Proceedings of the Seventeenth International AAAI Conference on Web and Social Media (ICWSM 2023)

Social Influence-Maximizing Group Recommendation

Yangke Sun[1], **Bogdan Cautis**[1,2], **Silviu Maniu**[1]
[1] Université Paris-Saclay, CNRS, LISN, Gif-sur-Yvette, France
[2] CNRS IPAL, Singapore
yangke.sun@lisn.upsaclay.fr, bogdan.cautis@lisn.upsaclay.fr, silviu.maniu@lisn.upsaclay.fr

Abstract

In this paper, we revisit the group recommendation problem, by taking into consideration the information diffusion in a social network, as one of the main criteria that must be maximised. While the well-known influence maximization problem has the objective to select k users (spread seeds) from a social network, so that a piece of information can spread to the largest possible number of people in the network, in our setting the seeds are known (given as a group), and we must decide which k items (pieces of information) should be recommended to them. Therefore, the recommended items should at the same time be the best match for that group's preferences, and have the potential to spread as much as possible in an underlying diffusion network, to which the group members (the seeds) belong. This problem is directly motivated by group recommendation scenarios where social networking is an inherent dimension that must be taken into account when assessing the potential impact of a certain recommendation. We present the model and formulate the problem of influence-aware group recommendation as a multiple objective optimization problem. We then describe a greedy approach for this problem and we design an optimisation approach, by adapting the top-k algorithms NRA and TA. We evaluate all these methods experimentally, in three different recommendation scenarios, for movie, micro-blog and book recommendations, based on real-world datasets from Flixster, Twitter, and Douban respectively. Unsurprisingly, with the introduction of information diffusion as an optimization criterion for group recommendation, the recommendation problem becomes more complex. However, we show that our algorithms enable spread efficiency without loss of recommendation precision, under reasonable latency.

Introduction

The explosive growth of e-commerce has led to the development of recommender systems, benefiting from a rich research literature in recent years (Aggarwal 2016; Ricci 2018). By mining the binary relationships between users and items (such as music, books, movies, news, tourist attractions, etc.), they can find items that users might be interested in from large amounts of data, generating personalized recommendations. Recommendation systems are nowadays an ubiquitous concept in Web applications, and have been applied successfully in many areas such as e-commerce, information retrieval, Web advertising, or social networking.

At the same time, with the development of social networks, a huge amount of information spreads online, generating significant research interest on the analysis of influence and information propagation, under the generic scope of *influence estimation* (Saito, Nakano, and Kimura 2008; Gomez-Rodriguez, Leskovec, and Krause 2012; Huang et al. 2019) and *influence (spread) maximisation* (Li et al. 2018; Kempe, Kleinberg, and Tardos 2003). The potential applications of influence maximization in social networks are quite diverse, including recommendation systems (Leskovec, Singh, and Kleinberg 2006), information diffusion (Matsubara et al. 2012), and viral marketing(Goldenberg, Libai, and Muller 2001; Jurvetson 2000). For example, a company may want to promote its new products, with a limited budget. It may hope that a small number of people can be selected to try the product for free. When the selected seed nodes are satisfied ("activated"), they will recommend (spread) the products to their families, friends, or colleagues, through the word-of-mount mechanisms embedded in social networking applications. This kind of scenario translates formally into the problem of finding the initial spread seeds (users) that could eventually influence the largest number of people within the diffusion network.

Among the popular recommendation scenarios, in recent years we have also seen the emergence of *group recommender systems* (Boratto and Felfernig 2018), where the items to be retrieved have as target not one individual but a group thereof. For example, such systems may recommend TV shows, movies, restaurants, music, trips, etc, to groups of friends that get together and require suggestions for a collective goal. In such contexts, as user preferences may be quite diverse even within a group (or even antagonistic), ideal recommendations are those that strike a balance between two orthogonal objectives, namely the overall *benefit* (the aggregated utility, a.k.a., the group's social welfare) and *fairness* (limiting as much as possible individual dissatisfaction).

In this context, it is therefore natural to revisit the group recommendation problem by assuming that the recommended information can also be spread in an underlying network, by word-of-mouth. More precisely, we consider in this paper a setting where the group members are assumed to be interconnected and are also members of a much larger social network, and we account for the effects of information diffusion in the group recommendation decision. Considering for example the movie recommendation problem in an online social network such as Facebook, when the target is a group of friends, it is reasonable to take into account the “bigger picture” and the potential implications of a recommendation not only for the group itself, but also for the social circles of the group’s members and beyond. Therefore, a good movie recommendation will not only satisfy the group being recommended, but also have the potential to spread further, from that group to other members of the social network, in the end leading to a much wider audience.

The main contributions of this paper are the following:

- We present a model that encompasses group recommendation and topic-aware influence, and we formulate *influence-aware group recommendation* as a multiple objective optimization problem.
- We describe a generic greedy algorithm, based on the model, to select the top-k items to be recommended to a given group.
- We consider optimizations over the generic algorithm, by adapting top-k algorithms such as NRA and TA (Fagin, Lotem, and Naor 2003), in order to improve the running time without loss of effectiveness.
- We evaluate our algorithms on three different recommendation scenarios, for movie, micro-blog and book recommendations to groups, over real-world datasets from Flixster, Twitter, Douban respectively, comparing them to several baseline methods using state-of-the-art ideas.

Related Work

There is a recent, extremely rich, and diverse research literature on recommender systems, motivated also by many industrial initiatives such as the Netflix prize challenge (Bell and Koren 2007). Generally, recommender systems can be divided into two main categories: personal recommender system and group recommender systems. For the former category, we refer the reader to general surveys such as (Ricci, Rokach, and Shapira 2011; Aggarwal 2016; Ricci 2018).

Group recommendation decisions (Boratto and Felfernig 2018) are more complex because of the different preferences between group members. Group members may perform different actions or have different views or preferences over to-be-recommended items (Elliot, Timothy, and Robin 2017). How to extract the common preferences of the group members and reduce the group members’ conflicts is therefore a key aspect of the problem. Obviously, there is no perfect, one-size-fits-all group recommendation approach and system, for all application scenarios, and one needs to adapt to the recommendation objective.

In the literature on group recommendation (Amer-Yahia et al. 2009), there are generally two categories of methods, based on either *preference aggregation* or *score aggregation*. The former makes recommendations based on the aggregated profile from all group members’ profiles (McCarthy and Anagnost 1998; Yu et al. 2006). The latter evaluate the score of recommendations for each group member respectively, then aggregate their recommendation results for the group (Baltrunas, Makcinskas, and Ricci 2010; Jameson and Smyth 2007). In order to make the recommendation result as adequate as possible for all group members, one needs to thoroughly consider the interactions of group members (Said, Berkovsky, and De Luca 2011).

But for ephemeral groups (Quintarelli, Rabosio, and Tanca 2016), who are constituted by people grouped together for the first time, the historical interactions may not exist, and one can only consider the aggregation of the individual preferences.

(O’connor et al. 2001) is one of the first research efforts on group recommendation, which introduced the Polylens system for movie recommendation to groups. In (Amer-Yahia et al. 2009), the authors analyzed the desiderata of group recommendation and proposed a formal semantics that accounts for both item relevance to a group and disagreements among group members. The works of (Ntoutsi et al. 2012; Amer-Yahia et al. 2009) also introduced how to select items that satisfy the group as a whole (social welfare). But unfairness (w.r.t. to certain members of a group) may still exist: the best recommendation satisfying most group members may still be disliked strongly by some users, as pointed out in (Qi et al. 2016). Towards fairness, some studies draw inspiration from game theory and voting theory (Yuan, Cong, and Lin 2014; Jameson and Smyth 2007; Elliot, Timothy, and Robin 2017; Baltrunas, Makcinskas, and Ricci 2010). In (Xiao et al. 2017), the authors consider balancing between the two objectives of group recommendation, namely the overall social welfare and fairness (w.r.t. individual preferences). Our work is based on the framework of fairness-aware group recommendation of (Xiao et al. 2017), and builds on it in order to account for topic-aware information diffusion for recommended items.

Among the research efforts that account for social networking aspects in group recommendation, we can mention approaches based on the network communities, such as (Sahebi and Cohen 2011), where the impact of similarity and interaction among members is considered, combined with collaborative filtering ideas, or on recommending semantic tags based on social relations (such as (Zheng et al. 2010)). In (Salehi-Abari and Boutilier 2015), the authors consider a group recommendation model in which preferences are correlated among people whom are connected in a social network, where preferences represent rankings over a set of options. Finally, there are some recent methods based on deep learning (Hu et al. 2014; Yuan, Cong, and Lin 2014), exploiting latent dependencies between user profiles and items.

