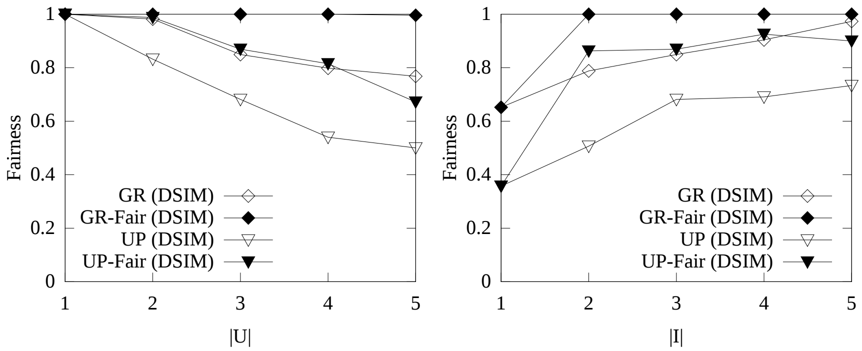
Fig. 10. Cost of Algorithms



(a) Yelp/DSIM, Varying $|U|$

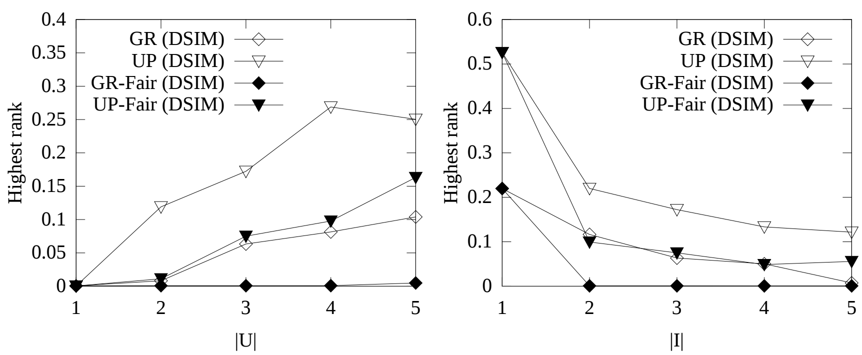
(b) Yelp/DSIM, Varying $|I|$

Fig. 11. Fair Models Without Category Constraints: Group Rating



(a) Yelp/DSIM, Varying $|U|$ (b) Yelp/DSIM, Varying $|I|$

Fig. 12. Fair Models Without Category Constraints: Fairness Degree



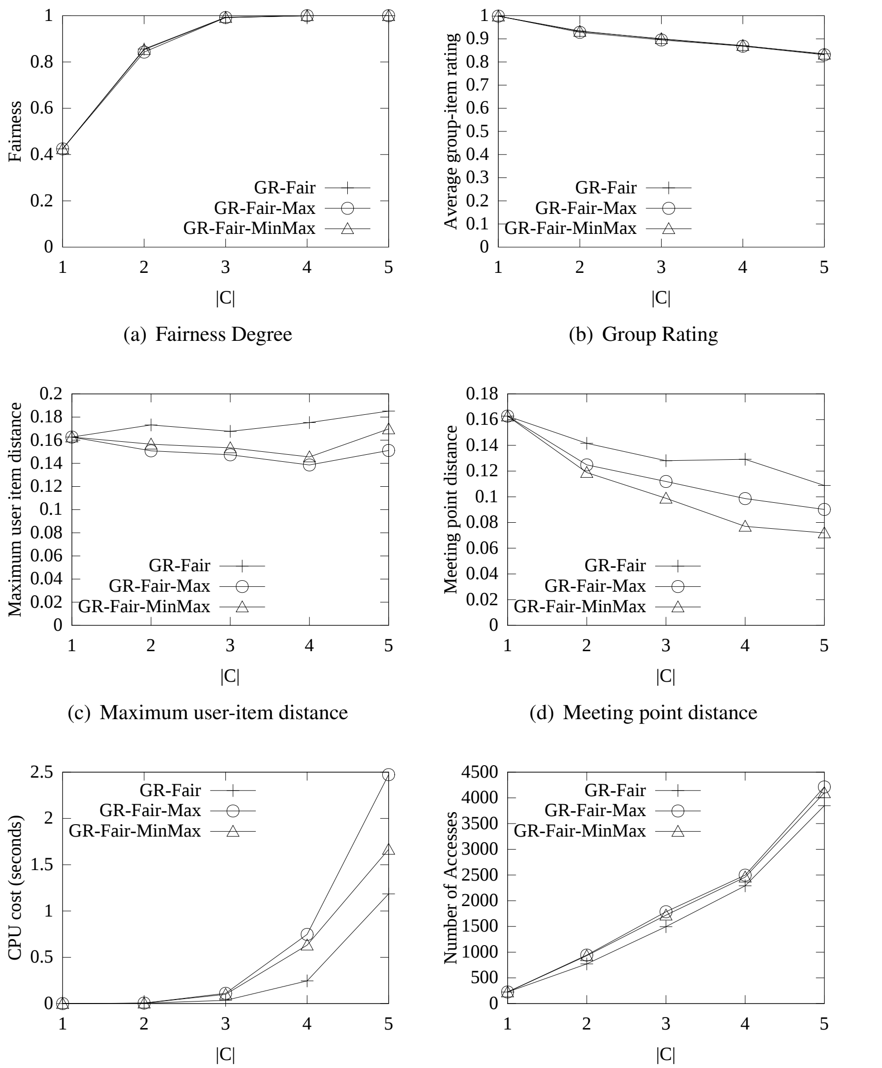
(a) Yelp/DSIM, Varying $|U|$ (b) Yelp/DSIM, Varying $|I|$

Fig. 13. Fair Models Without Category Constraints: Highest Rank

We compare three versions of GR-Fair: the location-agnostic version, which disregards the locations of the users in U and uses Equation 6; GR-Fair-Max, which considers the maximum distance between any user in U and any item in the recommended package as a selection criterion (i.e., Equations 11 and 9); and GR-Fair-MinMax, which considers the best meeting point in the recommended package and package viability (i.e., Equations 11 and 10).

In the first experiment, we measure the different aspects of the three models as a function of the recommended package size. As shown in Figures 14 the general trends agree with those in the previous experiments. The package quality of GR-Fair-Max is slightly inferior to that of the other two models, while the fairness of all three models is identical. Considering the maximum distance from any user to any item in the package, as expected GR-Fair-Max achieves the best performance, although the difference to GR-Fair-MinMax is not significant. As expected, the packages recommended by GR-Fair-MinMax have the lowest meeting point distance, since this model focuses on that factor. In terms of runtime cost, GR-Fair-MinMax is cheaper than GR-Fair-Max because it facilitates the faster termination of GR-INC, however, it is slower than GR-Fair. Recall that GR-Fair does not consider the distance and hence takes better advantage of the initial order of the user-item ratings.

Figure 15 tests the three methods as a function of the group size. In terms of maxi-



