Good Beginning Makes Good Ending: Coordinating Disagreement and Satisfaction in Group Formation for Recommendation

ABSTRACT

Group recommendation has attracted significant research efforts for its importance in benefiting a group of users. There are two steps involved in this process, which are group formation and making recommendations. The studies on making recommendations to a given group has been studied extensively, however seldom investigation has been put into the essential problem of how the groups should be formed. As pointed in existing studies on group recommendation, both satisfaction and disagreement are important factors in terms of recommendation quality. Satisfaction reflects the degree to which the item is preferred by the members; while disagreement reflects the level at which members disagree with each other. As it is difficult to solve group formation problem, none of existing studies ever considered both factors in group formation.

This paper investigates the satisfaction and disagreement aware group formation problem in group recommendation. In this work, we present a formulation of the satisfaction and disagreement aware group formation problem, and further show its NP-Hardness. We design an efficient optimization algorithm based on Projected Gradient Descent and further propose a swapping alike algorithm that accommodates to large datasets. We conduct extensive experiments on real-world datasets and the results verify that the performance of our algorithm is close to optimal. More importantly, our work reveals that proper group formation can lead to better performances of group recommendation in different scenarios. To our knowledge, we are the first to study the group formation problem with satisfaction and disagreement awareness for group recommendation.

ACM Reference format:

. 1997. Good Beginning Makes Good Ending: Coordinating Disagreement and Satisfaction in Group Formation for Recommendation. In *Proceedings of ACM Woodstock conference, El Paso, Texas USA, July 1997 (WOODSTOCK'97),* 10 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Recommender Systems give suggestions (on information and items) to users and are useful in countless scenarios when users face choices. While a considerable number of recommender systems are personalized, many activities are group based and personalized recommendation can not work when making recommendation to groups. Some off-line websites like Meetup and Plancast allow users to form groups and join in same activities [14]. Companies also need to segment users into groups and make group-specific strategies for certain business purposes [15]. Travel agents also need to partition tourists into groups for different travel plans and trajectories[21]. Notice that some groups are persistent (like families and friends) while some groups are ephemeral (like users on Meetup and segmented customers in business intelligence). In our work, we focus on non-persistent groups in recommendation.

Consider the example when users on Meetup (they do not know each other and may not contact each other before knowing each other) decide to attend group activities. There are two groups for choosing, including a book reading group, a music sharing group. Each group will choose particular books and music based on the preferences of component members. Since users do not know about other group members in advance before the group is formed, it is possible that a user chooses a group but does not get satisfied with the activity. In this case, a proper group formation beforehand will lead to a superior satisfaction for the users.

Group recommendation contains two steps: group formation and making recommendation to formed groups. For first step (group formation), only one paper [21] has considered group formation with an objective of maximizing group satisfaction (which reflects the degree to which the item is preferred by the members). For the second step, the studies focus on making recommendation to given groups are more sufficient. In these studies, the groups are assumed to be formed already. [1] proposed to consider both relevance and disagreement (reflect the level at which members disagree with each other) in recommendation, which provides more effective recommendations than considering only satisfaction.

Therefore there exists a huge gap between the two steps: although both satisfaction and disagreement are seen as two important factors in making recommendations to groups, no previous work has ever considered both satisfaction and disagreement in group formation. However, it is quite difficult to consider satisfaction and disagreement in group formation at the same time. Usually, there exists no solution that achieves highest satisfaction and lowest disagreement simultaneously. Therefore, a balance between these two objectives needs to be found. Moreover, group formation is different from making recommendations to existing groups. When making recommendations to existing groups, the satisfaction can be computed by following a specific semantic,

In our paper, we try to bridge the gap by proposing a unified framework that considers both satisfaction and disagreement at the first step (group formation). This problem aims at partitioning users into a fixed number of groups, so that once the items are recommended based on group recommendation semantics, the overall satisfaction of these groups can be maximized and the disagreement inside the groups can be minimized. Our strategic group formation is of potential interest to all group recommender system applications, as long as they use certain recommendation semantics. Instead of ad-hoc group formation [23][6][11], or grouping individuals based on similarity [10], or meta-data (e.g., socio-demographic factors [6]), we explicitly embed the underlying group recommendation semantics in the group formation phase, which improves recommendation quality.

Since our problem aims to partition users into groups with a good recommendation quality, it is close to clustering in nature. However, this problem can not be easily solved via clustering methods. First, most clustering algorithms work by defining distances that depict the similarity between users, which requires proper metrics for distance evaluation, which is not available in this problem; second, our problem optimizes two objectives simultaneously with a balance, traditional algorithms can not cluster users to achieve this.

More specifically, we formulate the Disagreement-Aware Group Formation problem as an integer programming problem (non-semidefinite quadratic programming) and prove that it is NP-Hard thus can not be solved with an optimal solution in polynomial time. As a result, neither combinatorial optimization methods nor common clustering algorithms can be directly applied to solve the problem. Considering the inefficiency of these two approaches, we adopt the iterative optimization methods to tackle with the problem, which origins from the widely used Gradient Descent algorithm. Since this approach is usually applied to unconstrained optimization problems, Projected Gradient Descent (PGD) algorithm is adapted to solve the optimization problem with constraints. However, the PGD algorithm has a computational drawback that limits its use in large datasets (it needs to compute the projected gradient in each iteration). Therefore, we propose a swapping alike algorithm that preserves the nature of projected gradient descent but only needs easier computations. As shown in experiments, our algorithms based on projected gradient descent and swapping alike procedures outperform other benchmark algorithms significantly, and our result is close to the optima.

The main contributions of this work include the following points: (1) As a first step of group recommendation, group formation is essential to the group recommendation performance, but has not been well studied. Meanwhile, It has been pointed out that disagreement is an important factor in group recommendation [1], yet no work has ever considered it in group formation. To our knowledge, we are the first to incorporate group disagreement as an explicit recommendation semantic into group formation. We formalize it into an integrated optimization framework and show its NP-Hardness; (2) We design an optimization algorithm that originates from Projected Gradient Descent and simplify it to a swapping alike algorithm; Notice that our algorithm adopts a generic optimization scheme, it does not depend on the semantics selected for group recommendation and works well for satisfaction maximization objective. This shows the scalability and generality of our framework and algorithm; (3) We conduct extensive experiments based on real-world datasets and the results are shown to be close to the optima, which validates our theory and proves that proper group formation can improve group recommendation quality in different scenarios significantly.