There is also a rich literature on the analysis of influence and information propagation in social networks, under the generic scope of *influence estimation* (Saito, Nakano, and Kimura 2008; Gomez-Rodriguez, Leskovec, and Krause 2012; Huang et al. 2019) and *influence (spread) maximisation* (Li et al. 2018; Kempe, Kleinberg, and Tardos 2003). The problem of influence maximization was first formulated in (Kempe, Kleinberg, and Tardos 2003) as a discrete optimization problem, and proven NP-hard problem under certain propagation models. Many research ideas and algorithmic solutions followed, in general focusing on approximations that have a reasonable trade-off between effectiveness and efficiency (Arora, Galhotra, and Ranu 2017). Among the most common applications of information diffusion in social networks we have recommender systems (Leskovec, Singh, and Kleinberg 2006) and viral marketing(Goldenberg, Libai, and Muller 2001; Jurvetson 2000).

What distinguishes our research from the majority of state-of-the-art research works in group recommendation is the fact that we integrate as a first-class objective the spread of information: we must adapt the recommendation strategy in order to also maximize the social impact of the recommended items. In that regard, the research efforts that are closest to our focus are (Christensen and Schiaffino 2014; Yin et al. 2019), which propose approaches to generate recommendations for groups on the basis of social factors extracted from a social network, factors that determine the in-group dynamics and the group's overall satisfaction for recommended items. In particular, in the recent work of (Yin et al. 2019), the underlying idea is that an ideal group recommender system should be able to accurately learn from logs of item adoptions not only the preferences of individuals, but also the preference aggregation strategy within a group that may be formed in an ad-hoc manner; in a social networking context, this strategy should account for the social influence of group members within the group. Then, (Yin et al. 2019) relies on an attention mechanism in order to learn each user's potential influence within different groups. Our research is orthogonal to these works, as we focus on the implications of information diffusion and influence outside the group, within the larger social network.

Preliminaries

We first formally define topic-aware influence maximization and fairness-aware group recommendation. We then propose our model for influence-aware group recommendation, by combining these two ingredients.

Diffusion networks and influence maximization. We model the social (diffusion) network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$, where $\mathcal{V}$ is a set of users, $\mathcal{E}$ contains all edges connection the users and $\mathcal{P}$ is a probability function on $\mathcal{E}$, with the semantics that the information will be propagated along an edge according to the probability of that edge, as specified by $\mathcal{P}$. Information spread (a.k.a. influence spread or influence cascades) is therefore captured by a stochastic process, following certain *propagation models*.

Influence maximization. In a diffusion network, starting from an initial group of nodes G – the seed set – that represent the initially activated nodes before the influence process is initiated, $Spread(G)$ denotes the random variable describing the expected size of the spread initiated from G. The problem of *influence maximisation* can be defined generically as follows: under a certain propagation model, select a set of seed nodes G, of size at most k, such that the expected spread of influence cascades starting from G (or the expected number of activated nodes) is maximized.

Independent Cascades diffusion model. We discuss next the most well-known propagation model, Independent Cascades (IC), which is the one we consider in our paper. We refer the interested reader to (Kempe, Kleinberg, and Tardos 2015) for a broader discussion on diffusion models. IC is a stochastic propagation model, in which each node of the social network can have two states: active and inactive. Active nodes will activate each inactive neighbor node with a certain probability, as follows. In the initial state $t = 0$, only a certain number of nodes from a seed set S are set to the active state. At any time step $t = i$, all nodes that transitioned from the inactive state to the active state at $t = i - 1$ will have one chance to activate all their inactive neighbor nodes, succeeding in doing so with the probability associated to the respective outgoing links. When there are no nodes remaining in the network with the ability to activate other nodes, the propagation process ends. The spread corresponds to the number of the activated nodes.

Topic-aware influence. As an extension to IC, we consider the topic-aware model proposed initially in (Barbieri, Bonchi, and Manco 2013), which takes into consideration the topical description of the information being diffused. More precisely, we will assume that each edge in the diffusion graph is a diffusion medium for information pertaining to a certain number d of topics: namely, $(u, v) \in \mathcal{E}$ is associated with a topic spreading-weight vector $\mathbf{p}_{u,v} = \left(p^1_{u,v}, p^2_{u,v}, ..., p^d_{u,v}\right)$, where $p^z_{u,v}$ is the weight associated to topic z, denoting the activation probability for user v if activated by user u under topic z. $p_{u,v}$ is normalised to sum up to 1. Then, given a topic distribution vector $\overrightarrow{\gamma}$ for the information that is diffused, $\overrightarrow{\gamma} = \left(\gamma^1, \gamma^2, ..., \gamma^d\right)$ such that $\sum_{1 \le i \le d} \gamma^i = 1$, for each edge (u, v), the propagation probability along that edge w.r.t. the topic distribution $\overrightarrow{\gamma}$ is:

$$p_{u,v}(\overrightarrow{\gamma}) = \langle \mathbf{p}_{u,v}, \overrightarrow{\gamma} \rangle = \mathbf{p}^{\top}_{u,v} \overrightarrow{\gamma},\ 0 \le \mathbf{p}_{u,v} \le 1 \quad (1)$$

It is this propagation probability that will be used in an *IC topic-aware diffusion process*, for edge (u, v), as described before. We denote now by $Spread(G|\overrightarrow{\gamma})$ the random variable describing the expected size of the spread initiated from G when $\overrightarrow{\gamma}$ is the description of the conveyed information.

Problem 1 (Topic-Aware Influence Maximization) *Given a topic-aware network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a query $\mathcal{Q} = (\overrightarrow{\gamma}, k)$, find a seed set $G^* = \arg\max_G Spread(G|\overrightarrow{\gamma})$, where $G \subset \mathcal{V}$, $|G| = k$.*

Greedy algorithm and Monte Carlo simulations. As the objective function in influence maximisation is *monotone* and *submodular* (Kempe, Kleinberg, and Tardos 2015), the general approach for finding an approximate solution is based on the greedy algorithm, selecting at each step a new seed node – the one having the largest marginal gain on expected spread. Since computing the expected spread is #P-hard (Kempe, Kleinberg, and Tardos 2015) – and this is straightforwardly also the case for topic-aware version $Spread(G|\overrightarrow{\gamma})$ (Barbieri, Bonchi, and Manco 2013) – many research works on influence maximisation rely on approaches involving *Monte-Carlo simulations*, to obtain an approximate influence spread efficiently at any given step in the greedy algorithm for seed selection (see (Arora, Galhotra, and Ranu 2017) and the references therein). In short, given a seed node, one can simulate r random cascades from it and average the number of influenced nodes, which can lead to a provably tight approximation of spread from that seed. For instance, when simulating a cascade in the IC model, for a newly activated node u and an inactive outgoing neighbour v, one needs to compare the activation probability $p_{u,v}$ with a number generated uniformly at random in $[0, 1]$ (a coin toss), activating v if this random number is lower. In the topic-aware setting, simulations are performed similarly, with the difference that $p_{u,v}(\overrightarrow{\gamma})$ is used for the edge (u, v), when simulating the diffusion of $\overrightarrow{\gamma}$ over it.

