Item-driven Group Formation

Daniel Valcarce[a], Igo Brilhante[b], Jose Antonio Macedo[b], Franco Maria Nardini[c,*], Raffaele Perego[c], Chiara Renso[c]

[a] *University of A Coruña, A Coruña, Spain*
[b] *Federal University of Ceará, Fortaleza, Brazil*
[c] *ISTI-CNR, Pisa, Italy*

Abstract

Several daily activities, such as traveling to a tourist attraction or watching a movie in the cinema, are better enjoyed with a group of friends. However, choosing the best companions may be difficult: we need to consider either the relations among the chosen friends and their interest in the proposed destination/item. In this paper, we address this problem from the perspective of recommender systems: given a user, her social network, and a (recommended) item that is relevant to the user, our User-Item Group Formation (UI-GF) problem aims to find the best group of friends with whom to enjoy such item. This problem differs from traditional group recommendation and group formation tasks since it maximizes two orthogonal aspects: i) the relevance of the recommended item for every member of the group, and ii) the intra-group social relationships. We formalize the UI-GF problem and we propose two different approaches to address it. In the first approach, the problem is modeled as the densest k-subgraph problem over a specific instance of the social network of the user, while the second approach is based on a probabilistic collaborative filtering method that exploit relevance-based language models. We perform an extensive assessment of several algorithms solving the two approaches in the domain of location recommendations by exploiting five publicly available Location-Based Social Network (LBSN) datasets. The experimental results achieved confirm the effectiveness and the feasibility of the proposed solutions that outperform strong baselines. Indeed, results reveal interesting and orthogonal properties of the two formulations. The probabilistic collaborative filtering approach is more effective than the graph-based one on datasets with sparse social networks but with more dense check-in data. On the contrary, the graph-based model performs very well on datasets which present high sparsity on the ratings and check-ins but a higher number of links among users.

Research Highlights:

- The definition of the User-Item Group Formation (UI-GF) Problem.
- Two formalizations of the UI-GF problem with different properties.
- Experiments on five public Location-Based Social Network (LBSN) datasets.
- Comprehensive comparisons of several algorithms for solving UI-GF.
- Experiments showing the effectiveness and efficiency of the proposed methods.

Keywords: Group Formation, Group Recommendation, Recommender Systems, Location-Based Social Networks

*Corresponding author.

Email addresses: daniel.valcarce@udc.es (Daniel Valcarce), igobrilhante@lia.ufc.br (Igo Brilhante), jose.macedo@lia.ufc.br (Jose Antonio Macedo), francomaria.nardini@isti.cnr.it (Franco Maria Nardini), raffaele.perego@isti.cnr.it (Raffaele Perego), chiara.renso@isti.cnr.it (Chiara Renso)

1. Introduction

Nowadays, recommender systems are a pervasive tool supporting several daily activities. Examples range from recommendations for books and music provided by popular services (such as Amazon[1] or Netflix[2]), to recommendations for attractions to visit and tour itineraries to follow (as the ones provided by TripAdvisor[3] and Skyscanner[4]). In many cases, recommended activities are better enjoyed with travel companions, thus shifting the recommendation paradigm. Instead of recommending items to each user independently, we deal with items and groups of users with social relationships. Traditional recommender systems primarily focus on identifying relevant items to single individuals using well-known techniques such as collaborative filtering [1] or content-based recommenders [2]. When the recommendation targets groups of users, it is referred to as "group recommendation", whose goal consists in identifying items that a given group of users may like [3]. The group recommendation problem is hard to solve as users have diverse preferences and finding a trade-off among these preferences may bring to unsatisfactory or even unsettling recommendations for some of the users involved.

In this paper, we address a complementary and even more challenging problem: given a user and a recommended item, we want to suggest the "best" group of friends with whom to enjoy the item. Consider, for example, a user who has been recommended to visit Paris and we want to be able to suggest travel companions who can join her. Ideally, the members of the group should be willing to visit Paris and be friends with each other to enjoy the staying together. Therefore, we need to carefully balance intragroup friendships and interests. We investigated this scenario and designed recommendation techniques able to suggest the "best" group of k friends for a pair $\langle user, item \rangle$ taking into account both the social relationships and the preferences of the user and the group. Since this approach focuses on the formation of a group given an item and a user, we refer to it as the *User-Item Group Formation* problem. In the remaining of the paper, we often refer to it as UI-GF or simply *group formation* for the sake of readability.