(a) Fairness Degree (b) Group Rating

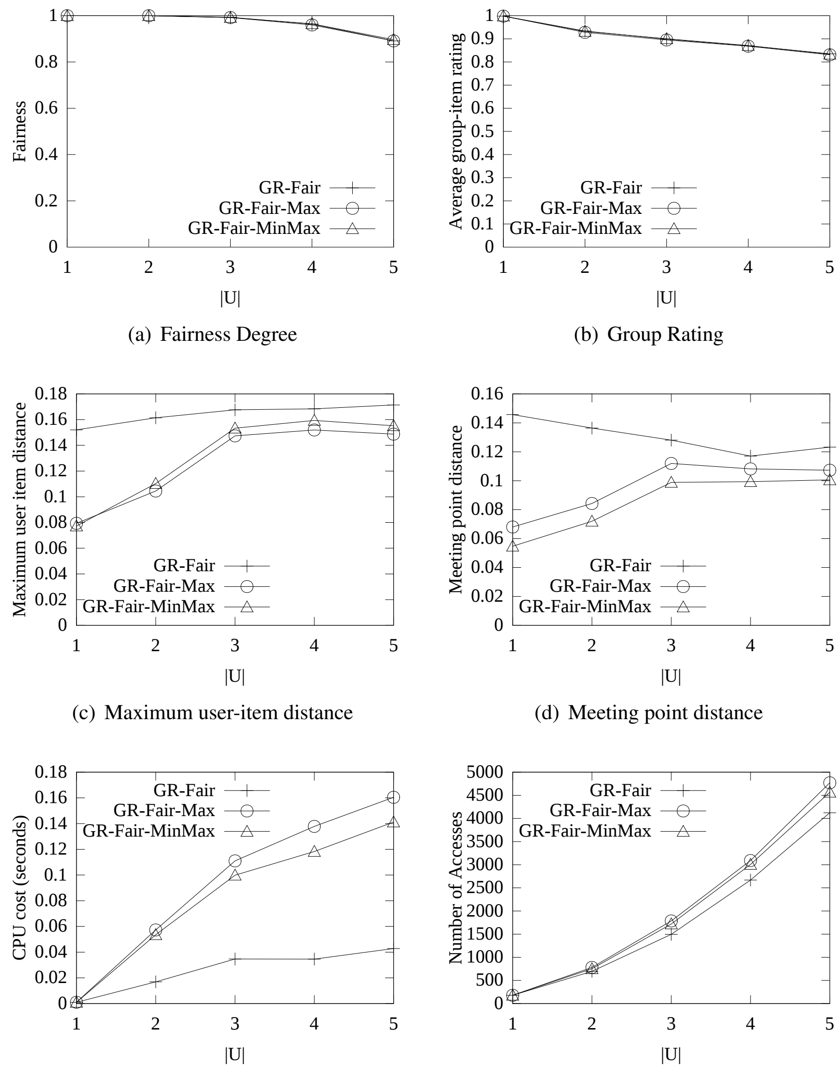
(c) Maximum user-item distance (d) Meeting point distance

(e) CPU Cost (f) I/O Access

Fig. 14. Location-based P2G: varying $|C|$ on Yelp

mum distance and meeting point distance, GR-Fair-Max and GR-Fair-MinMax recommend much better packages than GR-Fair, especially when the group size is small (<3). Again, each model performs best on the measure it optimizes.

In the last experiment, we tested the effect of the distance between the current locations of users in U to the quality of the recommendations. In specific, we computed the geometric centroid c of the original locations of all users and tested different *magnification* factors μ of this distance, by re-locating the users after multiplying their original distance to the centroid c by μ. For example, if $\mu = 1$, the locations of all users are the original ones, whereas if $\mu = 2$ the locations of all users are dispersed so that their