The rest of the paper is organized as follows: Section 2 briefly introduces the related works; Section 3 formally introduces disagreement-aware group formation problem, formulates it with an integer programming framework and proves its NP-Hardness; Section 4 introduces our algorithms based on Projected Gradient Descent and a simplified swapping-alike algorithm from the adaption of PGD; Section 5 presents the experimental results and the conclusions are in Section 6.

2 RELATED WORK

A collective of strategies that aggregate the individual information as group preferences are summarized in [13]. The semantics of group recommendation are formally proposed in [1], where the semantics about satisfaction and disagreement are introduced. Since then, more works considering how to make effective group recommendations are proposed: [5] tries to learn a factorization of latent factor space into subspaces that are shared across multiple behaviors. [8] considers the problem of recommending friends who are interested in joining the users for some activities in a location based social network. [18] considers the problem of recommendation of social media content to leaders (owners) of online communities within the enterprise. However, none of them considers the group formation problem in the group recommendation context.

The co-clustering technique is widely used in the area of recommender systems for considering both users and items in clustering. Spectral co-clustering treats the users and items as nodes in a bipartite graph and aims at minimizing the cut between clusters [7]. This is close to the group formation problem in form, but differs on some important aspects. First, spectral co-clustering clusters items into disjoint clusters, while in group formation different groups may be recommended some common Top-K rated items; Second, spectral co-clustering clusters all users and items into clusters, while the group formation problem only partitions users into groups. Cases are similar for other co-clustering algorithms such as Bregman Co-clustering [3] and Bayesian

Coclustering [22]. These algorithms cluster users and items into clusters so that the ratings inside each cluster exhibit low variances. We also include a clustering algorithm [21] in our experiment, which evaluates the user similarity based on their preferences on the items.

Meanwhile, there have been considerable number of works on a similar problem known as Team Formation [16][2][9]. The objective of team formation is to form a team of experts with different skills to accomplish some skill required tasks. [12] considers the possibility that some users may quit during work and try to form a robust team. Although both team formation and our problem aim to form a set of users based on some objectives, our work differs from the team formation papers in two important aspects: first, in this problem, users and tasks are associated with skills while in group formation for recommendation, such explicit connections do not exist; second, team formation mainly focus on user-task matching in terms of cost/budget minimizing while our problem emphasis matching among users in terms of satisfaction and disagreement.

Some works about group recommendation also partition the users into groups first then provide recommendations to the groups respectively. Some works partition users into groups randomly or cluster users into groups based on their profiles [20][10]. However, none of them considers the group formation problem from the perspective of group recommendation. [21] studies the group formation problem that aims to maximize the group satisfaction. Our work differs from this work in two important aspects: first, we consider both satisfaction and disagreement of groups in recommendation context while the previous work only considers satisfaction; second, we propose an efficient algorithm to solve the problem which originates from generic optimization method and does not rely on the group recommendation semantics while the algorithm proposed in [21] works in specific semantics.

3 PROBLEM FORMULATION

In this section, we formulate the Satisfaction and Disagreement Aware Group Formation (SDAGF) problem. First we introduce the group recommendation semantics, which have been used to evaluate the quality of recommendation to groups. Then we introduce SDAGF problem based on the semantics and formulate it as an integer programming.

3.1 Group Recommendation Semantics

We first give some introductions about the semantics in group recommendation problems, which have been widely used in related researches [1][21][17]. As a common setting in recommender systems, the individual preference of an individual user i on item j is depicted as a number $R_{ij} \in [R_{min}, R_{max}]$.

Definition 3.1. **Group Satisfaction:** Given an item j and a group of users U, the satisfaction score $Sc(U, j)$ of the group given the item recommended to them is defined as a function in $[R_{min}, R_{max}]$: $Sc(U, j) = f(\{R_{ij}, i \in U\})$. The function f is different according to the semantics, for Aggregated Voting semantic (which is adopted in this paper): $f(\{R_{ij}, i \in U\}) = \sum_{i \in U} \frac{1}{|U|} R_{ij}$.

Notice that some other semantics for describing satisfaction also exist, including Least Misery ($Sc(U, j) = \min_{i \in U} R_{ij}$) [1] and Multiplicative ($Sc(U, j) = (\prod_{i \in U} R_{ij})^{\frac{1}{|U|}}$) [17]. Though we do not include them in the problem formulation, the results in experiments show that the groups formed by our approach can lead to good performances in other semantics too.

Definition 3.2. **Group Disagreement:** Given an item j and a group of users U, the disagreement $D(U, j)$ of the group on item j is defined as a function in $[R_{min}, R_{max}]$: $D(U, j) = g(\{R_{ij}, i \in U\})$, the deviation of individual satisfaction from group average is used to evaluate the disagreement: $\sqrt{\frac{1}{|U|} \sum_{i \in U} |R_{ij} - \sum_{i \in U} \frac{1}{|U|} R_{ij}|^2}$.

The Group Satisfaction semantic aggregates the ratings of items recommended to all users inside the group while the Group Disagreement evaluates the consistency of ratings from group members. Since group recommendation concerns about the recommendation quality to a group of users rather than a single user, it is not enough to consider the satisfaction of individual users, a certain level of consistency is also of great importance. A low disagreement means the satisfaction achieved by a single user does not deviate much from the group average, so that the satisfactions achieved by users do not have severe differences. When all other conditions are equal (in this paper, the condition refers to the satisfaction), an item that members agree more on should have a higher score than an item with a lower overall group agreement. This provides a certain level of consistency to the group recommendation. Most recommender systems follow the Top-K recommendation, the Top-K items with high satisfaction and low disagreement are recommended to each group in our work.

3.2 Satisfaction and Disagreement Aware Group Formation (SDAGF)

Given the definitions introduced above, we formally introduce the Satisfaction and Disagreement Aware Group Formation (SDAGF) problem with an optimization framework. First we introduce the group formation problem with single objective and then the bi-objective optimization problem with an integer programming framework.

3.2.1 Group Formation with Single Objective. The group formation problem aims to divide the users into a fixed number (G) of groups such that the satisfaction is maximized or the disagreement is minimized. Depending on different objectives, the problem can be formulated as satisfaction-maximizing group formation or disagreement-minimizing group formation. More formally, given a set of users U and a set of items I, we want to divide the users into a fixed number (G) of groups such that:

- $\forall g, g' \in \{1, 2, ..., G\}$, we have $U_g \cap U_{g'} = \emptyset$ and $\cup_g U_g = U$, where U_g denotes the users in group g.
- satisfaction-maximizing: $\forall g \in \{1, 2, ..., G\}$, let the recommendation of each group follows Top-K procedure and the items recommended be denoted as I_g, we have a maximized objective function: $\sum_g \phi(g) \sum_{j \in I_g} Sc(U_g, j)$, where $\phi(g)$ is a weight for group g.
- disagreement-minimizing: $\forall g \in \{1, 2, ..., G\}$, let the recommendation of each group follows Top-K procedure and the items recommended be denoted as I_g, we have a minimized objective function: $\sum_g \phi(g) \sum_{j \in I_g} D(U_g, j)$, where $\phi(g)$ is a weight for group g.