Let us consider the example with 7 users and 3 items depicted in Figure 1. Suppose that we are interested in finding the best group of 3 users who can enjoy item i_2 together with user u_0. Figure 1a reports the relevance score s (ranging from 1 to 5, the higher the value, the greater the relevance) of the items for each user, while Figure 1b shows the social network of user u_0 (i.e., her ego network), where links represent friendship relationships. A trivial solution to our recommendation problem would be to choose those users with the highest relevance scores for the item i_2: users u_3, u_4, and u_2. However, when we look at the social relationships, the perspective changes, since Figure 1b shows that u_0's friend u_2 is not friend of u_3 and u_4. Indeed, a better group of u_0's friends to enjoy item i_2 should include u_3, u_4 and u_5, since these three users are all friends of each other and they still have a good relevance score for item i_2.

This example motivates and stresses the importance of considering both the user-item relevance and the strength of interpersonal relationships when addressing the UI-GF problem. To the best of our knowledge, the first proposal considering both social relations and user-item relevance in the group formation problem was our previous conference paper [4]. In that paper, we formalized the UI-GF problem and we modeled it as a graph problem. Specifically, we reduced the UI-GF problem to the problem of finding the densest k-subgraph in a graph obtained by enriching the user social network with item relevance information. The evaluation was conducted on five publicly available LBSN datasets and we found that the proposed solutions outperformed strong baselines. In this extended work, we aim to deal with the following research questions:

- Is it possible to model the UI-GF problem by means of probabilistic collaborative filtering?
- How can we solve the new probabilistic formalization of the UI-GF problem?
- What is the effectiveness/efficiency of the new proposals? How it behaves w.r.t. the graph-based approach previously introduced in [4]?

---

[1] https://www.amazon.com
[2] https://www.netflix.com
[3] https://www.tripadvisor.com
[4] https://www.skyscanner.com

s	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_1	2	3	1	2	2	1	3
i_2	2	1	4	5	5	2	2
i_3	2	4	3	1	1	3	1

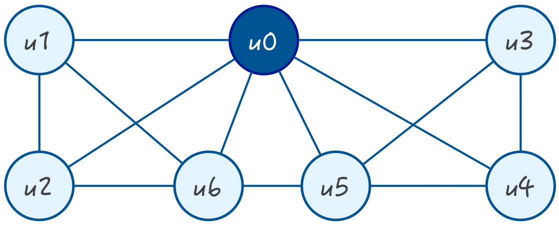

Figure 1: Toy instance of our group formation problem. Table (a) reports the relevance scores of three items for seven users, while the graph in (b) shows the ego network of user u_0 having the same set of users.

To answer the questions above, we present and discuss novel contributions that include:

- an alternative approach to the UI-GF problem. The new modeling employs a probabilistic collaborative filtering method. Collaborative filtering algorithms exploit the interactions between users and items to compute personalized recommendations [1]. In particular, we adapt the IRM2 model presented in [5] and propose different probability estimators to introduce the constraints of the UI-GF problem. This alternative formulation is more computationally expensive than the previous one, but it can yield significantly better results when large amounts of data are available;
- a new comprehensive experimental evaluation of both approaches that employs now five LBSN datasets and use new performance metrics. Experiments demonstrate the validity of the two approaches proposed for solving the UI-GF problem. In addition, results show that the behavior of the two approaches is complementary depending on the sparsity characteristics of the dataset employed.

The rest of the paper is organized as follows. In Section 2, we discuss some related works. Section 3 formalizes the UI-GF problem and we discuss the proposed solutions in Sections 3.1 and 3.2. The experiments for assessing our proposals are reported in Section 4. Finally, Section 5 discusses the implications of this work, draws conclusions and outlines future work.

2. Related Work

One work close to our proposal is the one by Basu Roi *et al.* that discusses the problem of group formation from a group recommendation perspective [6]. The authors indeed consider a problem that is complementary to UI-GF: *how to build groups such that their members are mostly satisfied with the top-k provided recommendations*. The problem consists in building at most l non-overlapping groups of users by considering the similarity between their top-k recommended items. Different methods are proposed to measure the group satisfaction. Although groups are built by considering items recommendations, this proposal ignores the social relationships between the users, which are one of the main focuses of our work. Moreover, in contrast to us, they do not restrict the size of the group which might lead to very large groups.

