

# Finding the Right Set of Users: Generalized Constraints for Group Recommendations

Kostas Stefanidis  
Dept. of Computer and Information Science  
Norwegian University of Science and  
Technology, Norway  
kstef@idi.ntnu.no

Evaggelia Pitoura  
Dept. of Computer Science  
University of Ioannina, Greece  
pitoura@cs.uoi.gr

## ABSTRACT

Recently, group recommendations have attracted considerable attention. Rather than recommending items to individual users, group recommenders recommend items to groups of users. In this position paper, we introduce the problem of forming an appropriate group of users to recommend an item when constraints apply to the members of the group. We present a formal model of the problem and an algorithm for its solution. Finally, we identify several directions for future work.

## 1. INTRODUCTION

Recommendation systems aim at suggesting to users items of potential interest to them. In general, this is achieved by estimating a rating for each item and user and then recommending to the user the item with the highest rating [1]. In the content-based approach (e.g., [11, 8]), the estimation of the rating of an item is based on the ratings that the user has assigned to similar items, whereas in collaborative filtering systems (e.g., [7, 4]), this rating is predicted using previous ratings of the item by similar users. Knowledge-based approaches enhance recommendations by exploiting domain knowledge [5]. Most previous work focuses on recommending individual items to individual users. However, recently, group recommendations have received considerable attention. Instead of recommending items to individual users, group recommenders make recommendations to groups of users (e.g., [13, 2, 9, 6, 3]).

An effective group recommendation system must take into account not only the preferences of individual users but also the group dynamics, that is, how groups of people make decisions. Since an item must be acceptable by all the members of the group, different consensus functions or strategies have been derived to characterize how much the item satisfies the group as a whole. For example, group consensus may be estimated based on disagreement and relevance, where disagreement captures the differences in the item ratings between group members, while relevance corresponds,

for example, to the average of the item ratings by all group members, or, to the highest (in the optimistic strategy), or lowest (in the least misery strategy) item rating by a group member [13].

The “reverse” problem has also been studied, that is, recommending groups or packages of items to individual users (e.g., [12, 14]). For example, in *top-k composite recommendations*, each recommendation consists of a set of items, where both an interest score or rating and a cost is associated with each item [14]. The user specifies a maximum total cost for any recommended set. Motivated by online shopping applications, recommending *composite items* is proposed in [12], where a set of related satellite items are recommended along with each central item. In addition, the problem of recommending sets of items to a user when the items to be recommended must satisfy several constraints is considered in [10]. Particular focus is given on recommending sets of courses to students in the context of the CourseRank project.

In this paper, we consider a different aspect of group recommendations. Whereas in previous research, group recommendations focus on the relevance of the item to the group members, here, we study group recommendations when specific constraints apply to the members of the group based on individual preferences that the members of the group express for the other group participants. For example, a vacation package may seem more attractive to a user, if the other members of the group are of a similar age. Furthermore, a course may be recommended to a group of students that have similar or diverse backgrounds depending on the scope of the course, whereas a user may prefer a recommendation for a specific restaurant, if the accompanying group members are non smokers.

In the rest of this paper, we present in some detail a specific motivating example. Then, we introduce a formal model of the problem and outline a solution. We conclude the paper with a list of issues for future work.

## 2. MOTIVATING EXAMPLE

Assume a travel agent web site promoting vacation packages. As an example, consider the package “Gems of the Aegean” referring to a cruise itinerary at Eastern Mediterranean with tagline: “cities, sights and sensational scenery”. The cruise starts at July 16, its duration is 8 days, costs 900 euro and visits Athens, Kalamata, Aghios Nikolaos, Kusadasi and Marmaras. The goal of such a travel agent company is to locate an appropriate group of people which will be highly interested in the cruise.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires a fee and/or special permission from the publisher, ACM.

*PersDB* 2012

Copyright 2009 VLDB Endowment, ACM 000-0-00000-000-0/00/00.

Each user in the system has expressed a set of constraints concerning his/her choices. These constraints refer to either the vacation package itself or the other members of the group that potentially he/she will be a member. For example, Alice, a 34 years old teacher planning her vacations, can spend up to 1000 euro for less than 10 days in July, is interested in seeing the cities and sights of Eastern Mediterranean, and prefers to be accompanied by people over thirty most of which are college graduates.