Notice that there are weights for different groups respectively in the objective function. We set the weights as number of users inside groups. It is used to avoid the situation when a large number of users are put into a group but they have to sacrifice a lot to achieve get a consensus. In this case, smaller groups get good results but at the cost of the quality of large groups.

3.2.2 Group Formation with Bi-Objective Optimization. However, both satisfaction and disagreement are important to the quality of group recommendation, it is difficult to achieve both highest satisfaction and lowest disagreement at the same time. We use a linear scalarization method to solve the bi-objective optimization problem. Therefore the objective function can be written as:

$$\omega \sum_{g=1}^{G} \sum_{j \in I_g} |U_g| Sc(U_g, j) + (\omega - 1) \sum_{g=1}^{G} \sum_{j \in I_g} |U_g| D(U_g, j) \tag{1}$$

We set variables X_{ig} and Y_{jg} as indicator variables deciding whether user i is in group g and item j is recommended to group g respectively. $0 < \omega \leq 1$ is a trade-off factor between satisfaction and disagreement. When $\omega \to 1$, the objective leans towards satisfaction maximization while $\omega \to 0$, the objective leans towards disagreement minimization.

Based on this, the Disagreement-Aware Group Formation (SDAGF) problem is rewritten into an integer programming:

$$\begin{aligned}
\max.\ & \omega \sum_{g=1}^{G} \sum_{i \in U} \sum_{j \in I} R_{ij} X_{ig} Y_{jg} + \\
& (\omega - 1) \sum_{g=1}^{G} \sum_{i \in U} \sum_{j \in I} \sqrt{\sum_{i \in U} X_{ig} |R_{ij} - \frac{\sum_{i \in U} X_{ig} R_{ij}}{\sum_{i \in U} X_{ig}}|^2} X_{ig} Y_{jg} \\
s.t.\ & \\
& \sum_{g=1}^{G} X_{ig} = 1, \forall i \in U \\
& \sum_{j \in I} Y_{jg} = K, \forall g \in \{1, 2, ..., G\} \\
& X_{ig} = \{0, 1\}, \forall i \in U, g \in \{1, 2, ..., G\} \\
& Y_{jg} = \{0, 1\}, \forall j \in I, g \in \{1, 2, ..., G\}
\end{aligned} \tag{2}$$

	I_1	I_2	I_3
u_1	5	4	2
u_2	4	4	3
u_3	2	3	4

Rating Matrix

	G1	G2
Members	u_1, u_3	u_2
Recommendation	I_1, I_2	I_1, I_2
Satisfaction	7, 7	4, 4
disagreement	1.5,0.5	0, 0

Group Formation 3

	G1	G2
Members	u_1, u_2	u_3
Recommendation	I_1, I_2	I_2, I_3
Satisfaction	9, 8	3, 4
disagreement	1.5, 1.5	0, 0

Group Formation 1

	G1	G2
Members	u_2, u_3	u_1
Recommendation	I_2, I_3	I_1, I_2
Satisfaction	7, 7	5, 4
disagreement	0.5, 0.5	0, 0

Group Formation 2

Figure 1: An example for Group Formation.

Consider the two constraints in our problem: The first constraint requires that one user is in exactly one of the groups, while the second constraint requires that each group is recommended with K items. Based on the maximization objective and the constraints together, the optimal solution of our programming formalization chooses the top-K items with the highest objective function for each group.

Consider the example in Fig. 1 where five users (denoted as $u_i, i = 1, ..., 3$) give ratings to 3 items (denoted as $I_j, j = 1, ..., 3$). In this example, the users are divided into two groups and Top-2 items are recommended to each group. All the possible group formation solutions (together with the corresponding recommendations, satisfaction and disagreement for each recommended item) in the tables in Fig. 1.

3.3 Hardness of SDAGF problem

We prove that this problem is NP-Hard, regardless of the choice of ω. Due to page limits, we leave out the mathematical details and only present general-level idea here.

THEOREM 3.3. *The Disagreement Aware Group Formation problem is NP-Hard regardless of the choice of ω.*

PROOF. when $\omega = 1, K = 1$, the problem reduces to the group formation problem that maximizes the satisfaction of groups when only one item can be recommended to each group, which has been proved to be NP-Hard in [21]. We can adopt a similar procedure to prove its NP-Hardness when $K > 1$ by duplicating the recommended items K times.

Now we consider when $0 < \omega < 1$: w.l.o.g, we restrict ratings to be 1 or 0. Since $D(g) > 0, \forall g$, the decision version of the problem: $obj \geq \omega \times |G|K$ is equivalent to the decision version of the satisfaction maximization problem: $Sc \geq |G|K$. When $Sc \geq |G|K$, the ratings inside each group are all 1, which leads to a disparity of 0. This leads to an objective function of $obj \geq \omega \times |G|K$. Since the satisfaction maximization problem is NP-Hard, the problem $obj \geq \omega \times F$ is NP-Hard too. □

4 ALGORITHMS

In this section, we formally introduce the algorithms for Disagreement Aware Group Formation problem. Gradient descent methods are widely adopted for solving unconstrained optimization problems and they achieve good performances while preserving high efficiency in computation. However gradient descent can not be directly applied to our problem due to the existence of different constraints. Based on the intuition of Projected Gradient Descent, we propose a simplified PGD algorithm for this problem and further introduce a swapping alike algorithm.

4.1 PGD Algorithm for Group Formation

For general values of ω, the problem is no longer a bilinear programming, where the approximation algorithm can not be applied directly. But the severation of X and Y in optimization is a good idea. We use $Y_{jg}(1-Y_{jg})=0$ to represent the constraint $Y_{jg}\in\{0,1\}$, and we derive the KKT condition (Karush-Kuhn-Tucker conditions [4]), the condition for Y_{jg} (β and μ_{jg} are Lagrangian Multipliers) is:

$$\begin{aligned}\frac{\partial L}{\partial Y_{jg}} =&\omega\sum_{i\in U}R_{ij}X_{ig}+(\omega-1)\sum_{i\in U}\sqrt{\sum_{i\in U}X_{ig}|R_{ij}-\frac{\sum_{i\in U}R_{ij}X_{ig}}{\sum_{i\in U}X_{ig}}|^2X_{ig}}\\ &+\beta_g+\mu_{jg}(1-2Y_{jg})=0\end{aligned}\tag{3}$$

As one of the KKT conditions Eq.3 shows, the optimal value of Y is solely determined by the value of X, thus in each iteration, we first update the value of X and then determine the value of Y based on the updated X, which is an alternative optimization method.