Some other important research topics are related to this work. In particular: i) *Group Recommendation*, ii) *Team Formation*, iii) *Community Discovery* and iv) *Spatial Social Networks*. In the following, we summarize some results in these fields.

Group Recommendation. This task consists in recommending a tailored list of items to a group of users considering the interests of each member of the group [7]. Ortega *et al.* present a classification of group recommendation techniques in collaborative filtering-based recommender systems [8]. Four different levels at which information about single users can be merged to obtain group-level information are surveyed: similarity metric, neighborhood analysis, prediction phase, determination of recommended items.

Hu *et al.* propose a group recommender system that accommodates both individual choices and group decisions in a joint model through a model built with collective deep belief networks and dual-wing restricted Boltzmann machines [9]. The authors claim that traditional methods aggregating users' preferences or predictions are very sensitive to noise in the data and they may fail to learn group preferences when the data are slightly inconsistent due to strict aggregation assumptions.

Garcia *et al.* introduce a recommender system for tourism able to provide suggestions to groups [10]. Authors design a recommender system taking into account the tastes of the users, their demographic classification and the places they have visited on former trips. The group recommendation is built from individual recommendations through the application of aggregation and intersection mechanisms. While intersection considers the user preferences that are shared by all the members in the group, aggregation takes into account the union of preferences of users in the group, weighted by average user-interest.

Gartrell *et al.* propose a group recommendation technique that integrates social, expertise, and interest dissimilarity of group members [11]. Amer-Yahia *et al.* propose a group recommendation model that takes into consideration the affinity between group members and its evolution over time [12]. They extend existing group recommendation semantics to include temporal affinity in recommendations and design an algorithm that produces temporal affinity-aware recommendations for ad-hoc groups. Kaššák *et al.* present a hybrid recommendation technique that combines both collaborative filtering and content-based approaches [13]. This technique can provide recommendations to either individual user or groups and focuses on the top-N recommendation task. Pera and Ng propose another hybrid approach which combines user tags, item content and item popularity to deliver group recommendations [14].

Recently, Anagnostopoulos *et al.* study the algorithmic implications of suggesting the best set of places that a group of people could perform together in the city [15]. Authors address the problem by providing several formulations that take into account the overall group preferences as well as the individual satisfaction and the length of the tour recommended. Authors provide a study of the computational complexity of these formulations, they provide effective and efficient algorithms, and, finally, they evaluate them on datasets constructed from real city data.

In group recommendation, the group of users is assumed to be known in advance. The task, thus, deals with recommending a list of items to that group. In contrast, we address a different scenario where a recommended item and a user are given and the group that maximizes the relevance of the recommended item for every member and the intra-group social relationships has to be computed.

Team Formation.

The team formation problem asks to build a group offering an optimal match between its members and a set of functional requirements [16, 17, 18]. Lappas *et al.* formulate the team formation problem as: given a social graph where nodes are labeled with a set of skills that each node possesses and given a task that requires a certain set of skills to be satisfied, the objective is to find a subgraph in which all skills are present and the communication cost is small [19]. Although both problems exploit a weighted social graph and the selection process requires group members to be socially close, the team formation problem deals with the coverage of a set of expertizes that make it very different from our UI-GF problem.

Community Discovery. The community discovery problem aims at finding, at the global level, groups (communities) of users with greater ties internally than to the rest of the network. In contrast, our approach focuses on finding the group that maximizes i) the relevance of the recommended item for every member of the group and ii) the intra-group social relationships, based on social network.

An interesting approach is the one proposed by Sozio *et al.* [20]. Here, authors study a query-dependent variant of the community discovery problem, which they call the *community search* problem: given a graph G, and a set of query nodes in the graph, authors propose to find a subgraph of G that contains the query nodes and it is densely connected. The approach differs from ours as i) the problem does not consider information about items as it only relies on the network structure of the graph, ii) they use an undirected graph to model the community whereas our approach can explicitly model asymmetric relationships in the graph.