Moreover, the company itself has defined a set of constraints expressing the preferences of the company for the group of customers that the company is targeting on. For instance, consider a constraint that directs the selection toward groups with middle-aged and senior people that is formulated as “include at least 30 users with average age greater than 40 years old”.

Given that for each user, there is an interest score available for each item, e.g., vacation package, that indicates how desirable the item is for the user, the task of the recommender is to recommend an item to a group of users, such that, (i) the (average) interest scores for all users in the group is maximized, and (ii) the constraints of all users in the group, as well as the constraints of the company, are satisfied.

### 3. MODEL AND PROBLEM STATEMENT

In this section, we present a model for constraints that captures different kinds of features for user group construction. We distinguish between user-to-item, user-to-group and group-to-group constraints. Then, we introduce the problem of constructing groups of users satisfying a set of constraints.

#### 3.1 User-to-Item Constraints

Let  $t$  be an item, i.e., a vacation package in our example, described by a set of attributes  $\{a_1, \dots, a_p\}$ , where each  $a_i$ ,  $1 \leq i \leq p$ , is of the form  $(a_i.attribute = a_i.value)$ . The description of the cruise *Gems of the Aegean* is shown in Figure 1.

|            |   |   |
|------------|---|---|
| title      | = | Gems of the Aegean                                    |
| type       | = | cruise  |
| place      | = | Eastern Mediterranean                                 |
| locations  | = | Athens, Kalamata, Aghios Nikolaos, Kusadasi, Marmaras |
| start_date | = | July 16, 2012   |
| end_date   | = | July 24, 2012   |
| duration   | = | 8   |
| cost       | = | 900   |

Figure 1: Item description example.

Let  $U$  be a set of users. We use  $score(u, t)$ ,  $u \in U$ , to denote the relevance, value or importance of item  $t$  for user  $u$ . The value of  $t$  for  $u$  can, for example, be directly induced by preferences that  $u$  has expressed in his/her profile. For example, given that Alice enjoys sightseeing in warm climates and likes to travel by sea, the *Gems of the Aegean* cruise will receive a high preference score. When there are no explicit preferences,  $score(u, t)$  may be the predicted rating of a recommender based on the past behavior of  $u$  or of other similar users.

In addition, each user  $u$  may specify a set of basic constraints  $\{b_1, \dots, b_q\}$  on the values of specific attributes of items that are candidates to be recommended. Each at-

tribute constraint consists of an attribute name, a binary operator and a value, i.e., has the form  $(b_i.attribute \theta_{b_i} b_i.value)$ . Attributes and values have the same form as in items. Binary operators include common operators, such as  $=$ ,  $\neq$ ,  $<$ ,  $\leq$ ,  $>$ ,  $\geq$  and *substring*. We refer to such constraints as *user-to-item constraints*. Figure 2 depicts the user-to-item constraints of Alice.

|            |        |                       |
|------------|--------|-----------------------|
| place      | =      | Eastern Mediterranean |
| start_date | $\geq$ | July 1, 2012          |
| end_date   | $\leq$ | July 31, 2012         |
| duration   | $<$    | 10                    |
| cost       | $\leq$ | 1000                  |

Figure 2: User-to-item constraints example.

Such user constraints are fulfilled with respect to an item  $t$ , if and only if, every attribute constraint of the user is satisfied by some attribute of  $t$ . Formally:

**DEFINITION 1 (USER-TO-ITEM CONSTRAINTS).** *An item  $t$  is a set  $\{a_1, \dots, a_p\}$ , where each  $a_i$ ,  $1 \leq i \leq p$ , is of the form  $(a_i.attribute = a_i.value)$ . The user-to-item constraints of user  $u$  is a set  $\{b_1, \dots, b_q\}$ , where each  $b_j$ ,  $1 \leq j \leq q$ , is of the form  $(b_j.attribute \theta_{b_j} b_j.value)$  and  $\theta_{b_j} \in \{=, \neq, <, \leq, >, \geq, substring\}$ .*

*$t$  satisfies the user-to-item constraints of  $u$ , if and only if,  $\forall b_j, \exists a_i$ , such that,  $a_i.attribute = b_j.attribute$  and  $((a_i.value) \theta_{b_j} (b_j.value))$ ,  $1 \leq i \leq p$ ,  $1 \leq j \leq q$ .*

For example, the package of Figure 1 satisfies the user-to-item constraints of Figure 2, which means that the personal constraints of Alice for the cruise are fulfilled.