PGD follows the gradient descent intuition so that the solution is updated along the gradient in each iteration. However, PGD can handle constraints by including a projection onto the set of constraints. Therefore we can go over the constraints and get the projected gradients accordingly.

We consider the update in each iteration: denote variables before iteration as X^0_{ig} and Y^0_{jg}, the stepsize of gradient descent as δ. Denote $s_i(g)$ as the projection of $\frac{\partial L}{\partial X_{ig}}$ and $s_j(g)$ as the projection of $\frac{\partial L}{\partial Y_{jg}}$. Thus before each iteration, the following constraints are satisfied:

$$\sum_{g=1}^{G}X^0_{ig}=1,\forall i\in U,\ \text{and}\ \sum_{j\in I}Y^0_{jg}=K,\forall g\in[1,G]$$

while after each iteration, the following constraints should be satisfied,

$$\begin{aligned}&\sum_{g=1}^{G}(X^0_{ig}+\delta s_i(g))=1,\forall i\in U,\ \text{and}\\ &\sum_{j\in I}(Y^0_{jg}+\delta s_j(g))=K,\forall g\in[1,G]\end{aligned}$$

Meanwhile, we want to ensure that the mapped gradients are close to $\frac{\partial L}{\partial X_{ig}}$, which is the fastest descent direction of objective function. This is equivalent to the following minimization problem, denote $L_i=[\frac{\partial L}{\partial X_{i1}},...,\frac{\partial L}{\partial X_{ig}},...],\forall i\in U$:

$$\begin{aligned}&\min\|s_i-L_i\|_2,\ s.t.\sum_g s_i(g)=0,\ \text{and}\\ &\begin{cases}s_i(g^p)\le 0,\forall g^p\in\{g:X_{ig}=1\}\\ s_i(g^n)\ge 0,\forall g^n\in\{g:X_{ig}=0\}\end{cases}\end{aligned}\tag{4}$$

Algorithm 1 Projected Gradient Descent (PGD)

Input: Rating matrix R, the set of users U and items I
Output: Formed groups: $X_{ig},\forall i\in U,g$; $Y_{jg},\forall j\in I,g$

1: Initialize the indicators: $X_{ig},\forall(i,g)$;$Y_{jg},\forall(j,g)$;
2: **while** $|F^T-F^{T+1}|\le\epsilon$ OR iter<MaxIter **do**
3: **for** each user $i\in U$: **do**
4: Solve the equality constrained convex optimization problem Eq. 4;
5: Compute X with projected gradient as in Eq. 5;
6: **end for**
7: **for** each group $g\in\{1,2,...,G\}$: **do**
8: Find K items as in Eq. 6;
9: **end for**
10: **end while**

This is a convex optimization problem which can be solved with an optimal solution in finite steps. We solve this problem for each user and get a projected gradient s_i, then we use it to update the current solution:

$$X^{t+1}_{ig}=X^t_{ig}+\delta s_i(g),\forall i\in U,g\in\{1,2,...,G\}\tag{5}$$

When the users are assigned to groups in a new iteration, we can get the items recommended to groups easily by taking the top K items with highest values of

$$\omega\sum_{i\in U}R_{ij}X_{ig}+(\omega-1)\sum_{i\in U}\sqrt{\sum_{i\in U}X_{ig}|R_{ij}-\frac{\sum_{i\in U}R_{ij}X_{ig}}{\sum_{i\in U}X_{ig}}|^2X_{ig}}\tag{6}$$

For clear understanding, the Projected Gradient Descent algorithm is presented in Alg. 1. F^T denotes the value of objective function at iteration T, ϵ is denoted as the threshold for the difference between objective functions in consecutive iterations.

4.2 Disagreement And Satisfaction aware Group Optimization (DASGO) Algorithm

As shown in previous sections, the key of Projected Gradient Descent is the projection of original gradient so that the update with projected gradient does not violate the constraints. We introduce a simple yet effective projection method for the problem which acts like a swapping between groups.

Consider X_{ig}, the projected gradient s_i and the original gradient L_i in current iteration: we need to solve the optimization problem for user i in Eq. 4, where

$$L_i(g)\approx\omega\sum_{j\in I}R_{ij}Y_{jg}+(\omega-1)\sum_{j\in I}\sum_{i\in U}\sqrt{\sum_{i\in U}X_{ig}|R_{ij}-\frac{\sum_{i\in U}R_{ij}X_{ig}}{\sum_{i\in U}X_{ig}}|^2Y_{jg}}\tag{7}$$

Since computing the exact solution of this sub-problem of Eq. 4 is time-consuming for large datasets (when $|U|$ is large), we relax the requirement of objective function so that the computed gradient is a descent direction for the objective, i.e. $L_i\cdot s_i\ge 0$ and we have the following constraint

Algorithm 2 Disagreement And Satisfaction aware Group Optimization (DASGO)

Input: Rating matrix R, the set of users U and items I
Output: Formed groups: $X_{ig}, \forall i \in U, g$; $Y_{jg}, \forall j \in I, g$

Initialize the group indicators of users and items: $X_{ig}, \forall i \in U, Y_{jg}, \forall j \in I$;
while $|F^t - F^{t+1}| \leq \epsilon$ OR iter<MaxIter **do**
  for each group g **do**
    Calculate the Top-K items of group g: $S(g, K) = \{j | Y_{jg} = 1, \forall j \in I\}$;
  end for;
  for each user i **do**
    for each group g **do**
      Calculate the gradient $L_i(g)$ as Eq. 7
    end for;
    Assign the user to $g = \max_{g \in [1,G]}\{L_i(g), \forall g\}$;
  end for;
end while

set (without objective functions):

$$s.t. \sum_g s_i(g) = 0,\ L_i \cdot s_i \geq 0,\ \text{and} \begin{cases} s_i(g^p) \leq 0, \forall X_{ig^p} = 1 \\ s_i(g^n) \geq 0, \forall X_{ig^n} = 0 \end{cases} \tag{8}$$

This new sub-problem has a simple solution. When $X_{ig^p} = 1$ and $L_i(g^p) \neq \max\{L_i(g)\}$:

$$s_i(g) = \begin{cases} 1, L_i(g) = \max\{L_i(g)\} \\ -1, X_{ig} = 1 \\ 0, \text{otherwise} \end{cases} \tag{9}$$

Otherwise, we have $s_i = 0$.