Group recommendation with social welfare and fairness. Following the most common dimensions of models for group recommendation, we assume that users have access – or are recommended – a set of items from a domain of items $\mathcal{I}$, and each user establishes a value of *relevance* for each item, corresponding to their preferences.This pairwise relevance can then be normalized as $\text{rel}(u, i) \in [0, 1]$. The utility function $\text{Utility}(u, I)$ – of user u with respect to a set of items $I \subseteq \mathcal{I}$ – is the sum of normalized relevance scores of users u for items i, i.e. $rel(u, i)$ scores:

$$\text{Utility}(u, I) = \min(1, \sum_{i \in I} \text{rel}(u, i)) \tag{2}$$

Then, for evaluating the overall utility of a group, one can aggregate the individual utility values into a *social welfare* score, and further consider *fairness* as an indicator for the imbalance between the individual users' utility values. More precisely, when we have users belonging to a group $G \subset \mathcal{V}$, we can aggregate the individual utility values, $\text{Utility}(u, I)$ from equation (2), to measure the extent of the *social welfare* of the group G in relation to item set I, as follows:

$$\text{SW}(G, I) = \frac{1}{|G|} \sum_{u \in G} \text{Utility}(u, I). \tag{3}$$

Complementary, a measure of *fairness* will assess how satisfied are all the members of the group with the item selection I, as a function of all $Utility(u, I)$ values. While there are several alternative formulations for fairness, the common aim is to minimize the utility gap between group members (Amer-Yahia et al. 2009). In the following, we will adopt the *Least Misery* formulation, defined as the minimal utility of a user in the group, but our results can be easily adapted for other fairness formulations:

$$F_{LM}(G, I) = \min_{u \in G}(\text{Utility}(u_1, I), \text{Utility}(u_2, I), .., 1). \tag{4}$$

So the group's fairness is directly tied to the minimal utility a group member would get from the items I. Based on these ingredients, we can now introduce the *group recommendation problem*, as considered recently in (Xiao et al. 2017):

Problem 2 (Group Recommendation) *Given a user group G and a set of items $\tilde{I} \subseteq \mathcal{I}$ up for recommendation, recommend a set $I \subseteq \tilde{I}$ of k items s.t. social welfare $SW(G, I)$ and fairness $F(G, I)$ are* maximized.

Influence-aware Group Recommendation

Based on the preliminaries of the previous section, we describe next the recommendation problem we consider. Recall that in group recommendation with fairness and social welfare the objective is to select an item set I of a given size, from a large number of items $\tilde{I}$, so that the group members are "satisfied". In topic-aware information propagation, the objective is to select a set of users from the social network who can influence the largest number of people in the network. In our influence-aware group recommendation setting, the goal is to combine these two perspectives, i.e., assuming that the group members are part of a social network where information propagates beyond the group's social scope.

Associating topics to edges, users, and items. As explained before, our framework assumes that the diffusion network and information being spread have a topical model, with topics are derived from the items being diffused, as a d-dimensional space, i.e. $i \in \mathbb{R}^d$. We next describe how this is extended to encompass users and item relevance.

Edges. Assuming that information spreading in the network pertains to these d topics, recall that each edge $(u, v) \in \mathcal{E}$ has a topic spreading-weight vector $\left(p^1_{u,v}, p^2_{u,v}, ..., p^d_{u,v}\right)$, where $p^z_{u,v}$ is the weight on topic z.

Users. Similarly, for each user $u \in \mathcal{V}$ and topic z, $\text{rel}(u, z)$ expresses the propensity of an user u to influence neighboring nodes and generate spread on a specific topic z in which they are interested and have authority on. If u has a spreading-weight $p^z_{u,v}$ to his neighbor $v \in C(u)$ on topic z, it is reasonable to assume that his interest and authoritativeness on topic z must be higher than $p^z_{u,v}$. Therefore, in our model, $rel(u, z)$ can be determined as follows:

$$\text{rel}(u, z) \in [\max\{p^z_{u,v}, \forall v \in C(u)\}, 1] \tag{5}$$

and overall each user profile can be described by the relevance vector $\text{rel}(u) = (\text{rel}(u, 1), \dots \text{rel}(u, d))$.

Items. Generally speaking, the essence of item i is a topical distribution vector $\overrightarrow{\gamma_i} = \left(\gamma^1_i, \gamma^2_i, ..., \gamma^d_i\right)$ in which γ^z_i denotes the probability that item i belongs to topic z, such that $\sum_{z=1}^d \gamma^z_i = 1$. The topical distribution $\overrightarrow{\gamma_I}$ of a set of items I is simply obtained through an aggregation over each topic and upper bounded by 1. With $rel(u, z)$ and $\overrightarrow{\gamma_I}$, we can determine the relevance of candidate items I to each user $rel(u, I)$, to bring the reasoning from a topic-wise one to a set-of-items-wise one, by the scalar product between the d-dimensional vectors $rel(u)$ and $\overrightarrow{\gamma_I}$. We can then correspondingly generalise the $Utility$, SW, and F functions that are common in group recommendation models.

Problem 3 (Influence-Aware Group Recommendation) *Given a social network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$ over which a topic-aware influence model is defined, given a group of users $G \subseteq \mathcal{V}$ and a set of items $\tilde{I} \subseteq \mathcal{I}$, with each item $i \in \tilde{I}$ represented as a topical distribution $\overrightarrow{\gamma_i} = \left(\gamma^1_i, \gamma^2_i, ..., \gamma^d_i\right)$ in which γ^z_i denotes the probability that i belongs to topic z, select and recommend to G a set $I \subseteq \tilde{I}$ of k items, with overall (set-wise) topical distribution $\overrightarrow{\gamma_I}$ such that the social*

Algorithm 1 Greedy Influence-Aware Group Recommendation

Require: Network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; group $G \subset \mathcal{V}$; items $\tilde{I}$; budget k; scalarization parameters α and β.
1: Initialize $I = \emptyset$
2: **while** $|I| \leq k$ **do**
3: $I := I \cup argmax_{j \in \tilde{I} \setminus I} \; [\alpha \cdot SW(G, I \cup \{j\}) + \beta \cdot F_{LM}(G, I \cup \{j\}) + (1 - \alpha - \beta) \cdot Spread(G|\overrightarrow{\gamma}_{I \cup \{j\}})]$
4: **end while**;
5: **return** I

welfare $SW(G, I)$, fairness $F(G, I)$, and topic-aware spread Spread$(G|\overrightarrow{\gamma_I})$ are maximized.

Note that an essential building block for evaluating a topic-aware influence maximization is to compute the topic-aware influence spread of the topics induced by the item set I, $Spread(G|\overrightarrow{\gamma_I})$. The problem we study is therefore "reverting" the classic influence maximisation perspective, in the sense that, instead of looking for a set of spread seeds from which to diffuse a given piece of information, we are looking for the right piece of information to be diffused by a given set of spread seeds (the group); we focus on *what* is diffused (items) rather that from *whom* the diffusion process is initiated. The items recommended to the group and that will be spread from it should be consensual (a good match) to the group, so that the process can be seen as successful.

Finally, with the topic-aware network $\mathcal{G}(\mathcal{E}, \mathcal{V}, \mathcal{P})$, we can compute the expected spread within $\mathcal{G}$ for a set of items I, from a group G, namely $Spread(G|\overrightarrow{\gamma_I})$, by MC simulations.

Greedy Algorithm

The particular multi-objective optimization we consider in this paper is clearly NP-hard, since we bring in an orthogonal objective with respect to the ones considered initially in (Xiao et al. 2017), namely in our problem we want to maximize jointly social welfare $SW(G, I)$, fairness $F_{LM}(G, I)$, and influence spread $Spread(G|\overrightarrow{\gamma_I})$. In practice, one approximate approach for solving such a problem is based on *scalarization* (Eichfelder 2009), which allows us to compute a solution that is Pareto-optimal (in other words, no one dimension "wins" out), based on a convex sum with weights assigned to each objective. Under this approach, the problem becomes one of maximizing the following objective:

$$\alpha \cdot SW(G, I) + \beta \cdot F_{LM}(G, I) + \\ +(1 - \alpha - \beta) \cdot Spread(G|\overrightarrow{\gamma_I})$$
$$\text{for } G \subset \mathcal{V},\ \tilde{I},\ I \in \tilde{I},\ |I| = k,\ 0 < \alpha, \beta, \alpha + \beta < 1 \quad (6)$$

This program can be solved approximately by a greedy approach, selecting at each step the item having the highest marginal potential, i.e., achieving the highest combination of fairness, social welfare, and influence when added to the current solution. The flow of this greedy algorithm is given in Alg. 1. Note that, when estimating the spread for a set of items, we "bundle" them into one aggregated item to be spread, by averaging the composing items over each topic.