(a) Fairness Degree

(b) Group Rating

(c) Maximum user-item distance

(d) Meeting point distance

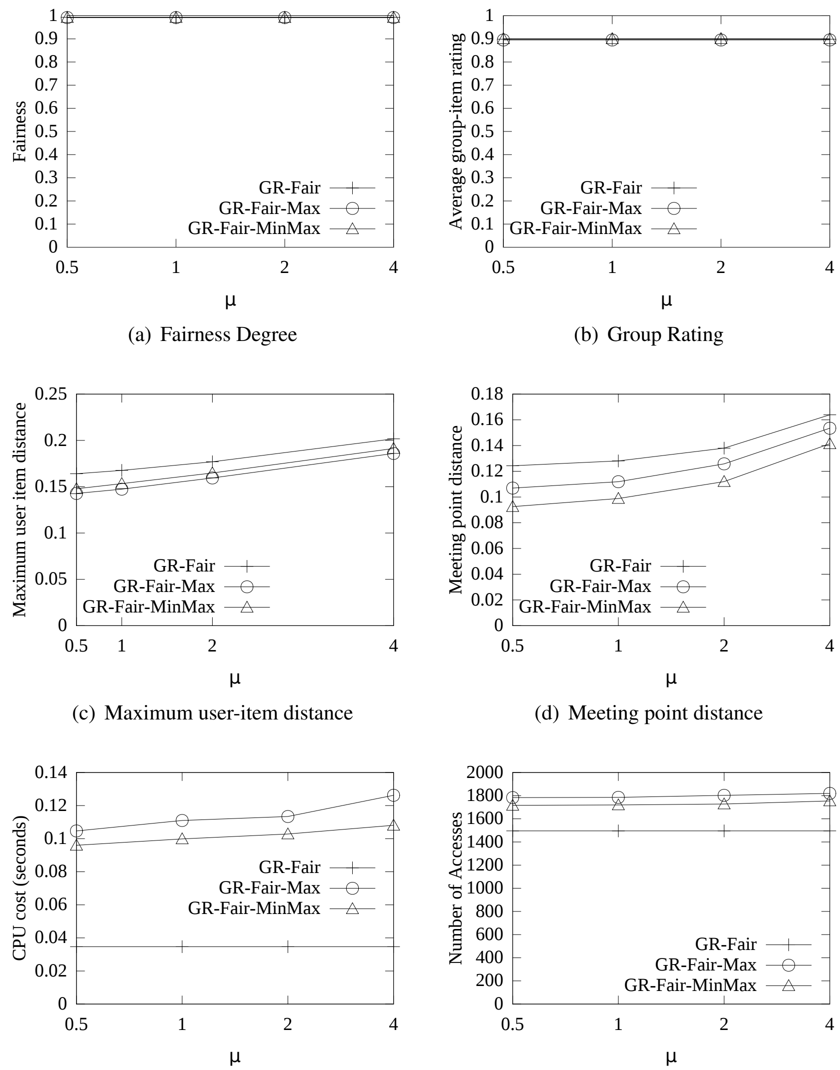
(e) CPU Cost

(f) I/O Access

Fig. 15. Location-based P2G: varying $|U|$ on Yelp

distances to the centroid are doubled. Figure 16 shows the quality of recommendations for different values of μ. Note that the recommendation quality and fairness are not affected by μ. On the other hand, as expected the maximum and meeting point distances increase with μ, but the relative performance of all three methods is not affected.

In summary, our location-based models (GR-Fair-Max and GR-Fair-MinMax) can gracefully be integrated in our P2G recommendation framework. Our experiments show that GR-Fair-MinMax is superior to the simpler GR-Fair-Max model because it achieves similar performance with respect to the max distance measure but significantly better



(a) Fairness Degree (b) Group Rating

(c) Maximum user-item distance (d) Meeting point distance

(e) CPU Cost (f) I/O Accesses

Fig. 16. Location-based P2G: varying $|U|$ on Yelp

performance with respect to the meeting point distance, which makes more sense in a real application. Finally GR-Fair-MinMax is cheaper to compute.

8. Related Work

8.1. Package to User Recommendation

One category of previous work deals with recommending a package of items to a single user. The recommender by (Xie et al. 2010) finds packages of items that collectively maximize the user's interest, but whose total cost does not exceed a given budget. Budget-based package selection, considering diversity and complementarity is studied in (Amer-Yahia et al. 2014). In (Deng et al. 2013), it is shown that selecting a package of items is a hard problem because of the larger search space; strict user-defined constraints can reduce this complexity (e.g., see the work of (Parameswaran et al. 2011)). To avoid searching the whole space, Xie et al. (2014) propose a learning process for predicting the interestingness of packages to users. Interdonato et al. (2013) form packages for different models under item type compatibility and given contextual constraints, and then rank them based on the user's ratings and model/item property preferences. Zhu et al. (2014) study the problem of recommending packages to a user by maximizing the expected reward of the packages. The reward expectation of a package depends on the probability of the user buying all its items together, which can be derived from the transactions history. Package viability (discussed in Section 3.3) is a generalization of the reward defined in (Zhu et al. 2014).