Judging from the derivation, the main idea of our swapping procedure is to swap users between groups. For a given group formation, we first calculate the Top-K recommended items in each group. Suppose that the items are fixed, we find those users who can obtain higher ratings of Top-K items if swapped into other groups. For those users, we finally swap them into the group where they can get the highest increase of objective function. We repeat the swapping procedure until no user can get higher increase on objective function by swapping. The detailed specification of the algorithm is presented in Alg. 2.

Therefore the swapping procedure provides a simple yet effective way to reach the local optima from an initial solution. Considering that mapping methods can vary, there can be different variations for the PGD algorithms.

5 EXPERIMENT

5.1 Experiment Settings

5.1.1 Datasets: The real-world datasets are chosen from MovieLens, Filmtrust and Epinions. The first two datasets are released by the two famous movie websites Movielens and Filmtrust. "Epinions" is an opinion sharing website where users can share their opinions towards all kinds of stuff. Some statistical details of the datasets are shown in Table 1. For ML-10M (MovieLens-10M, released by MovieLens) and Epinions, We choose 10000 (from ML-10M and Epinions respectively) users and 10000 items randomly. The ratings of these datasets take values from 1 to 5 and the missing entries are estimated with state-of-the-art Collaborative Filtering method, which is commonly used in the literature, such as [1] and [21]). In this way, we achieve the completed ratings matrix with the empty entries filled with the estimations by PMF (Probabilistic Matrix Factorization) [19].

Table 1: Statistics of the datasets.

Dataset	FilmTrust	ML-1M	ML-10M	Epinions
#Users	1,508	6,040	71,567	49,289
#Items	2,071	3,907	10,677	139,738

5.1.2 Algorithms for Comparisons: We compare our approaches with some state-of-art approaches, including:

GRD[21]: The Group RecommenDation (GRD) algorithm greedily selects the users with same highest satisfaction to form a group, until all the users are divided into G groups. The algorithm first hashed all the users with their Top-K items of their highest ratings, and therefore each user is represented as a sequence of IDs of the K items. Then, it finds $G-1$ sequences with the highest group satisfaction and form each sequence as a group, respectively. The remaining users are formed into the last group.

Spectral Co-Clustering (SCC)[7]: The Spectral Coclustering algorithm sees the rating matrix as a bipartite graph where the users and items are nodes on each side and the ratings are weights of links between nodes from two sides. The algorithm aims at coclustering nodes into a fixed number of clusters so that the weights inside clusters are maximized.

Bayesian Co-Clustering[22]: The BCC algorithm assumes that the users and items belong to different clusters with some different probabilities. The ratings inside the same cluster are assumed to be of a low variance.

KTD-Alg[21]: Apart from the algorithms above, we also adopt the benchmark algorithm used in [21], the algorithm evaluates the similarity of two users with Kendall-Tau Distance (KTD) and run the K-means algorithm to cluster the users, and we thus denote this algorithm with KTD-Alg.

PGD: It is the Projected Gradient Descent (PGD) algorithm proposed in this paper.

DASGO: It is the Disagreement and Satisfaction aware Group Optimization (DASGO) algorithm proposed in this paper.

5.1.3 Evaluation Metrics: Since there are two objectives for the evaluation of group formation quality in the objective function, we also provide two metrics for the experiment:

The first metric is the **Average Fulfilment (AF)**:

$$AF = \frac{\sum_i \sum_{j \in I_g} R_{ij}}{\sum_i \sum_{j \in I(i,K)} R_{ij}} \tag{10}$$

which represents how much the users are satisfied with the formed groups compared to the satisfaction from Top-K items of one's own, which is actually the optimal satisfaction the user can get. I_g denotes the set of items recommended to

Table 2: AF and AD of the Algorithms with the setting of $w = 0.8$, $G = 10$, $K = 10$, Higher AF and Lower AD values are better

Metrics	Average Fulfilment				Average Disagreement			
Dataset	ML-1M	F.T.	ML-10M	Epinions	ML-1M	F.T.	ML-10M	Epinions
GRD	0.921*	0.818*	0.841*	0.848*	0.555*	0.625*	0.544*	0.359*
SCC	0.954*	0.929*	0.850*	0.903*	0.448*	0.491	0.515*	0.369*
KTD	0.954*	0.912*	/	/	0.444*	0.490	/	/
BCC	0.954*	0.894*	0.853*	0.887*	0.443*	0.521*	0.459	0.346
DASGO	**0.966**	**0.942**	**0.893**	**0.921**	**0.399**	0.501	**0.457**	0.350
PGD	**0.971**	**0.951**	/	/	**0.397**	0.498	/	/

Table 3: AF and AD of the Algorithms with the setting of $w = 0.2$, $G = 10$, $K = 10$, Higher AF and Lower AD values are better

Metrics	Average Fulfilment				Average Disagreement			
Dataset	ML-1M	F.T.	ML-10M	Epinions	ML-1M	F.T.	ML-10M	Epinions
GRD	0.908*	0.689*	0.744*	0.787*	0.492*	0.269*	0.098*	0.181
SCC	0.947*	0.844*	0.761*	0.849*	0.413*	0.327*	0.098*	0.208*
KTD	0.943*	0.814*	/	/	0.414*	0.266*	/	/
BCC	0.946*	0.805*	0.779	0.849*	0.386*	0.288*	0.076*	0.211*
DASGO	**0.963**	**0.850**	0.773	**0.870**	**0.374**	**0.198**	**0.036**	**0.179**
PGD	**0.965**	**0.853**	/	/	**0.369**	**0.192**	/	/

the group g; $I(i, K)$ denotes the set of K items with highest ratings from user i.

The second metric is the **Average Disagreement (AD)**:

$$AD = \frac{\sum_g \sum_{j \in I_g} |U_g| D(U_g, j)}{K \times \sum_g |U_g|} \tag{11}$$

which evaluates the disagreement between users inside same groups on the recommendation.

Intuitively, AF evaluates the ratio of user ratings on the recommended items in their group against the ratings of their favourite items. Note that the optimal solution can never gain higher ratings than the sum of all the ratings of each user's favourite items, as a result, we have $AF \leq 1$; Therefore, higher AF and lower AD are expected. Meanwhile, we also use the value of objective function as a metric, as it represents the quality of group recommendation under different levels of trade-offs between disagreement and satisfaction.