Algorithm 2 Compute user-item relevance

Require: Given user $u \in G$; set of targeted items $I \cup \{j\}$; topic-unit items: γ_j, for $1 \leq j \leq d$
Ensure: $\text{rel}(u, I \cup \{j\})$
1: **for** $j \leftarrow 1$ to d **do**
2: compute (or retrieve if already computed) $rel(u, \gamma_j)$
3: **end for**
4: **return** $\text{rel}(u, I \cup \{j\}) = (I \cup \{j\}) \cdot (\text{rel}(u, \gamma_1), \ldots, \text{rel}(u, \gamma_d))^\top$

The complexity of the algorithm depends on the one of each dimension in the scalarization program. For d topics, the social welfare and fairness computations each take $\mathcal{O}(d \times |I| \times |\mathcal{V}|)$. The spread computation depends on R Monte-Carlo sampling rounds shown in the preliminaries, each taking linear time in the graph and the dimensionality, so $\mathcal{O}(d \times R \times |\mathcal{E}|)$; this is repeated k times. Overall, the complexity is therefore $\mathcal{O}\left(d \times |I| \times |\mathcal{V}| + k \times (d \times R \times |\mathcal{E}|)\right)$.

Exploiting item similarity. The model and algorithm described up to this point represent a first solution for the problem of diffusion-aware group recommendation, and we detail experimental results for it in the experiments section.

At the core of our greedy algorithm (Algorithm 1) is the correlation $rel(u, I \cup \{j\})$ between each user in G and each potential new partial solution (a subset of $\tilde{I}$), in our case each such partial solution $I \cup \{j\}$ being described by a topical distribution. The naive approach to computing these relevance scores requires us to look at every pair of items and users, for each step, and to recompute the relevance scores, welfare, and fairness. When we are dealing with large groups and many items, this may incur a high cost at query time. It is therefore important to optimise and make this recurrent step less computationally intensive.

A first optimisation idea we employ here (Algorithm 2) is the following : for a given user $u \in G$, we will only compute the relevance of the d unit vectors of topical distribution. For all other items or sets thereof, we then compute their relevance for u by applying a linear combination of the unit vectors' relevance. This optimisation can be easily "plugged" in Algorithm 1: both social welfare(denoted as $SW(G, I)$) and fairness (denoted as $F(G, I)$) use the overall utility for the group's members, given the recommendation I: $Utility(u, I)$, $\forall u \in G$, $\forall I$. The utility function $Utility(u, I)$ of user $u \in G$ is a function of the relevance of the individual recommended items $rel(u, i)$, now computed approximately based on the topic-unit items. This optimisation only works for linearly aggregated utility functions, such as the one in Equation (2).

Alternative for effectiveness. As we are studying a multi-objective optimization, it is important to compare our methods with multi-objective optimization solutions that trade efficiency for effectiveness, providing what could be seen as an *almost golden standard*. We consider the Multiplicative-Weight-Updates (MWU) algorithm (Udwani 2018), which focuses on the generic problem of multi-objective maximization of monotone submodular functions subject to car-

Algorithm 3 MWU (Udwani 2018)

Require: $\delta, T = \frac{2\ln m}{\delta^2}, \lambda_i^1 = 1/m\ \tilde{f}_i(.) = \frac{f_i(.|S_1)}{V_i - F_i(S_1)}$;

1: **while** $1 \leq t \leq (T)$ **do**

2: $\quad g^t(.) = \sum_{i=1}^m \lambda_i^t \tilde{f}_i(); \quad X^t = A(g^t, k_1)$

3: $\quad m_i^t = \tilde{f}_i(X^t) - \alpha; \quad \lambda_i^{t+1} = \lambda_i^t(1 - \delta m_i^t)$

4: **end while**

5: **return** $x_2 = \frac{1}{T}\sum_{t=1}^T X^t$

dinality constraints, which is precisely our setting.

In short, (Udwani 2018) uses a three-step algorithm, whose aim is to find the best weights in a scalarization approach, but using slightly different functions. These steps, achieving a randomized $(1 - 1/e) - \epsilon$ approximation, are:

1. Finding an initial feasible vector – in our case, these would be initial candidate recommendations.
2. *Multiplicative Weight Updating* (MWU), in which the weights assigned to each dimension get updated depending on the variation of the benefit (Alg. 3)
3. The rounding stage: as the best vector found may contain fractional values, we round it to the nearest topic vector.

In our MWU implementation, δ is a subtractive term in the approximation guarantee, and λ is a parameter of the functions. Step 1 is achieved by running our greedy algorithm, giving the initial set of k items. In step 2, we notice that line 2 of MWU allows any optimization algorithm. Hence, we simply plug-in our greedy algorithm using the scalarization technique presented before. Evidently, this means the multi-objective algorithm will always be significantly slower than our approach; however, its potential to find better values for α and β makes it an interesting alternative for comparing effectiveness. The parameters to be set manually are δ, in our case is 0.2, and m (the number of functions) is 3.

MWU outputs an average from the best vectors found in every step. In our case, taking a simple average would not give us a real item. Instead, for step 3 (rounding), we computed the nearest real item – in terms of d-dimensional euclidean distances – to the averaged vector.

Top-k Algorithms

Our *on-the-fly* greedy recommendation algorithm requires no precomputation and leads to highly relevant results for any given group. However, its complexity and execution time are too large for an online setting. Inspired by well-known top-k algorithms, such as Threshold Algorithm (TA) and No Random Access Algorithm (NRA) (Fagin, Lotem, and Naor 2003), we consider next a setting where some of the item-group relevance score ingredients are *precomputed* and *sorted*, based on which recommendations at query time can be rendered significantly faster. As we do not need to compute the exact items' scores, but mainly to select which ones should be recommended for a given group, we first adapt our model to meet the requirements of top-k algorithms. We start with the observation that, given the monotone submodularity of the objectives, we can rely on pre-computed item-user relevance and spread scores. First, note that *social welfare* (SW) and *fairness* (F) can be computed by a linear combination of each individual score $rel(u, i)$, as defined in Section . In order to compute them at query time more easily, for each user, we assume to make available a sorted list of items by relevance score.

We also pre-compute for each user a list of per-item lower and upper bound scores for individual spread of items. These will be ordered by the upper bounds, and can obtained based on Monte-Carlo simulations, as shown in (Zhou et al. 2015).

So, for a given query G, the computation starting point at query time will be, for each user in G, two lists consisting of the candidate items I ordered by rel scores and $spread$ lower and upper bound scores (denoted $\sigma^+(u|\overrightarrow{\gamma_i})$ and $\sigma^-(u|\overrightarrow{\gamma_i})$).

By design, the generic NRA and TA algorithms work by accessing sorted lists of partial scores, from which the overall score can be obtained by a monotone function thereof. In our case, these will correspond to a linear combination of rel and $spread$ scores, for all users inside the seed group G. The pool of candidate items (denoted here as B) scored by the linear combination of partial scores, is maintained sorted by lower bound on the overall score; the item at some rank r in B is denoted B_r. A run stops when in B there are k candidate items whose lower bound is higher than the upper-bound of all other items (including unseen ones) – this is denoted as the *early-termination condition*.

Recall that social welfare and fairness are based on a linear combination of $rel(u, i)$ scores, $\forall u \in G, i \in I$, as:

$$\mathrm{SW}(G, i) = \frac{1}{|G|}\sum_{u \in G} rel(u, i). \tag{7}$$

$$\mathrm{F}_{LM}(G, i) = \min_{u \in G}\{rel(u, i)\}. \tag{8}$$

Similarly, based on the individual $spread(u, i)$ score intervals, $\forall u \in G, i \in I$, the group's spread objective has an upper-bound given by the sum of all users' upper bound scores, and a lower bound given by the maximal individual lower bound on spread, as follows:

$$spread^+(G, i) = \sum_{u \in G}\{\sigma^+(u|\overrightarrow{\gamma_i})\} \tag{9}$$

$$spread^-(G, i) = \max_{u \in G}\{\sigma^-(u|\overrightarrow{\gamma_i})\} \tag{10}$$

The NRA algorithm accesses sequentially all lists in parallel until the early termination condition is met (Alg. 4). TA allows random accesses to fill in the scores (Alg. 5).

In both algorithms, for each item i retrieved by a sorted sequential access at some iteration, the overall score $S(i)$ will be combined from $2 \times |G|$ sorted lists of items (denoted L). L includes for each user $u \in |G|$ the list containing $rel(u, i)$ scores (for building SW and F) and the list of pairs of upper / lower bound spread scores $(\sigma^-(u|\overrightarrow{\gamma_i}), \sigma^+(u|\overrightarrow{\gamma_i}))$.