8.2. Item to Group Recommendation

Another line of work deals with the recommendation of single items to a group of users. Some approaches (Jameson & Smyth 2007) combine the ratings of all group members, in order to derive the ratings of a single artificial *representative* user for the group; then, a base recommender is used. Other methods compute recommendations for each group member separately and then aggregate them (O'Connor et al. 2001). For the computation of the combined rating, Amer-Yahia et al. (2009) also consider the agreement between group members. Some recent works (e.g., (Li et al. 2014)) use the social relationships between members to derive group recommendations. Using feedback from users to improve group recommendation has been studied in (Recio-García et al. 2009, Roy et al. 2014). Gorla et al. (2013) define the probability of a group liking an item, based on the item's relevance to each user as an individual. The I2G component of our GR model (see Section 3.1) is an extension of (Gorla et al. 2013) where we also consider the impact of each user on the different categories. Liu et al. (2012) propose a personal impact weighted topic model, where each user has different impact on the group's selection of topics and thus items; i.e., the group selection may be more biased to the preferences of the more influential user.

Yuan et al. (2014) propose an improved consensus model (COM) which differentiates the preference of a user to a topic as an individual or a group member and defines topic-specific user impacts. Our P2G recommendation problem differs from those studied in (Liu et al. 2012) and (Yuan et al. 2014); in our case, the group requests recommendations of items from particular categories. Therefore, we use a probabilistic model with users' item preference in each category, instead of a topic model with users' topic and item distribution, to derive the group preferences. Finally, Yuan et al. (2014) considers content information (e.g. venue distance) to improve group recommendation. However, such information is derived from the user selection history and is used to infer the user's historical preference. This is different from our definition of package viability, which models the potential of a set of items being selected as a package.

In a follow-up work (Serbos et al. 2017) of this paper, we focus on the problem of fairness in P2G recommendation. Specifically, we model the problem of maximizing fairness in package recommendations as an NP-hard coverage problem and propose greedy algorithms that compute approximate solutions within acceptable time. On the other hand, our focus here is the maximization of package quality in combination with the satisfaction of fairness.

8.3. Location Recommendation

Our work is also related to the problem of recommending locations, such as points of interest (POIs), to users. Most of the studies in this direction assume that the users form a location-based social network (like Foursquare) and use their past check-in records as well as contextual information such as the semantics of places and social, location, and temporal information. Some recent works that model and use social influences via friendship links in POI recommendation are (Wang et al. 2013, Ye et al. 2010). Geographical influences have been considered in probabilistic recommendation models in (Liu et al. 2013, Zhang & Chow 2013, Lian et al. 2014, Li et al. 2015). For example, Rank-GeoFM (Li et al. 2015) is a ranking based geographical factorization method for POI recommendation. Yuan et al. (2013) proposed time-aware POI recommendation, where the goal is to recommend activity venues (shopping malls, movie theaters) to a user, by considering the times and locations of activities, via user-based collaborative filtering. Semantics have also been considered in location recommendation models (Gao et al. 2015, Zhang et al. 2015). A recent work (Lu et al. 2017) studies the problem of group recommendation for location data.

Besides user check-in records, GPS data from mobile services have also been used as a basis for location recommendation. For example, GPS trajectories were analyzed in (Zheng et al. 2009), in order to discover POIs as locations where the movements stall for a long time. Zheng et al. (2010) studied the recommendation of activities based on their locations and GPS data from the target users.

9. Conclusion

In this paper, we studied the problem of recommending one or more packages of items to a group of users. We proposed two probabilistic models (GR and UP), both of which incorporate individual ratings by users to items, user impacts, and package viability. In addition, we introduced fairness which is a unique but important feature of the P2G problem. Algorithms were proposed to efficiently implement the two models. Our experiments show that the GR-Fair model finds packages of superior quality in terms of user satisfaction, package viability, and fairness, compared to baseline approaches and UP models. In addition, our algorithms GR-INC and UP-INC clearly outperform baseline implementations. Finally, we showed that GR-Fair can gracefully be adapted for a location-based P2G recommendation problem, where the current locations of the group users and their distances to the suggested items are taken into consideration. In the future, we plan to study additional classes of P2G problems, e.g., when items are selected based on soft/hard budget constraints. We also plan to investigate more issues related to fair P2G recommendation, for example algorithms that find packages of maximum fairness.