In the following, we present the results of our experiments from the aspect of group formation quality with the metrics, the effects of different parameters on the quality, as well as the comparative analysis with other algorithms. The results presented in tables from later chapters are marked with $*$, indicating that the improvements of DASGO compared with baseline algorithms are statistically significant with a p-value of 0.01.

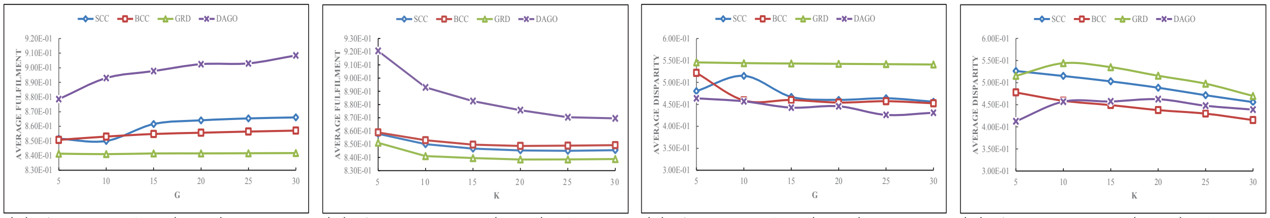
5.2 Performances under AF&AD Metric

The performances of the algorithms under the metrics of AF and AD are summarized in Table 2 and Table 3, where the settings we choose are $\omega = 0.8$ and $\omega = 0.2$ (for different levels of trade-off between satisfaction and disagreement), $G = 10$, and $K = 10$. We will tune the parameters (including the trade-off factor ω, the number of groups to be divided G and the number of items to recommend K) to see their effects in the following experiments. In this table, those values with $*$ have passed the significance test on the level of $p < 0.01$. Notice that PGD and KTD does not fit for large datasets, we list their performances on RGDS and FilmTrust.

From the results in the tables, we see that our algorithm has a remarkable better performance than the other benchmark algorithms on almost all datasets. Besides, our algorithm achieves not only better overall satisfaction, but also relatively lower disagreement. Notice that when $\omega = 0.8$, the objective leans towards maximizing the satisfaction rather than minimizing the disagreement, DASGO achieves **highest** AF on all datasets and also induces **low** disagreement; when $\omega = 0.2$, the objective leans towards minimizing the disagreement rather than maximizing the satisfaction, DASGO induces **lowest** disagreement on all datasets and also achieves **high** satisfaction. This indicates that DASGO has a good flexibility in accordance with the value of ω and outperforms other approaches given different specified objectives (determined by the value of ω).

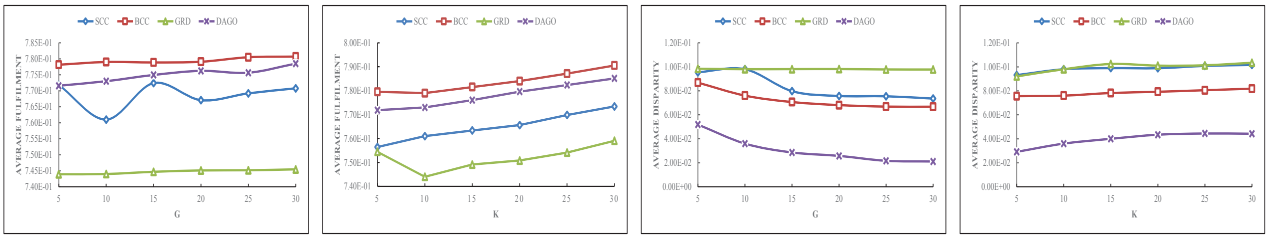
5.3 Performances under Various Parameters

Fig.2 and Fig.6 further show the AF and AD values under different choices of group number G and the number of recommended items K on dataset ML-10M with different values of ω. As shown in both figures, the impact of group number G is consistent when $\omega = 0.2$ and $\omega = 0.8$ respectively (see the sub-figures (a) and (c) in Fig.2 and Fig.6). A larger number of groups means more chances to induce better personalization in recommendation, which helps to reduce disagreement among group members and increase satisfaction for each individual user. Therefore AD decreases with the increase of group number while AF increases with the increase of group number.



(a) AF with $G \in (5, 30)$, K=10 (b) AF with $K \in (5, 30)$, G=10 (c) AD with $G \in (5, 30)$, K=10 (d) AD with $K \in (5, 30)$, G=10

Figure 2: AF and AD value with different G and K on ML-10M, $\omega = 0.8$



(a) AF with $G \in (5, 30)$, K=10 (b) AF with $K \in (5, 30)$, G=10 (c) AD with $G \in (5, 30)$, K=10 (d) AD with $K \in (5, 30)$, G=10

Figure 3: AF and AD value with different G and K on ML-10M, $\omega = 0.2$

The impact of K is relatively complex, and it is also related to the value of ω (see the sub-figures (b) and (d) in Fig.2 and Fig.6). Here we present a rough analysis: we can classify items into four categories: items with High Satisfaction and Low Disagreement (HSLD), items with High Satisfaction and High Disagreement (HSHD), items with Low Satisfaction and Low Disagreement (LSLD) and items with Low Satisfaction and High Disagreement (LSHD). Different values of ω means a different preference ordering over the four categories of items in group recommendation:

When $\omega \to 1$: the objective leans towards maximizing satisfaction, hence the group follows a preference ordering of $HSLD > HSHD > LSLD > LSHD$ (where "$a > b$" means the group prefers a over b). When K increases, items are recommended following the ordering of categories above. Therefore, it leads to a decrease of average satisfaction and the average disagreement first increases, then decreases and then increases. When $\omega = 0.8$, the change of AD is shown in Fig. 2 and it coincides with the analysis.

When $\omega \to 0$: the objective leans towards maximizing satisfaction, hence the group follows a preference ordering of $HSLD > LSLD > HSHD > LSHD$. When K increases, items are recommended following the preference ordering above. This leads to an increase of average disagreement and the average satisfaction first decreases, then increases and then decreases. When $\omega = 0.2$, the change of AF is shown in Fig. 6 and it basically coincides with the analysis.

Notice that ω acts as a trade-off between satisfaction and disagreement, therefore the two objectives can be impacted by the values of ω. As shown in Table. 4, a lower ω means the objective function considers the disagreement as a more important part, which typically leads to a lower value of disagreement and a loss of satisfaction. However, our algorithm does not cause much loss of satisfaction when ω is lower, and symmetrically does not cause too much loss of disagreement when ω is higher.