More precisely, from L, we would obtain the final score estimations in two parts, as follows. For the part using rel:

$$\alpha \times \mathrm{SW}(G, i) + \beta \times \mathrm{F}_{LM}(G, i) =$$

$$= \alpha \times \frac{1}{|G|}\sum_{u \in G} rel(u, i) + \beta \times \min_{u \in G}\{rel(u, i)\} \tag{11}$$

Algorithm 4 NRA-based recommendation algorithm

Require: result size k, $2 \times |G|$ lists L
Ensure: B has best k items ordered by lower bound score

while $S^-(B_k) < T$ and not at the end of the lists L **do**
  for each $j \in \{1, \dots, 2 \times |G|\}$ **do**
    $(i, p_j(i)) \leftarrow getNext(L_j), head_j \leftarrow p_j(i)$
    if i not yet in B **then**
      compute $S^-(i), S^+(i)$, based on $p_j(i)$
      add $\{i, S^-(i), S^+(i)\}$ to B
    else
      update $S^-(i), S^+(i)$, based on $p_j(i)$
    end if
    update $S^+(d)$ for all other items d in B and not yet seen in L_j, based on the new value for $head_j$
    update $S^+(*)$ and T
    maintain B sorted by lower bound score (S^-)
  end for
end while
return $B_1, \dots B_k$

and for the spread part, lower or upper bound:

$$(1 - \alpha - \beta) \times spread^+(G, i) = \\ = (1 - \alpha - \beta) \times \sum_{u \in G} \{\sigma^+(u|\overrightarrow{\gamma_i})\} \quad (12)$$

$$(1 - \alpha - \beta) \times spread^-(G, i) = \\ = (1 - \alpha - \beta) \times \max_{u \in G} \{\sigma^-(u|\overrightarrow{\gamma_i})\} \quad (13)$$

We assume a sorted access function $getNext(L_j) \rightarrow (i, p_j(i))$ which retrieves the item Id i of the next best item in the list L_j and its partial score p_j, which can be $rel(u, i)$ or the lower-upper bound pair on spread estimation.

As soon as we access a new item via a sequential access in one of the lists, we can establish an overall score lower bound $S(i)^-$, as well as an overall score upper bound $S(i)^+$. The purpose of these bounds is to establish the early termination condition: as soon as the lower bound of the kth item in the candidate list B is lower than the upper bound of all other possible items outside the top-k, the algorithm stops.

In NRA, the overall score bounds are obtained as follows:

1. $S^-(i)$ uses 0 for all the lists L_j in which the item i has not been encountered yet, and $p_j(i)$ for the other lists.
2. $S^+(i)$ uses $head_j$ for all lists L_j in which item i has not been encountered yet, and $p_j(i)$ for the other lists. $head_j$ denotes the current *upper bound* on the remaining scores in list L_j (i.e., the value at the current head of list L_j).

It is convenient for the algorithmic construction to assume that among the previously encountered items there exists a *virtual item* $*$, representing all unseen items, whose corresponding upper bound score $S^+(*)$ is obtained by plugging in the $head_j$ values for all list L_j – this, in other words, represents the hypothetical best score for any not yet encountered item. The termination condition is $S^-(B_k) < T$, for

$$T = \max\left(S^+(*), \left\{S^+(i) | i \in B \setminus \{B_1, \dots B_k\}\right\}\right) \quad (14)$$

Algorithm 5 TA-based recommendation algorithm

Require: result size k, $2 \times |G|$ lists L
Ensure: B has best k items ordered by lower bound score

while $S^-(B_k) < T$ and not at the end of the lists L **do**
  for each $j \in \{1, \dots, 2 \times |G|\}$ **do**
    $(i, p_j(i)) \leftarrow getNext(L_j), head_j \leftarrow p_j(i)$
    if i not yet in B **then**
      compute $S^-(i), S^+(i)$, based on newly seen $p_j(i)$ and scores $p_l(i)$ obtained by *random accesses* (lookup by i) in all other lists L_l, $l \neq j$
    end if
    update $S^+(*)$ and T
    maintain B sorted by lower bound score (S^-)
  end for
end while
return $B_1, \dots B_k$

Our NRA adaptation, which only uses sequential accesses over the lists L, has the flow outlined in Alg. 4. The TA adaptation works in a similar manner to the NRA one, but has as main difference the fact that it is also allowed random accesses: whenever a new item i is encountered when sequentially accessing some list L_j, getting $L_j(i)$, all other lists $L_l, l \neq j$, are accessed to obtain the actual scores $L_l(i)$ as well. So TA requires both a sequential and random (lookup by item Id) access to lists. Its flow is outlined in Alg. 5.

As a final note here, our TA-based adaptation should be preferred whenever random accesses to pre-computed results do not incur a high cost, for instance when stored in a flash drive. Whenever randomly accessing data is significantly more costly than sequentially accessing it, the NRA-based adaptation should be preferred instead.

Experimental Evaluation

We conducted experiments to assess the performance of our solutions (greedy and top-k algorithms), in scenarios of movie, micro-blog, or book recommendation to groups. For evaluation, we are interested in 4 main dimensions: overall benefit (overall score following scalarization), recommendation precision (to be defined shortly), spread efficiency, and also the execution time required to select and recommend data items with different scalarization parameters.

As mentioned, the potentially-high execution time of the greedy approach led us to design two alternative top-k methods, based on precomputed individual scores, by adapting the TA/NRA algorithms. With respect to them, we also conducted focused experiments on recommendation execution time and the early-stopping capacity (number of accesses that were performed by TA or NRA).

Movie recommendation in Flixster. We rely on a Flixster dataset from (Jamali and Ester 2010). Flixster was a social movie platform allowing users to share movie ratings, discover movies, and meet others with similar tastes. This dataset has users links constituting a social network and timestamped action logs of user's movie ratings. We completed this dataset by crawling the IMDB API in order to

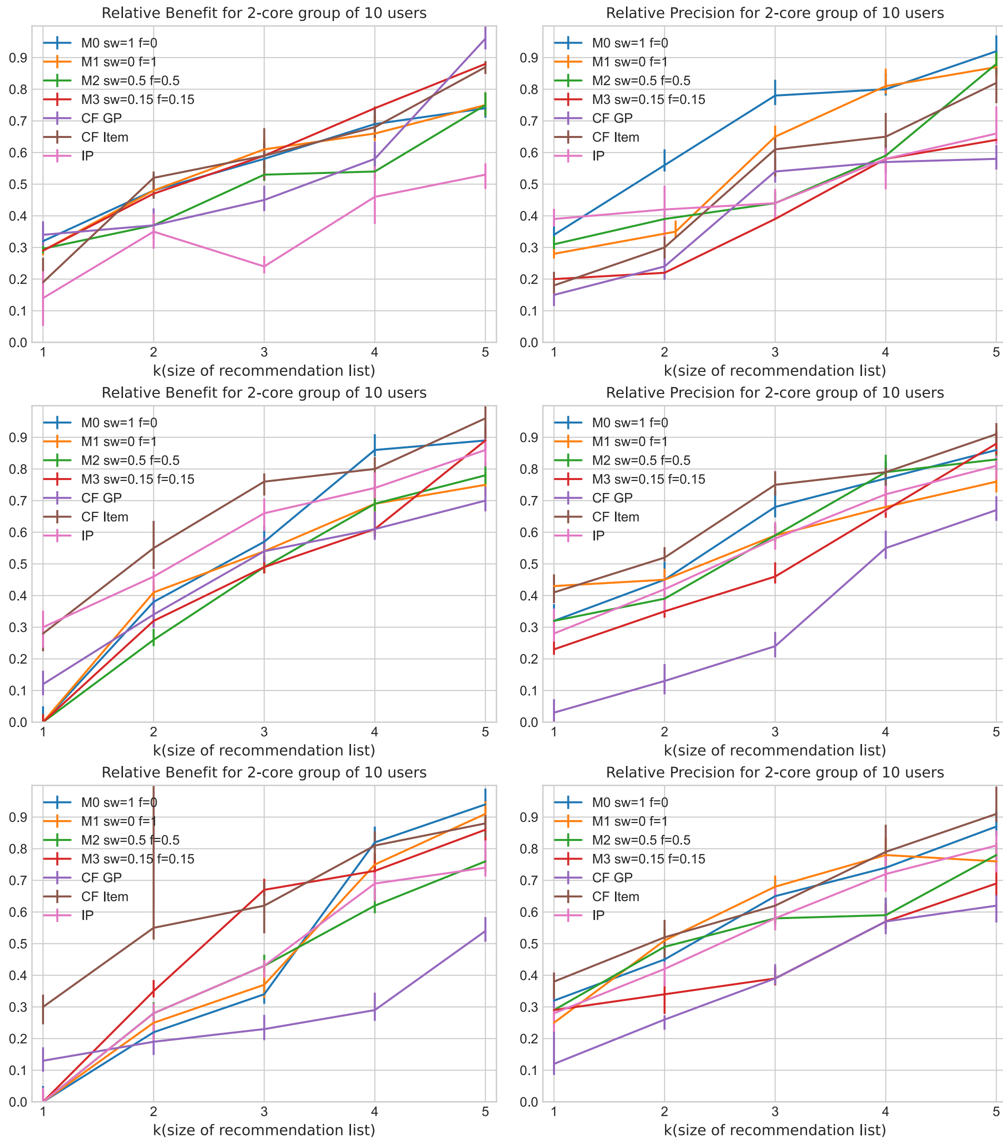

Figure 1: Experimental results (benefit, precision) for 2-cores of 10 users, with confidence intervals (Flixstr, Twitter, Douban).