A classification of community discovery methods is proposed in [21]. The authors classify the methods based on different definitions of communities in the literature. Communities may involve several features like *overlapping*, *weighted* and/or *directed links*, and *dynamics*.

These communities have been exploited in the recommendation process. Lee and Brusilovsky present a recommendation technique that leverage community membership of the users as a useful information source for dealing with cold-start users [22], i.e., users for whom the system do not have enough personal information to provide effective recommendations. However, the authors only focus on regular user-item recommendations and do not explore group recommendations.

Spatial Social Networks Some approaches from the spatial social networks literature are also related to our proposals. Those approaches try to find groups of users with social relations among them that satisfy a given spatial constraint. In contrast, in this work, we model social networks with relevance information about items. Nevertheless, in some cases, we can argue that we can substitute the spatial distance with a metric based on item relevance to tackle the a similar problem to the UI-GF. For example, Yang *et al.* [23] propose Socio-Spatial Group Query (SSGQ) to select a group of nearby people with tight social relations. They show that the problem is NP-hard and design an efficient algorithm SSGSelect to solve it. Although we can replace the spatial distance with a notion of item relevance, the approach is different from the one proposed here for several reasons. First of all, Yang *et al.* model the SSGQ by introducing a parameter k which specifies the average number of unfamiliar people an invitee may have. In our proposed formulation, the notion of familiarity is implicitly modeled by the weighted links of our graph representation or explicitly enforced by probability distributions that takes into account both the social relationship and item relevance for the group members. In any case, it is not controlled by a fixed parameter k. Moreover, SSGQ aims at minimizing the total spatial distance while we address the problem from a user-item relevance point of view by employing aggregation measures that consider the interest of the users for the recommended item.

Liu *et al.* [24] propose another similar socio-spatial approach. They present a new query called *Circle of Friend Query* (CoFQ) to allow finding a group of k people that are close to the target user in terms of physical distance and in terms of social distance. Authors show that the problem is NP-Hard and propose an ϵ-approximation for that. This method has some important differences w.r.t. our proposed approach because they aim to minimize the diameter, i.e., the maximum distance between every two vertices of the group formed. Moreover, they employ a new distance as a weighted average between the geographical distance and the closeness, in terms of social information while we maximize the density of the formed group. As they try to minimize a different function, this may lead to important differences in the resulting groups formed by the two approaches.

3. User-Item Group Formation

The User-Item Group Formation problem asks for a set of users $\mathcal{U}$ and a set of items $\mathcal{I}$. The social network connecting users is modeled as a graph $\mathcal{S} = \{\mathcal{U}, E\}$ where $\mathcal{U}$ is the set of users and E is the set of undirected edges representing the friendship relationship between pairs of users in $\mathcal{U}$. We assume that each edge $e_{uv} \in E$ has a weight $w(u, v)$ indicating the *strength* of the friendship between u and v. Given the target user u, we call $\mathcal{S}_u = \{F_u, E_u\}$ the subgraph of $\mathcal{S}$ representing the social network of u. The nodes $F_u \subseteq \mathcal{U}$ constitute the set of friends of u and $E_u \subseteq E$ are the edges modeling the friendship relationships between these users.

The User-Item Group Formation is a new recommendation problem that takes a user-item pair $\langle u, i\rangle$ as input and asks to find the best group of friends of u for enjoying i by considering two different dimensions:

- **Friendship**. The best group to enjoy an item together should be preferably formed by people that are all friends of each other. Strong ties among users help to enjoy an item together. Thus, we take into account the strength of the friendship among all the members of the proposed group.
- **Item relevance for the group**. The item should be interesting for all the members of the proposed group individually. The users in the group should have, at least, some affinity with the recommended item.

Given these two orthogonal dimensions, the UI-GF problem can be defined as follows:

Definition 1 (User-Item Group Formation)**.** *Given a user u, her social network $\mathcal{S}_u$ and an item i relevant to u, the UI-GF problem seeks to find the group of k friends of u, $F_u^k \subseteq F_u$, that maximizes their "satisfaction", i.e., a measure that takes into account both the relevance of item i for all the members of the group and the intra-group friendship.*

We propose two formalizations of the UI-GF by instantiating two different versions of the above measure of *satisfaction*. In the first approach we formulate UI-GF as a densest k-subgraph problem over an enriched graph built from $\mathcal{S}_u$ and propose two algorithms to address it. The second approach is instead based on collaborative filtering and exploits a probabilistic technique to model the item relevance by taking into account friendship. The graph-based and the collaborative filtering formulations of UI-GF are discussed in Section 3.1 and Section 3.2, respectively.