#### 3.2 User-to-Group Constraints

Our overall goal is to form a group of users for which the recommended item is highly valuable for the group as a whole. Apart from the user-to-item constraints, we are also interested in satisfying user requirements that refer to user choices concerning the other members of the group. The fulfilling of these requirements works towards ensuring low disagreement between the group members. We refer to such constraints as *user-to-group constraints*.

In particular, similar to items, we assume that each user  $u \in U$  is described by a set of attributes  $\{x_1, \dots, x_s\}$ , where each  $x_i$ ,  $1 \leq i \leq s$ , is of the form  $(x_i.attribute = x_i.value)$ . For example, an attribute can be *name*, *education*, *occupation*, *gender* or *age*. See, for instance, Figure 3 for a user description.

|            |   |                  |
|------------|---|------------------|
| name       | = | Alice            |
| education  | = | college graduate |
| occupation | = | educator         |
| gender     | = | female           |
| age        | = | 34               |

Figure 3: User description example.

Each user defines a set of basic constraints  $\{y_1, \dots, y_r\}$  on the values of specific attributes of the other users belonging to his/her group. For example, consider the preference of Alice for groups with members over thirty, i.e.,  $age > 30$ . This kind of constraints, called *value constraints*, are similar to the user-to-item constraints.

More complex constraints that express aggregation requirements on the values of the set, termed *aggregation constraints*, are also permitted. In particular, the constraint  $z_i$

defined as:

$(aggr_{\mathcal{G}}(z_i.attribute) \theta_{z_i} z_i.value)$ , where  $\theta_{z_i} \in \{=, <, \leq, >, \geq\}$  and  $aggr_{\mathcal{G}} \in \{avg_{\mathcal{G}}, sum_{\mathcal{G}}, max_{\mathcal{G}}, min_{\mathcal{G}}, count_{\mathcal{G}}\}$  is a constraint respectively on the average, sum, maximum, minimum or number of the values of  $z_i.attribute$  of group  $\mathcal{G}$ . For example, take Scott, a 20 years old college student. The constraint  $avg_{\mathcal{G}}(age) < 25$  of Scott means that he is interested in groups with users with average age smaller than 25 years old.

Value and aggregation constraints apply to all members of the group. Often, we want to specify constraints on subsets of the group. To this end, we define *composite constraint*  $w_i$  of the form:

*include at least  $l$  users with  $(w_i.attribute \theta_{w_i} w_i.value)$ , or include at least  $l$  users with  $(aggr_{\mathcal{G}}(w_i.attribute) \theta_{w_i} w_i.value)$ .*

For example, *include at least 10 users with  $(age > 35)$* , expresses the preference of a user in groups with at least 10 users with age greater than 35 years old. Clearly, we could also consider conjunction of constraints. An example such constraint could be for groups that include at least 10 women of average age below twenty.

Given a group  $\mathcal{G}$  of users and a user  $u$ , we say that the user-to-group constraints of  $u$  are fulfilled with respect to  $\mathcal{G}$ , if and only if, every value constraint of  $u$  is satisfied by every user in  $\mathcal{G}$  and every aggregation and composite constraint of  $u$  is satisfied by the members of  $\mathcal{G}$  as a whole.

The following definition formalizes our model of user-to-group constraints.

**DEFINITION 2 (USER-TO-GROUP CONSTRAINTS).** *Let  $u$  be a user described by a set of attributes  $\{x_1, \dots, x_s\}$ , where each  $x_i$ ,  $1 \leq i \leq s$ , is of the form  $(x_i.attribute = x_i.value)$ , and  $\mathcal{G}$  be a set of users.  $\mathcal{G}$  satisfies the user-to-group constraints of  $u$ , if and only if:*