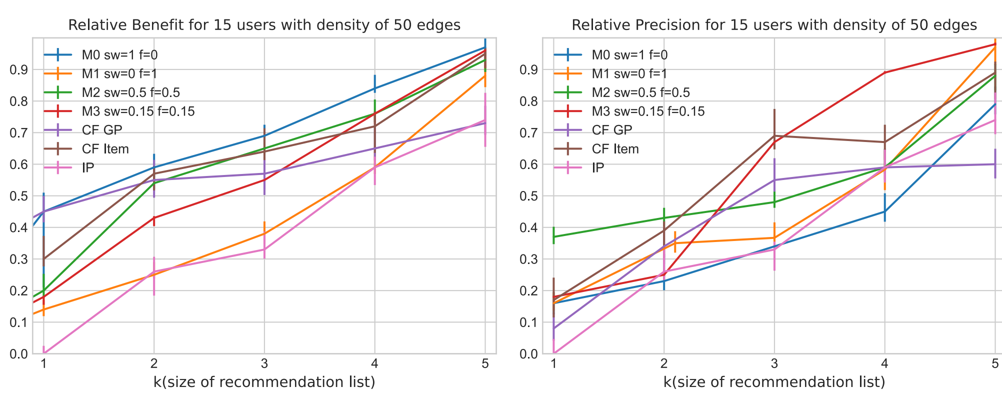

Figure 2: Experimental results (benefit, precision) for 15 users and density of 50 edges, with confidence intervals (Flixstr).

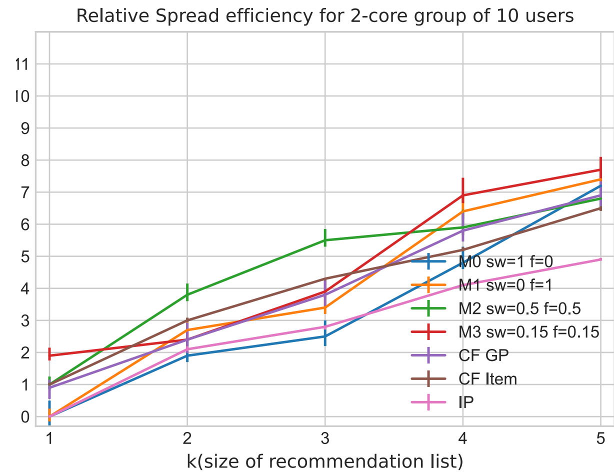

Figure 3: Spread efficiency (2-cores, 10 users, Flixstr).

have the genres of movies. Each movie description includes one or more categories out of 29 possible IMDB genres, obtaining a topical distribution vector $(\gamma_i^1, \gamma_i^2, ..., \gamma_i^d)$ whose sum of values equals 1 after normalization. A necessary step over this dataset was to complete it with influence probability values among nodes. We followed here a simple logic, assuming that common movies of connected nodes u and v are good indicators for the social influence over link (u, v).

1. We removed any social link (u, v) with less than 5 common movies between u and v.
2. Then, for a link (u, v), v is influenced by u on common movies m s.t. $timestamp(u, m) < timestamp(v, m)$ and $|rating(u, m)$ - $rating(v, m)| \leq \delta = 3$.
3. For each topic z, the influence probability $p^z_{u,v}$ between u and v is given by the ratio between the number of common z-movies rated first by u and then by v with similar scores and the number of common z-movies.

The resulting Flixster diffusion network has 35,735 nodes, 183,898 edges, 23,750 movies, 314,5202 ratings, an average degree of 10.29, and a maximal degree of 309.

Tweet recommendation in Twitter. We have built a second dataset for tweet recommendation to groups, by crawling the Twitter platform via the Twitter API. This dataset contains a social network (nodes representing Twitter users, edges between them corresponding to follower-followee relationships) and tweets (retweets) posted by the network's members. Therefore, items are now tweets, which may be posted by one or several users (via retweeting), ratings are now binary values (tweeted or not).

Over this dataset, considering that the items to be recommended are tweets, we must devise the topical distribution thereof. For that, similar to (Shin et al. 2015), we model each word appearing in tweets by a topical vector by using the word embedding technique *word2vec* (Mikolov et al. 2013). We then cluster these vector representations of tweet text into a predefined number of topics with the k–means algorithm. Using the cluster information with the aid of Silhouette Coefficient (Aranganayagi and Thangavel 2007), we then create a histogram of clusters for each tweet. In this way each tweet is encoded as a distribution over a fixed number of topic. For the best balance between quality and number of clusters, we vary the number of cluster between 1 and 50 and get the best coefficient of 0.11, corresponding to 36 clusters. We can then compute the influence probability values among nodes in the same way as for Flixster.

The resulting Twitter diffusion network has 67,598 nodes, 349,630 edges, 98,560 tweets, 9,358,790 ratings, an average degree of 24.87, and a maximal degree of 2,927.

Book recommendation in Douban. We describe next the Douban dataset, collected by (Zhao and Ji 2018). Douban is a popular Chinese social network that allows users to rate and publish content related to movies, books, music, and events. Focusing on books, the dataset includes rating information along with social links, and users can join a variety of groups. Moreover, Douban has recently added a geo-location feature. We exploited the groups information and the geo-location to extract a cohesive network of diffusion, in order to test our algorithms on a network that is denser than the previous two. As before, we used common book ratings between users to establish the topic-wise influence probability vectors. In Douban, we have in total 897 tags (tags are originally generated by users). We first set a threshold corresponding to a minimum number of books as 100, and we use *word2vec* to cluster the remaining 626 tags to 59 clusters, corresponding to the best Silhouette Coefficient (Aranganayagi and Thangavel 2007). The resulting dataset has 10,353 users, 17,454 books, 2,356 groups, 143,464 ratings, average degree of 45.36, maximal degree of 5,546.

Performance indicators. In order to understand how the overall objective is impacted by the addition of spread, as well as the impact of the weights associated with the three objective, the first performance indicator we consider is the *social benefit score*, i.e., the overall score

$$\alpha \cdot SW(G, I) + \beta \cdot F(G, I) + (1 - \alpha - \beta) \cdot Spread(G|\overrightarrow{\gamma}_I) \quad (15)$$

We consider *validation precision* as a second indicator, aiming to evaluate how often the recommendation retrieves items that are already rated by the group in the real world. After the items already rated by the group are "muted" (removed from the rating collection), through our algorithms, we recommend a list of selected items to the targeted group and, whenever a "muted" item is in this list, we count this as a hit (a binary value), leading in the end to an overall hit ratio for the recommended top-k list of items.

While the objective of our model is to satisfy the group users interests, which we evaluate with the social benefit score and validation precision, *spread beyond the group* is another indicator for performance that we must take into account. By evaluating *spread efficiency*, we denote the later propagation of the recommendation items within the given social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, beyond the given group $G \subset \mathcal{V}$.