Acknowledgments

This work was supported by grant 17205015 from Hong Kong RGC, by European Unions Horizon 2020 research and innovation programme under grant agreement No 657347, and by Marie Curie Reintegration Grant project titled JMUGCS which has received research funding from the European Union.

References

Amer-Yahia, S., Bonchi, F., Castillo, C., Feuerstein, E., Méndez-Díaz, I. & Zabala, P. (2014), 'Composite retrieval of diverse and complementary bundles', *IEEE TKDE* **26**(11), 2662–2675.

Amer-Yahia, S., Roy, S. B., Chawla, A., Das, G. & Yu, C. (2009), 'Group recommendation: Semantics and efficiency', *PVLDB* **2**(1), 754–765.

Deng, T., Fan, W. & Geerts, F. (2013), 'On the complexity of package recommendation problems', *SIAM J. Comput.* **42**(5), 1940–1986.

Fagin, R., Lotem, A. & Naor, M. (2003), 'Optimal aggregation algorithms for middleware', *J. Comput. Syst. Sci.* **66**(4), 614–656.

Gao, H. G., Tang, J., Hu, X. & Liu, H. (2015), Content-aware point of interest recommendation on location-based social networks, *in* 'AAAI', pp. 1721–1727.

Gorla, J., Lathia, N., Robertson, S. & Wang, J. (2013), Probabilistic group recommendation via information matching, *in* 'WWW', pp. 495–504.

Ilyas, I. F., Aref, W. G. & Elmagarmid, A. K. (2003), Supporting top-k join queries in relational databases, *in* 'VLDB', pp. 754–765.

Interdonato, R., Romeo, S., Tagarelli, A. & Karypis, G. (2013), A versatile graph-based approach to package recommendation, *in* 'ICTAI', pp. 857–864.

Jameson, A. & Smyth, B. (2007), Recommendation to groups, *in* 'The Adaptive Web, Methods and Strategies of Web Personalization', pp. 596–627.

Li, K., Lu, W., Bhagat, S., Lakshmanan, L. V. S. & Yu, C. (2014), On social event organization, *in* 'KDD', pp. 1206–1215.

Li, X., Cong, G., Li, X.-L., Pham, T.-A. N. & Krishnaswamy, S. (2015), Rank-geofm: A ranking based geographical factorization method for point of interest recommendation, *in* 'SIGIR', pp. 433–442.

Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E. & Rui, Y. (2014), GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation, *in* 'KDD', pp. 831–840.

Liu, B., Fu, Y., Yao, Z. & Xiong, H. (2013), Learning geographical preferences for point-of-interest recommendation, *in* 'KDD', pp. 1043–1051.

Liu, X., Tian, Y., Ye, M. & Lee, W.-C. (2012), Exploring personal impact for group recommendation, *in* 'CIKM', pp. 674–683.

Lu, Z., Li, H., Mamoulis, N. & Cheung, D. W. (2017), Hbgg: a hierarchical bayesian geographical model for group recommendation, *in* 'SDM'.

O'Connor, M., Cosley, D., Konstan, J. A. & Riedl, J. (2001), Polylens: A recommender system for groups of user, *in* 'ECSCW', pp. 199–218.

Parameswaran, A. G., Venetis, P. & Garcia-Molina, H. (2011), 'Recommendation systems with complex constraints: A course recommendation perspective', *ACM TOIS* **29**(4), 20.

Qi, S., Mamoulis, N., Pitoura, E. & Tsaparas, P. (2016), Recommending packages to groups, *in* 'ICDM', pp. 449–458.

Recio-García, J. A., Jiménez-Díaz, G., Sánchez-Ruiz-Granados, A. A. & Díaz-Agudo, B. (2009), Personality aware recommendations to groups, *in* 'RecSys', pp. 325–328.

Roy, S. B., Thirumuruganathan, S., Amer-Yahia, S., Das, G. & Yu, C. (2014), Exploiting group recommendation functions for flexible preferences, *in* 'ICDE', pp. 412–423.

Sarwar, B. M., Karypis, G., Konstan, J. A. & Riedl, J. (2001), Item-based collaborative filtering recommendation algorithms, *in* 'WWW', pp. 285–295.