- (i) [ $\mathcal{G}$  satisfies the value constraints of  $u$ ] *For each value constraint  $y$  of the form  $(y.attribute \theta_y y.value)$  of  $u$ ,  $\theta_y \in \{=, \neq, <, \leq, >, \geq, substring\}$ , there exists, for each user in  $\mathcal{G}$ , a  $x$  attribute, such that,  $x.attribute = y.attribute$  and  $((x.value) \theta_y (y.value))$ .*
- (ii) [ $\mathcal{G}$  satisfies the aggregation constraints of  $u$ ] *For each aggregation constraint  $z$  of  $u$ , it holds  $aggr_{\mathcal{G}}(z.attribute) \theta_z z.value$ , where  $\theta_z \in \{=, <, \leq, >, \geq\}$  and  $aggr_{\mathcal{G}} \in \{avg_{\mathcal{G}}, sum_{\mathcal{G}}, max_{\mathcal{G}}, min_{\mathcal{G}}, count_{\mathcal{G}}\}$ .*
- (iii) [ $\mathcal{G}$  satisfies the composite constraints of  $u$ ] *For each composite constraint “include at least  $l$  users with  $w$ ” of  $u$ , there exist at least  $l$  users in  $\mathcal{G}$  forming a group  $\mathcal{G}'$ ,  $\mathcal{G}' \subseteq \mathcal{G}$  and  $|\mathcal{G}'| = l$ , such that,  $\mathcal{G}'$  satisfies the value or aggregation constraint  $w$  of  $u$ .*

### 3.3 Group-to-Group Constraints

Both user-to-item and user-to-group constraints describe limitations from the user, or customer, perspective. From the perspective of the company, *group-to-group constraints* refer to a set of properties that the group under construction must satisfy. These properties express the requirements of the company concerning the group that a product, or item, is targeting on.

As with the user-to-group constraints, we distinguish group-to-group constraints into value, aggregation and composite ones. For example, assume that the travel agent company managing the package *Gems of the Aegean* targets

mainly college graduates or say, middle-aged and senior customers. In this case, the company aims at a group  $\mathcal{G}$  that should qualify, for instance, the composite constraint *include at least 20 users with education = ‘college graduate’* or the composite constraint *include at least 20 users with  $avg_{\mathcal{G}}(age) > 40$* . When such constraints are fulfilled for a group  $\mathcal{G}$ , we say that  $\mathcal{G}$  *satisfies the group-to-group constraints*. (The formal definition of group-to-group constraints is skipped, since it is similar to the definition of user-to-group constraints.)

### 3.4 Problem Statement

A group  $\mathcal{G}$  is called *satisfiable* if all the *user-to-item*, *user-to-group* and *group-to-group* constraints for  $\mathcal{G}$  are fulfilled with respect to an item  $t$ .

**DEFINITION 3 (SATISFIABLE GROUP).** *Given an item  $t$  and a group  $\mathcal{G}$ ,  $\mathcal{G}$  is satisfiable for  $t$ , if and only if:*

- (i)  $t$  satisfies the user-to-item constraints of  $u$ ,  $\forall u \in \mathcal{G}$ ,
- (ii)  $\mathcal{G}$  satisfies the user-to-group constraints of  $u$ ,  $\forall u \in \mathcal{G}$ , and
- (iii)  $\mathcal{G}$  satisfies the group-to-group constraints.

Next, we formally define the problem of finding an appropriate group of users with the maximum relevance score for a specific item.

**DEFINITION 4 (PROBLEM DEFINITION).** *Given an item  $t$ , a set of users  $U$  and an integer  $k$ , identify the group of users  $\mathcal{G}$ ,  $\mathcal{G} \subseteq U$ , with cardinality  $k$ , such that:*

- (i)  $\mathcal{G}$  is satisfiable for  $t$  and
- (ii)  $\mathcal{G}$  has the maximum value  $score(\mathcal{G}, t)$ , where

$$score(\mathcal{G}, t) = \sum_{u \in \mathcal{G}} (score(u, t)),$$

among all satisfiable groups with cardinality  $k$ .

In this paper, we assume that constraints are strong, in the sense, that for recommending an item to a group all related constraints must be satisfied. One could also envision models where users prioritized their constraints, or where one type of constraint is more important than some other, for instance, user-to-group constraints may be considered as more relevant than group-to-group constraints. Further, we assume that it is possible to form satisfiable groups. Clearly, in the general case, it may not be possible to construct such groups. In this case, we should look for approximate solutions, for example, for a group that satisfies the largest number of constraints or for a group that satisfies the constraints of the majority of its members.

## 4. A GREEDY ALGORITHM

Given an item  $t$ , a satisfiable group  $\mathcal{G}$  for  $t$  is a set of users that comply with a set of user-to-item, user-to-group and group-to-group constraints. For this specific item  $t$ , a technical challenge is to efficiently construct the satisfiable group  $\mathcal{G}$  with the highest value  $score(\mathcal{G}, t)$ , given a budget  $k$  on the size of  $\mathcal{G}$ .