5.4 Performance on Other Satisfaction Semantics

As there are different semantics for group satisfaction, we present the performance of different algorithms under these semantics in Fig. 4. Since the objective function is a linear combination of satisfaction and disagreement, a higher objective function means better group formation for a fixed ω. in Fig. 4, the performance of GRD is used as a baseline and the improvements of other algorithms are presented. For different semantic of satisfaction, DASGO can be seamlessly adapted to it by swapping the users to the group with greatest increment of objective function in each iteration (similar to step 10 in Alg. 2).

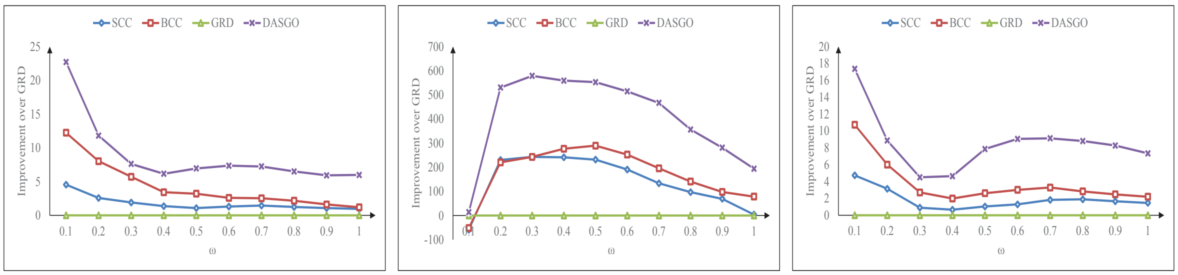
The results indicate that DASGO outperforms other methods regardless of which semantic for satisfaction is adopted. The results show the universality of DASGO algorithm to the group formation problem. Notice that the improvement over GRD on LM semantic is much more significant than the improvements on other semantics. The reason is that least misery concerns with the lowest rating of all users inside a same group, therefore the satisfaction of the group is relatively (actually smaller than other semantics in magnitude) small. Then a small increase of the objective function can cause great relative improvement.

5.5 Performance on Personalized Recommendation Metrics

We also conduct experiments with typical recommendation metrics for evaluation, including Precision, Recall, MAP and NDCG. We split each dataset into 5 folds and conduct a

Table 4: Performances of DASGO under different ω on ML-10M, $G = 10$, $K = 10$

ω	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
AD	0.0264	0.0358	0.0502	0.1356	0.2371	0.3672	0.4252	0.4574	0.4829	0.5001
AF	0.7550	0.7728	0.7841	0.8242	0.8522	0.8814	0.8874	0.8894	0.8903	0.8935
Obj.($\times 10^5$)	0.351	0.738	1.13	1.55	2.00	2.48	2.96	3.44	3.93	4.43



(a) satisfaction as average semantic (b) satisfaction as Least Misery semantic (c) satisfaction as Multiplicative semantic

Figure 4: Improvement of Objective Function Value over GRD on the ML-10M Dataset, $G = 10, K = 10$, satisfaction as different semantics with different values of ω

cross-fold validation with four folds as training set and the remaining fold as testing set. Since this work focuses on group formation for group recommendation, we compare the performances of group recommendation under different group formation methods. For items that have been seen or rated by the users, we do not count it as a relevant item in the metrics (i.e. set $R_{ij} = 0$ if user i already rated item j and use the predicted value for R_{ij} if user i has not rated item j). The experiments are conducted on all datasets and the results on Movielens-10M and Epinion datasets are presented due to page limit. We fix $\omega = 1$ for the recommendation task since the item recommendation metrics are used for evaluating the quality of personal recommendations which does not concern about the consistency of user satisfactions in group recommendation. The results are presented in Table. 5-6.

Table 5: Recommendation Performances on M.L.-10M, $G = 10$ with different Group Formations

Methods	Prec@10	Rec@10	MAP@10	NDCG@10
SCC	0.0856*	0.0987*	0.0252*	0.2724*
BCC	0.0800*	0.0902*	0.0207*	0.2539*
GRD	0.0856*	0.0980*	0.0258*	0.2708*
DASGO	**0.1131**	**0.1295**	**0.0339**	**0.3222**

Table 6: Recommendation Performances on Epinions, $G = 10$ with different Group Formations

Methods	Prec@10	Rec@10	MAP@10	NDCG@10
SCC	0.0103*	0.0276*	0.0081*	0.0418*
BCC	0.0101*	0.0269*	0.0080*	0.0421*
GRD	0.0118*	0.019*	0.0054*	0.0297*
DASGO	**0.0127**	**0.0423**	**0.0127**	**0.0620**

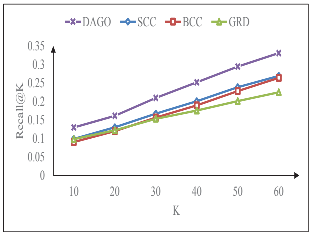
Based on the results presented above, we can get the conclusion that under the given group recommendation semantics (majority voting), our method provides a group formation with best recommendation quality. Although the metrics are used for evaluating personalized recommendation, they can still evaluate how close the group recommendations are to personalized recommendation.

We also present the results of item recommendation with various numbers of groups (G) and items to recommend (K). The results show that our algorithm keeps a superior performance over others with various G and K. More groups allow for more personalization for recommendation, therefore all the metrics get improved; more items to recommend can increase Recall, MAP and NDCG, but cause the decrease of Precision, which is similar to personalized recommendation.

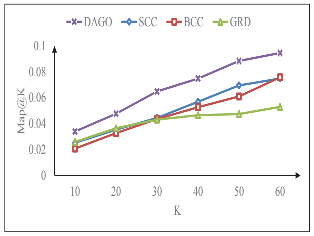
6 CONCLUSION

In this paper, we propose the Satisfaction and Disagreement Aware Group Formation problem which divides users into a fixed number of groups, so that the satisfaction of users can be maximized and the disagreement is minimized when the items are recommended following specific group recommendation semantics. As the studies on group recommendation are rich, both satisfaction and disagreement are important factors that impact the recommendation quality, none of the existing studies ever consider both factors in group formation problems. To the best of our knowledge, it is the first work to study the group formation problem that considers both satisfaction and disagreement simultaneously.