3.1. UI-GF as a densest k-subgraph problem

We show how we can formulate UI-GF as a densest k-subgraph problem on an enriched instance of $\mathcal{S}_u$, called user-item ego network. First, we discuss how to estimate the item relevance. Then, we show how the enriched instance of $\mathcal{S}_u$ is built and how UI-GF is modeled as a densest k-subgraph problem. Finally, we propose two algorithms to address the problem stemming from the aforementioned graph.

Our graph-based approach relies on the possibility of estimating the relevance $R(u, i)$ of any item in $\mathcal{I}$ for any user in $\mathcal{U}$. We compute such estimates by means of a content-based technique that considers the similarity between the items with which the target user interacted in the past and item i [2].

Without loss of generality, in this paper we estimate the relevance $R(u, i)$ by exploiting the categories describing the venues since these are available in all the LBSN datasets used for the experiments. Let us denote the set of these categories as $\mathcal{C}$. For each venue $i \in \mathcal{I}$ we can easily build its *relevance vector* $\vec{v}_i \in \{0, 1\}^{|\mathcal{C}|}$ where the j-th element of $\vec{v}_i$ is set to 1 iff venue i belongs to category j. Moreover, for each user $u \in \mathcal{U}$, we compute her *preference vector* $\vec{v}_u \in [0, 1]^{|\mathcal{C}|}$ as the normalized sum of the *relevance vectors* of all the venues that u visited in the past [2, 25]. To estimate $R(u, i)$ we exploit the cosine similarity because this metric has shown good results in previous work in recommender systems [26].

Definition 2 (Item Relevance)**.** *Given a user $u \in \mathcal{U}$ and an item $i \in \mathcal{I}$, the relevance $R(u, i)$ of i for u is computed as the cosine similarity between $\vec{v}_u$ and $\vec{v}_i$:*

$$R(u, i) = \frac{\vec{v}_u \cdot \vec{v}_i}{||\vec{v}_u|| \times ||\vec{v}_i||} \tag{1}$$

We can capture the group relevance for a given item using different aggregate strategies [27, 28, 3, 12, 7]. We derived a pairwise version of Aggregated Voting and Least Misery to weight differently the interest of a given item for a pair of users since they are two popular and effective aggregation strategies.

Definition 3 (Pairwise User-Item Relevance)**.** *Given an item $i \in \mathcal{I}$ and users $u, v \in \mathcal{U}$, we define $R_P(u, v, i)$ to be a generic function measuring the pairwise user-item relevance of i for the two users u, v. We can derive two different pairwise user-item relevance measures $R_P(\cdot, \cdot, \cdot)$ from well-known group recommendation counterparts: Aggregated Voting (the sum of the recommendation score of the item for each member) and Least Misery (the minimum of the recommendation scores of the item for each member).*

- Pairwise Aggregated Voting (PAV)*:*

$$R_{PAV}(u, v, i) = R(u, i) + R(v, i) \tag{2}$$

- Pairwise Least Misery (PLM):

$$R_{PLM}(u, v, i) = \min_{z \in \{u, v\}} R(z, i) \tag{3}$$

Since our satisfaction function mixes two orthogonal dimensions, i.e., user-item relevance and friendship relationships, we are now able to define a *pairwise satisfaction* measure that considers both the "strength" of the relationship between users and the relevance of the given item i for those users.

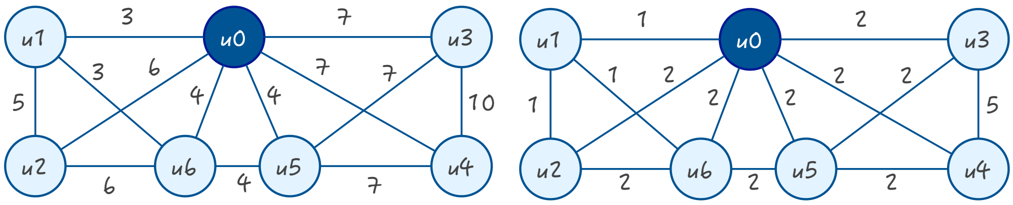

Figure 2: PAV (a) and PLM (b) pairwise satisfaction in the user-item ego network for target user u_0 and item i_2.