Groups and items. Over the three datasets, we selected the targeted recommendation groups G and lists of items I

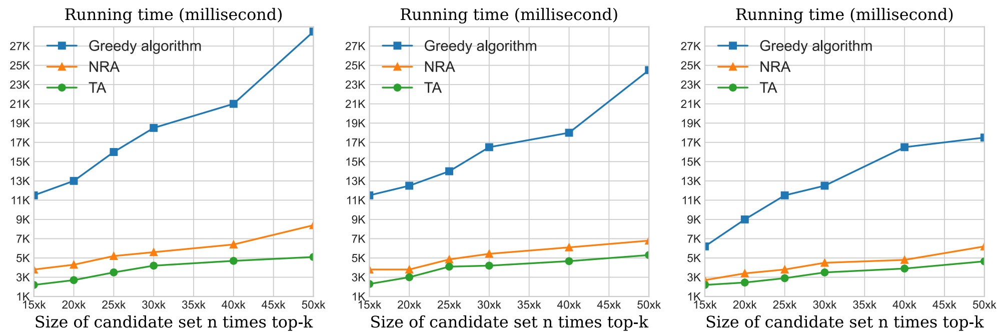

Figure 4: Execution time for greedy and top-k algorithms, with size of candidate set between $20 \times k$ and $50 \times k$, $k = 3$.

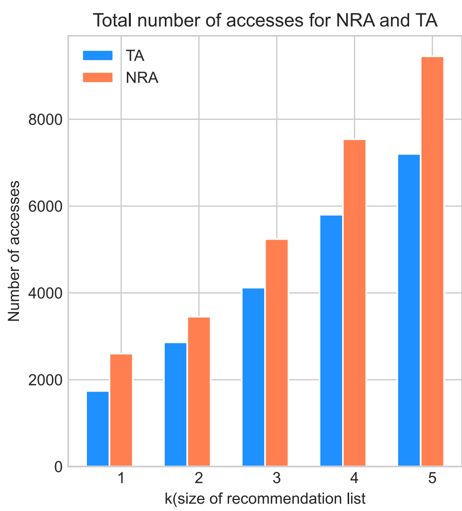

Figure 5: TA / NRA comparison by number of accesses.

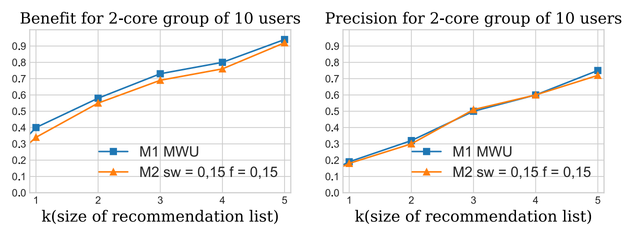

Figure 6: Comparison with MWU on benefit, precision (Flixster).

from which we recommend in our experiments. We did this selection by two methods, plus a random baseline.

In a first method, *groups-to-items*, we firstly select an x-core group G of users (x-cores are tightly connected communities, where each member has at least x links to other members of that community (Ding et al. 2017)). Then, we initiate the list of items I up for recommendation by selecting k items (for several values of k), among the most rated by the group. For this selection method, we considered 2-cores with 10 or 20 users, and 3-cores with 10 or 15 users.

In a second method, *items-to-groups*, we randomly select k items, based on rating distribution, from the top-100 most rated ones. We then find the users who rated these items, in order to construct a group we recommend to (we stop when a first usable group is found, depending on edge density within). For this selection method, we consider groups of 10, 15, and 20 users, with a density of at least 50 edges.

Finally, we also consider as a baseline selection method one for 20 randomly chosen users and k items among the most rated common items from them.

The final list of items I to be recommended consists then of the selected k relevant items plus $9 \times k$ items randomly selected from the overall item set. k, the recommendation result size, varies from 1 to 5. For all items up for recommendation, we remove the ratings relations between group members before running our recommendation algorithms. For each setting, the methods are executed 50 times (for different groups/items), with results averaged.

Greedy evaluation results. We benchmark first the performance of the greedy algorithm, in the following scenarios: (a) social welfare only, (b) fairness only, (c) welfare and fairness equally, without spread, (d) influence spread only, and (e) all three objectives jointly. Specifically, the parameter values are (i) $\alpha = 1, \beta = 0$, i.e., $\max SW(G, I)$, (ii) $\alpha = 0, \beta = 1$, i.e., $\max F(G, I)$, (iii) $\alpha = 0.5, \beta = 0.5$, i.e., $\max 0.5 \cdot SW(G, I) + 0.5 \cdot F(G, I)$, (iv) $\alpha = 0, \beta = 0$, i.e., $\max Spread(G|\overrightarrow{\gamma}_I)$, (v) $\alpha = 0.15, \beta = 0.15$, i.e., $\max 0.15 \cdot SW(G, I) + 0.15 \cdot F(G, I) + 0.7 \cdot Spread(G|\overrightarrow{\gamma}_I)$. The linear combination by $\alpha = 0.15, \beta = 0.15$ did slightly better than other combinations of the three objectives, such as $\alpha = \beta = 0.45$, or $\alpha = \beta = 0.3$. We also include for comparison three baseline methods: two based on the collaborative filtering approach of (Koren and Bell 2015), described below and denoted CF GP and CF Item, along with one based on Integer Programming (IP) (Figures 1, 2, and 3). These results show consistently that by the addition of spread as an objective for group recommendation (along social welfare and fairness), across different datasets and settings, we can indeed obtain higher spread, while maintaining similar benefit levels and only slightly losing in terms of precision. The results over the 50 runs are generally stable, as can be seen from the confidence intervals in the plots.

Collaborative Filtering (CF) baseline. We consider two baseline methods that directly apply collaborative filtering (Koren and Bell 2015).The first one is on an aggregated user profile. For user-user collaborative filtering, we aggregate the relevance vectors of the group's members into a single, group profile. With respect to this aggregated vector, we then take similar, individual user profiles (10 nearest neighbors, by cosine similarity, in our experiments) and their adopted items are used for the recommendation. The second one is item-item based. We rank items by the relevance score $rel(u, i)$ and choose each k recommended items per user u. For a give a group G, we sum the top-k items per every user u, and pick the final top-k items as the recommendation result for the group.

Integer Programming (IP). Since the function that we adopt is not an overly complex one, using an integer programming solver could be a reasonable baseline for the combination of the SW and F functions. This integer program can be formulated as in (Xiao et al. 2017), with the main flow in 6:

$$\max \alpha \cdot SW(G, I) + \beta \cdot \frac{1}{k}\text{Utility}_{min}$$
$$\text{s.t.} \sum_i X_i = k,\ \text{Utility}(u, I) \geq \text{Utility}_{min}, \forall u \in G$$
$$X_i \in \{0, 1\} \quad (16)$$

Due to space reasons, we leave out of this paper some of our experimental results, for (i) 2-cores groups with 20 users and 3-cores groups with 10 or 15 users, (ii) groups of 10 and 20 users, with a density of at least 50 edges. These experimental results show similar trends of gains, in terms of spread, with good performance for benefit and precision.

Comparison with top-k algorithms. We next investigate the performance of our adaptations of the top-k algorithms NRA / TA, comparing them with the greedy algorithm on precision and spread efficiency (we did not revisit benefit as we were mainly focused on model validation).

The initial objective for these methods was to improve upon the execution time of the greedy approach. Therefore, in a first experiment, we compared execution time, for a candidate set size that has a wider rage of values (instead of $10 \times k$), in Figure 4. This was done for the setting of the 2-core group of 10 users and for the group of 15 users having at least 50 intra-group edges, for the three datasets. As shown in Fig. 4, the execution times of the top-k approaches are three to four times less than the ones of the greedy method, and these values grow at a slower pace as the candidate set size grows. We ignore MWU in Fig. 4, as it is much too expensive compared to the greedy method.

We then looked into the number of accesses in the precomputed lists, to grasp when each top-k algorithm may perform better. Since the NRA adaptation requires only sequential accesses, while the TA-one relies on random accesses by item lookup, we compare the two by setting a 37-to-1 cost ratio for random vs. sequential accesses (Bonér 2012) (cost by reading 1MB from SSD) in Fig. 5, where we can see that overall NRA tends to outperform TA, even if its stop-early condition may be met later. We also observed in our experiments that the two top-k algorithms remain close to the greedy one for precision, and perform comparably well for spread.

MWU comparison for precision and benefit. We present in Fig. 6 the comparison with the MWU meta-parameter approach. We compare MWU (denoted *M1 MWU*) with the top-k algorithm approach having $\alpha = 0.15$, $\beta = 0.15$. Recall that MWU has a high computational cost – it is in effect running the optimization algorithm multiple times – so the comparison is made only on the Flixster dataset, using the same group settings as the previous experiments.

We can observe that MWU does indeed help with finding higher quality recommendation lists, yet our method remains quite close. Overall, MWU's computation cost makes the marginal gains in effectiveness hard to justify in practice, but we leave this direction, of further optimizing an MWU-like meta-parameter approach, for future work.