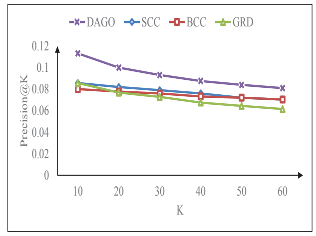
We present theoretical formulations for the satisfaction and disagreement aware group formation problem, including an optimization based formulation and a proof of its NP-Hardness. We utilize Projected Gradient Decent approach to develop an optimization framework for the problem and further propose a swapping alike algorithm with better scalability. Since our algorithm originates from generic optimization method, it does not depend on specific group recommendations semantics. Moreover, extensive experiments have been conducted on real-world datasets. The results show that the performances of our algorithms are close to optima and proper group formation before hand can lead to better group recommendation quality in different scenarios.



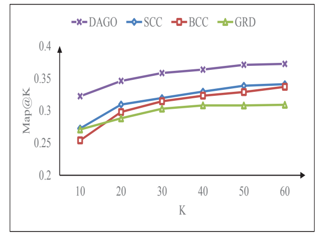
(a) Recall with $K \in (10, 60)$



(b) Map with $K \in (10, 60)$

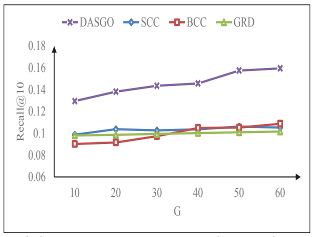


(c) Precision with $K \in (10, 60)$

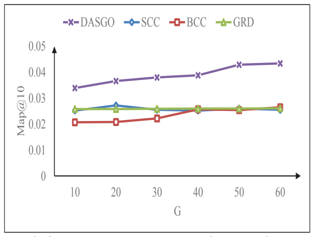


(d) Ndcg with $K \in (10, 60)$

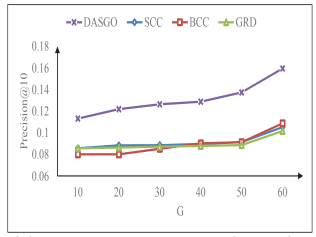
Figure 5: The item recommendation Performances with different K on ML-10M, $\omega = 1$, $G = 10$



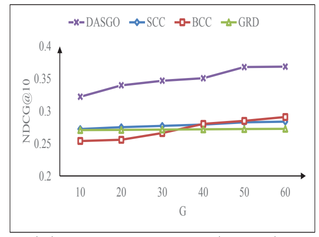
(a) Recall with $G \in (10, 60)$



(b) Map with $G \in (10, 60)$



(c) Precision with $G \in (10, 60)$



(d) Ndcg with $G \in (10, 60)$

Figure 6: The item recommendation Performances with different G on ML-10M, $\omega = 1$, $K = 10$

REFERENCES

[1] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, Gautam Das, and Cong Yu. 2009. Group Recommendation: Semantics and Efficiency. *Proc. VLDB Endow.* 2, 1 (Aug. 2009), 12.

[2] Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, Aristides Gionis, and Stefano Leonardi. 2012. Online Team Formation in Social Networks. In *WWW '12*. ACM, New York, NY, USA, 10.

[3] Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S. Modha. 2007. A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation. *J. Mach. Learn. Res.* 8 (Dec. 2007), 68.

[4] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[5] Jian Cheng, Ting Yuan, Jinqiao Wang, and Hanqing Lu. 2014. Group latent factor model for recommendation with multiple user behaviors. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*. 995–998.

[6] Andrew Crossen, Jay Budzik, and Kristian J. Hammond. 2002. Flytrap: Intelligent Group Music Recommendation. In *IUI '02*. ACM, New York, NY, USA, 2.

[7] Inderjit S. Dhillon. 2001. Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning. In *KDD '01*. ACM, New York, NY, USA, 6.

[8] Yi Liao, Wai Lam, Shoaib Jameel, Steven Schockaert, and Xing Xie. 2016. Who Wants to Join Me?: Companion Recommendation in Location Based Social Networks. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR '16)*. ACM, New York, NY, USA, 271–280.

[9] Anirban Majumder, Samik Datta, and K.V.M. Naidu. 2012. Capacitated Team Formation Problem on Social Networks. In *KDD '12*. ACM, New York, NY, USA, 9.

[10] Eirini Ntoutsi, Kostas Stefanidis, Kjetil Nørvåg, and Hans-Peter Kriegel. 2012. Fast Group Recommendations by Applying User Clustering. In *ER'12*. Springer-Verlag, Berlin, Heidelberg, 15.

[11] Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. 2001. *PolyLens: A Recommender System for Groups of Users*. Springer Netherlands, Dordrecht, 199–218.

[12] Tenda Okimoto, Nicolas Schwind, Maxime Clement, Tony Ribeiro, Katsumi Inoue, and Pierre Marquis. 2015. How to form a task-oriented robust team. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. 395–403.

[13] Piotr Faliszewski Piotr Skowron and Jerome Lang. 2014. Finding a Collective Set of Items: From Proportional Multirepresentation to Group Recommendation. *CoRR* abs/1402.3044 (2014).

[14] Elisa Quintarelli, Emanuele Rabosio, and Letizia Tanca. 2016. Recommending New Items to Ephemeral Groups Using Contextual User Influence. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 285–292.

[15] Balakrishnan Raju and Bhatt Rushi P. 2012. Real-time Bid Optimization for Group-buying Ads. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*. ACM, New York, NY, USA, 1707–1711.

[16] Syama Sundar Rangapuram, Thomas Bühler, and Matthias Hein. 2013. Towards Realistic Team Formation in Social Networks Based on Densest Subgraphs. In *Proceedings of the 22Nd International Conference on World Wide Web (WWW '13)*. ACM, New York, NY, USA, 1077–1088.

[17] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. *Introduction to recommender systems handbook*. Springer.

[18] Inbal Ronen, Ido Guy, Elad Kravi, and Maya Barnea. 2014. Recommending Social Media Content to Community Owners. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '14)*. ACM, New York, NY, USA, 243–252.

[19] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo. In *ICML '08*. ACM, New York, NY, USA, 8.

[20] Amirali Salehi-Abari and Craig Boutilier. 2015. Preference-oriented Social Networks: Group Recommendation and Inference. In *RecSys '15*. ACM, New York, NY, USA, 8.

[21] Laks V.S. Lakshmanan Senjuti Basu Roy and Rui Liu. 2015. From Group Recommendations to Group Formation. In *SIGMOD '15*. ACM, New York, NY, USA, 14.

[22] Hanhuai Shan and Arindam Banerjee. 2008. Bayesian Co-clustering. In *ICDM '08*. IEEE Computer Society, Washington, DC, USA, 10.

[23] Berkovsky Shlomo and Freyne Jill. 2010. Group-based Recipe Recommendations: Analysis of Data Aggregation Strategies. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 111–118.