A brute-force method to identify such a group is to first construct all combinations of  $k$  users forming a satisfiable

group  $\mathcal{G}$  and then pick the one with the maximum  $score(\mathcal{G}, t)$ . Constructing the group  $\mathcal{G}$  using such a straightforward algorithm is computational costly, since the number of groups to examine for satisfyingness can be overwhelming even for a small number of users. As a result, we propose an alternative algorithm that computes an approximate group  $\mathcal{G}$ .

In particular, we use the following intuitive heuristic. We incrementally construct a group of users by selecting at each step a user that: (i) adds the most to the score value of the group and (ii) after joining the group of users already selected, the group satisfies more constraints than after adding any other user.

More specifically, let  $t$  be an item,  $U$  be a set of users and  $\mathcal{G}$  be the set we want to construct. Let also  $\mathcal{A}$  be the users in  $U$ , such that,  $\forall u \in \mathcal{A}$ ,  $t$  satisfies the user-to-item constraints of  $u$ . Note that, we assume that there is at least one satisfiable group, thus item  $t$  satisfies the user-to-item constraints of at least  $k$  users, i.e.,  $|\mathcal{A}| \geq k$ . Initially,  $\mathcal{G}$  is empty. We first construct a single-user group by adding to  $\mathcal{G}$  a user  $u$ ,  $u \in \mathcal{A}$ , randomly selected among the ones with the maximum value  $score(u, t)$ . Then, at each step, we select the users in  $\mathcal{A} \setminus \mathcal{G}$  with the maximum value score for  $t$ , and add to  $\mathcal{G}$  the one that satisfies along with the other members of  $\mathcal{G}$  the largest number of user-to-group and group-to-group constraints. This procedure stops after generating a group  $\mathcal{G}$  with  $k$  users. Algorithm 1 illustrates our *Group Construction Algorithm*.

---

#### Algorithm 1 Group Construction Algorithm

---

**Input:** An item  $t$ , a set of users  $U$  and an integer  $k$ .

**Output:** A group  $\mathcal{G}$ ,  $\mathcal{G} \subseteq U$ , of  $k$  users.

---

```

1: begin
2:  $\mathcal{G} \leftarrow \emptyset$ ;
3: find the users  $\mathcal{A}$ ,  $\mathcal{A} \subseteq U$ , such that,  $\forall u \in \mathcal{A}$ ,  $t$  satisfies the
   user-to-item constraints of  $u$ ;
4: find the users  $\mathcal{B}$ ,  $\mathcal{B} \subseteq \mathcal{A}$ , with the maximum value score for  $t$ ;
5: randomly select a user  $u \in \mathcal{B}$ ;
6:  $\mathcal{G} \leftarrow \mathcal{G} \cup \{u\}$ ;
7: while  $|\mathcal{G}| < k$  do
8:   find the users  $\mathcal{C}$ ,  $\mathcal{C} \subseteq \mathcal{A} \setminus \mathcal{G}$ , with the maximum value score
     for  $t$ ;
9:   select the user  $u_{add} \in \mathcal{C}$ , such that,  $\mathcal{G} \cup \{u_{add}\}$  satisfies the
     most user-to-group and group-to-group constraints compared to the groups
      $\mathcal{G} \cup \{u\}$ ,  $\forall u \in \mathcal{C}$ ;
10:   $\mathcal{G} \leftarrow \mathcal{G} \cup \{u_{add}\}$ ;
11: end while
12: return  $\mathcal{G}$ ;
13: end

```

---

We illustrate the final step of this algorithm with the following example. Consider the package *Gems of the Aegean* and a travel agent company that wants to construct a group of 80 users. At the last step, the 79 users have already been selected. Then, the users, say Alice and Scott, with the maximum interest score for the cruise, say 0.9, are identified. The cruise satisfies the user-to-item constraints of both users. Based on the assumption that when adding Alice to the group more user-to-group and group-to-group constraints are fulfilled than when adding Scott, Alice is the last user that will join the group.

Note that in line 9 of the algorithm, we are interested in locating the user that when joining a group, the group fulfills more constraints or, in other words, violates less constraints than when a different user joins the group. Several meanings can be assigned to the expression “*fulfills more*”

or, respectively, “*violates less*” constraints. For instance, assume a value user-to-group constraint expressing a preference of a user  $u$ . Assume also the following two scenarios: (i) all users in a group violate the value constraint of  $u$  or (ii) all users, except one, fulfill this constraint. It is clear that in both cases the group does not satisfy the constraint of  $u$ . However, different policies for counting the number of the violated constraints can be applied. For this example, for instance, we can consider a scheme that assigns a weight to each constraint depending on the number of the non-satisfied users, instead of considering each constraint as being either satisfiable or not.