Algorithm 6 Integer Programming Based Algorithm

Require: Given user $u \in G$; set of targeted items I; recommendation list :k ; $\alpha = \beta = 0.15$
1: Solve the convex programming with an IP solver, denote the solution as $X_i, \forall i \in I$ based on equation.16;
2: Select top items with best value;
3: Swap items to the final objective function with spread, and output the recommendation list and find the k item with greatest value $L = X_i \in I^k$

Conclusions and Future Work

Influence-aware group recommendation is a challenging extension of the classic group recommendation problem, where the balance between in-group coherence and outside spread efficiency needs to be taken into account. We presented a model and problem formulation for it, and we analyzed the scalarization approach and the associated greedy algorithm. We then designed optimizations based on top-k methods, in order to improve the running time of the recommendations. In terms of future work, we aim to investigate further the link between multi-objective optimization and our group recommendation problem, starting with the multiplicative weight update (MWU) method. We also look forward to investigate scenarios in which users try to "game the system", and also provide solutions to re-use results from other users of a system.

Ethics and Competing Interests

The positive outcome of our research is more effective recommendations, leading to more awareness and adoption of items. In our view, there are no negative outcomes and no ethics implications pertaining to the data collection process.

Acknowledgements

This research is part of the programme DesCartes and is supported by the National Research Foundation, Prime Ministers Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

Aggarwal, C. C. 2016. *Recommender Systems - The Textbook*. Springer.

Amer-Yahia, S.; Roy, S. B.; Chawlat, A.; Das, G.; and Yu, C. 2009. Group recommendation: Semantics and efficiency. *VLDB*.

Aranganayagi, S.; and Thangavel, K. 2007. Clustering categorical data using silhouette coefficient as a relocating measure. In *International conference on computational intelligence and multimedia applications (ICCIMA 2007)*, volume 2, 13–17. IEEE.

Arora, A.; Galhotra, S.; and Ranu, S. 2017. Debunking the Myths of Influence Maximization: An In-Depth Benchmarking Study. In *SIGMOD*.

Baltrunas, L.; Makcinskas, T.; and Ricci, F. 2010. Group recommendations with rank aggregation and collaborative filtering. In *ACM ResSys*.

Barbieri, N.; Bonchi, F.; and Manco, G. 2013. Topic-aware social influence propagation models. *Knowledge & Inf. Systems*.

Bell, R. M.; and Koren, Y. 2007. Lessons from the Netflix prize challenge. *SIGKDD Explor*.

Bonér, J. 2012. Latency numbers every programmer should know. *May. Accessed March*.

Boratto, L.; and Felfernig, A. 2018. Group Recommendations. In *Collaborative Recommendations - Algorithms, Practical Challenges and Applications*.

Christensen, I. A.; and Schiaffino, S. N. 2014. Social influence in group recommender systems. *Online Inf. Rev*.

Ding, D.; Li, H.; Huang, Z.; and Mamoulis, N. 2017. Efficient fault-tolerant group recommendation using alpha-beta-core. In *CIKM*.

Eichfelder, G. 2009. An adaptive scalarization method in multiobjective optimization. *SIAM Journal on Optimization*.

Elliot, A.; Timothy, D.; and Robin, M. 2017. *Social Psychology*.

Fagin, R.; Lotem, A.; and Naor, M. 2003. Optimal aggregation algorithms for middleware. *J. of computer and system sciences*.

Goldenberg, J.; Libai, B.; and Muller, E. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*.

Gomez-Rodriguez, M.; Leskovec, J.; and Krause, A. 2012. Inferring Networks of Diffusion and Influence. *ACM TKDD*.

Hu, L.; Cao, J.; Xu, G.; Cao, L.; Gu, Z.; and Cao, W. 2014. Deep Modeling of Group Preferences for Group-Based Recommendation. In *AAAI*.

Huang, H.; Yan, Q.; Chen, L.; Gao, Y.; and Jensen, C. S. 2019. Statistical Inference of Diffusion Networks. *IEEE TKDE*.

Jamali, M.; and Ester, M. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *ACM RecSys*.

Jameson, A.; and Smyth, B. 2007. Recommendation to groups. In *The Adaptive Web*.

Jurvetson, S. 2000. What exactly is viral marketing. *Red Herring*.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *ACM SIGKDD*.

Kempe, D.; Kleinberg, J. M.; and Tardos, É. 2015. Maximizing the Spread of Influence through a Social Network. *Theory Comput.*

Koren, Y.; and Bell, R. 2015. Advances in collaborative filtering. *Recommender systems handbook*, 77–118.

Leskovec, J.; Singh, A.; and Kleinberg, J. 2006. Patterns of influence in a recommendation network. In *PAKDD*.

Li, Y.; Fan, J.; Wang, Y.; and Tan, K. 2018. Influence Maximization on Social Graphs: A Survey. *IEEE TKDE*.

Matsubara, Y.; Sakurai, Y.; Prakash, B. A.; Li, L.; and Faloutsos, C. 2012. Rise and fall patterns of information diffusion: model and implications. In *ACM SIGKDD*.

McCarthy, J. F.; and Anagnost, T. D. 1998. MusicFX: an arbiter of group preferences for computer supported collaborative workouts. In *ACM CSCW*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*.

Ntoutsi, E.; Stefanidis, K.; Nørvåg, K.; and Kriegel, H.-P. 2012. Fast group recommendations by applying user clustering. In *International conference on conceptual modeling*.

O'connor, M.; Cosley, D.; Konstan, J. A.; and Riedl, J. 2001. PolyLens: a recommender system for groups of users. In *ECSCW*.

Qi, S.; Mamoulis, N.; Pitoura, E.; and Tsaparas, P. 2016. Recommending packages to groups. In *IEEE ICDM*.

Quintarelli, E.; Rabosio, E.; and Tanca, L. 2016. Recommending new items to ephemeral groups using contextual user influence. In *ACM RecSys*.

Ricci, F. 2018. Recommender Systems: Models and Techniques. In Alhajj, R.; and Rokne, J. G., eds., *Encyclopedia of Social Network Analysis and Mining, 2nd Edition*. Springer.

Ricci, F.; Rokach, L.; and Shapira, B. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*.

Sahebi, S.; and Cohen, W. W. 2011. Community-based recommendations: a solution to the cold start problem. In *RSWeb*.

Said, A.; Berkovsky, S.; and De Luca, E. W. 2011. Group recommendation in context. In *Proceedings of the 2nd challenge on context-aware movie recommendation*.

Saito, K.; Nakano, R.; and Kimura, M. 2008. Prediction of information diffusion probabilities for independent cascade model. In *KES*.

Salehi-Abari, A.; and Boutilier, C. 2015. Preference-oriented social networks: Group recommendation and inference. In *ACM RecSys*.

Shin, D.; Cetintas, S.; Lee, K.; and Dhillon, I. S. 2015. Tumblr Blog Recommendation with Boosted Inductive Matrix Completion. In *CIKM*.

Udwani, R. 2018. Multi-objective maximization of monotone submodular functions with cardinality constraint. In *NeurIPS*.

Xiao, L.; Min, Z.; Yongfeng, Z.; Zhaoquan, G.; Yiqun, L.; and Shaoping, M. 2017. Fairness-aware group recommendation with pareto-efficiency. In *ACM RecSys*.

Yin, H.; Wang, Q.; Zheng, K.; Li, Z.; Yang, J.; and Zhou, X. 2019. Social Influence-Based Group Representation Learning for Group Recommendation. In *ICDE*.

Yu, Z.; Zhou, X.; Hao, Y.; and Gu, J. 2006. TV program recommendation for multiple viewers based on user profile merging. *User modeling and user-adapted interaction*.

Yuan, Q.; Cong, G.; and Lin, C.-Y. 2014. COM: a generative model for group recommendation. In *ACM SIGKDD*.

Zhao, J.; and Ji, Z. 2018. LSICC: A Large Scale Informal Chinese Corpus. *arXiv preprint arXiv:1811.10167*.

Zheng, N.; Li, Q.; Liao, S.; and Zhang, L. 2010. Flickr group recommendation based on tensor decomposition. In *ACM SIGIR*.

Zhou, C.; Zhang, P.; Zang, W.; and Guo, L. 2015. On the upper bounds of spread for greedy algorithms in social network influence maximization. *IEEE TKDE*.