Serbos, D., Qi, S., Mamoulis, N., Pitoura, E. & Tsaparas, P. (2017), Fairness in package-to-group recommendations, *in* 'WWW', pp. 371–379.

Steinhaus, H. (1948), 'The problem of fair division', *Econometrica* **16**(1), 101–104.

Stettinger, M. (2014), Choicla: Towards domain-independent decision support for groups of users, *in* 'RecSys', pp. 425–428.

Stettinger, M., Felfernig, A., Leitner, G. & Reiterer, S. (2015), Counteracting anchoring effects in group decision making, *in* 'UMAP', pp. 118–130.

Wang, H., Terrovitis, M. & Mamoulis, N. (2013), Location recommendation in location-based social networks using user check-in data, *in* 'Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems', SIGSPATIAL GIS, pp. 364–373.

Xie, M., Lakshmanan, L. V. S. & Wood, P. T. (2010), Breaking out of the box of recommendations: from items to packages, *in* 'RecSys', pp. 151–158.

Xie, M., Lakshmanan, L. V. S. & Wood, P. T. (2014), 'Generating top-k packages via preference elicitation', *PVLDB* **7**(14), 1941–1952.

Ye, M., Yin, P. & Lee, W.-C. (2010), Location recommendation for location-based social networks, *in* 'SIGSPATIAL GIS', pp. 458–461.

Yuan, Q., Cong, G. & Lin, C.-Y. (2014), Com: A generative model for group recommendation, *in* 'KDD', pp. 163–172.

Yuan, Q., Cong, G., Ma, Z., Sun, A. & Thalmann, N. M. (2013), Time-aware point-of-interest recommendation, *in* 'SIGIR', pp. 363–372.

Zhang, J.-D. & Chow, C.-Y. (2013), iGSLR: Personalized geo-social location recommendation - a kernel density estimation approach, *in* 'SIGSPATIAL GIS', pp. 324–333.

Zhang, J.-D., Chow, C.-Y. & Zheng, Y. (2015), Orec: An opinion-based point-of-interest recommendation framework, *in* 'CIKM', pp. 1641–1650.

Zheng, V. W., Zheng, Y., Xie, X. & Yang, Q. (2010), Collaborative location and activity recommendations with GPS history data, *in* 'WWW', pp. 1029–1038.

Zheng, Y., Zhang, L., Xie, X. & Ma, W.-Y. (2009), Mining interesting locations and travel sequences from GPS trajectories, *in* 'WWW', pp. 791–800.

Zhu, T., Harrington, P., Li, J. & Tang, L. (2014), Bundle recommendation in ecommerce, *in* 'SIGIR', pp. 657–666.

Author Biographies

Shuyao Qi received his PhD degree from the Department of Computer Science, University of Hong Kong, in 2016, and his Bachelor's Degree of Engineering from the Department of Computer Science and Technology in Zhejiang University, Hangzhou, China, in 2012. His research interests include data management, query processing and information retrieval over complex data types.

Nikos Mamoulis received a diploma in Computer Engineering and Informatics in 1995 from the University of Patras, Greece, and a PhD in Computer Science in 2000 from the Hong Kong University of Science and Technology. He is currently an associate professor at the Department of Computer Science and Engineering, University of Ioannina. His research focuses on management and mining of complex data types, including spatial, spatio-temporal, object-relational, multimedia, text and semi-structured data. He has served on the program committee of over 100 international conferences on data management and mining. He is an associate editor for KAIS and the VLDB Journal.

Evaggelia Pitoura is a Professor at the Computer Science and Engineering Department of the University of Ioannina, Greece. She holds a PhD degree from Purdue University, USA. Her research interests are in the area of data management systems with a recent focus on social networks and evolving graphs.

Panayiotis Tsaparas received his PhD degree from the University of Toronto in 2004. Since then he has worked at University of Rome "La Sapienza", at University of Helsinki, and at Microsoft Research. He joined University of Ioannina in 2011, where he is currently an Associate Professor. His research interests include Data Mining, Information Retrieval, and Social Network Analysis.