## 5. DISCUSSION AND FUTURE WORK

In this short position paper, we introduced the generalized group formation problem: how to form a group of users for recommending an item, such that, all members of the group are satisfied both from the selection of the recommended item and from the selection of the other group members. Whereas the first component of the model, that is, group consensus on an item, has received considerable attention, the other dimension of the problem, that is, group consensus on the other group members is, to our knowledge, novel.

In this paper, we outlined a model for the problem and presented a first greedy algorithm for its solution. Clearly, there are many directions for future work including modeling constraints, combining item and group preferences, efficient algorithms and finally implementations in specific contexts. Next, we elaborate on these issues a bit further.

We modeled user preferences on items and other group members as constraints on the values of the attributes of the item and the user respectively. Other models are feasible as well. In particular, the user-to-item part can be expressed using traditional content or collaborative filtering approaches and be replaced by a single relevance value. One can envision a similar approach to expressing the user-to-group part. Recommendation techniques could also be used to recommend group members that would be compatible with a specific user, for example, similar users or users that liked similar items in the past.

In many cases, a group that satisfies all user-to-item and user-to-group constraints may not exist. In such cases, an appropriate consensus function should be sought for. For instance, one may ask for a group that satisfies the largest number of its members. Other approaches, such as relaxing constraints or approximating them, are also feasible. Further, an item may be highly relevant to many users but these users may be incompatible with each other. In this case, it may be wise to recommend this item to a less relevant but more compatible group.

Moreover, depending on the type and complexity of the constraints, more efficient algorithms than the greedy algorithm can be designed. Finally, in this paper, we used travelling as an example. Group constraints apply in many areas, where people perform activities in groups. For example, one could explore them in conjunction with social networks.

## 6. ACKNOWLEDGMENTS

The work of the first author was carried out during the tenure of an ERCIM “Alain Bensoussan” Fellowship Programme. This Programme is supported by the Marie Curie

Co-funding of Regional, National and International Programmes (COFUND) of the European Commission. The work of the second author is partially supported by project “Intersocial”, financed by the European Territorial Cooperation Operational Programme “Greece-Italy” 2007-2013, co-funded by the European Union (European Regional Development Fund) and by National Funds of Greece and Italy.

## 7. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.
- [2] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *RecSys*, pages 119–126, 2010.
- [3] L. Boratto and S. Carta. State-of-the-art in group recommendation and new approaches for automatic identification of groups. In A. Soro, E. Vargiu, G. Armano, and G. Paddeu, editors, *Information Retrieval and Mining in Distributed Environments*, volume 324 of *Studies in Computational Intelligence*, pages 1–20. Springer Berlin / Heidelberg, 2011.
- [4] J. S. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52, 1998.
- [5] R. Burke. Knowledge-based Recommender Systems. In *Encyclopedia of Library and Information Systems*, volume 69, 2000.
- [6] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada. Enhancing group recommendation by incorporating social relationship interactions. In *GROUP*, pages 97–106, 2010.
- [7] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87, 1997.
- [8] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *ACM DL*, pages 195–204, 2000.
- [9] I. Ntoutsis, K. Stefanidis, K. Nørnvåg, and H.-P. Kriegel. grecs: A group recommendation system based on user clustering. In *DASFAA (2)*, pages 299–303, 2012.
- [10] A. G. Parameswaran, P. Venetis, and H. Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Trans. Inf. Syst.*, 29(4):20, 2011.
- [11] M. J. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.
- [12] S. B. Roy, S. Amer-Yahia, A. Chawla, G. Das, and C. Yu. Constructing and exploring composite items. In *SIGMOD Conference*, pages 843–854, 2010.
- [13] S. B. Roy, S. Amer-Yahia, A. Chawla, G. Das, and C. Yu. Space efficiency in group recommendation. *VLDB J.*, 19(6):877–900, 2010.
- [14] M. Xie, L. V. S. Lakshmanan, and P. T. Wood. Breaking out of the box of recommendations: from items to packages. In *RecSys*, pages 151–158, 2010.