Definition 4 (Pairwise Satisfaction)**.** *Given an item $i \in \mathcal{I}$, two users $u, v \in \mathcal{U}$, and the strength $w(u,v)$ of their friendship, the pairwise satisfaction $PS(u,v,i)$ of users u and v w.r.t. the item i is given by:*

$$PS(u,v,i) = w(u,v) \cdot R_P(u,v,i) \tag{4}$$

It is worth noticing that our formalization allows to use any strength measure $w(\cdot,\cdot)$. As an example, we could exploit information about the interactions between pairs of users in the social network, e.g., messages exchanged, common likes, common check-ins, common friends, etc. to measure the strenght of the relation.

We use the pairwise satisfaction measure from Definition 4, using either Aggregated Voting (PAV) or Least Misery (PLM), to build the user-item ego network $\Gamma_{u,i}$ for the target user u and item i:

Definition 5 (User-Item Ego Network)**.** *Given an user u and an item i, the user-item ego network for the pair $\langle u, i \rangle$ is defined as an undirected weighted graph $\Gamma_{u,i} = (F_u, E_{u,i})$ where $F_u \subseteq \mathcal{U}$ is the set of friends of u in the original graph $\mathcal{S}$, and $E_{u,i}$ is the set of edges between nodes in F_u weighted by pairwise satisfaction $PS(\cdot,\cdot,i)$.*

Considering again the example reported in Figure 1. In Figures 2a and 2b, we show the user-item ego network for target user u_0 and item i_2 obtained by weighting edges according to the Pairwise Aggregated Voting and Pairwise Least Misery measures, respectively. The values on the edges represent thus the pairwise satisfaction $PS(\cdot,\cdot,i_2)$.

We model the graph-based UI-GF problem of finding the best group of friends of a user for a recommended item as the problem of finding the densest k-subgraph over the user-item ego network. In this formulation, we aim to find a subgraph of exactly k users that maximizes the following measure of pairwise satisfaction density:

Definition 6 (Pairwise Satisfaction Density)**.** *Given the target user $u \in \mathcal{U}$ and the recommended item $i \in \mathcal{I}$, the pairwise satisfaction density of the subgraph $G_{u,i} = (F_u^G, E_{u,i}^G)$ of $\Gamma_{u,i}$ where $|F_u^G| = k$ is given by:*

$$\rho(G_{u,i}) = \frac{2\sum_{\forall v,w \in F_u^G} PS(v,w,i)}{k(k-1)} \tag{5}$$

This density measure allows us to choose in F_u a group of k users characterized by strong friendship relationships and high interest to the proposed item i. The graph-based UI-GF problem can be thus formulated as the following maximization problem:

Definition 7 (UI-GF as Densest k-Subgraph Problem)**.** *Given the target user $u \in \mathcal{U}$ and the recommended item $i \in \mathcal{I}$, the user-item ego network $\Gamma_{u,i}$ and an integer k, the User-Item Group Formation problem asks to find the subgraph $G_{u,i} = (F_u^G, E_{u,i}^G)$ of $\Gamma_{u,i}$ where $|F_u^G| = k$ that maximizes the pairwise satisfaction density:*

$$\begin{aligned} G_{u,i} = \arg\max_{G^*_{u,i}} \quad & \rho(G^*_{u,i}) \\ \text{s.t.} \quad & G^*_{u,i} \subseteq \Gamma_{u,i}, |F_u^{G^*}| = k \end{aligned} \tag{6}$$

The densest k-subgraph problem is NP-hard since it generalizes the clique problem [29]. Therefore, we address the graph-based UI-GF problem by means of an approximation algorithm (Greedy) and a k-Nearest-Neighbor heuristic ($k-$NN). Both these algorithms exploit a measure of pairwise satisfaction aggregated at the level of each user to maximize the pairwise satisfaction density. We call this measure *aggregated user satisfaction.*

Definition 8 (Aggregated User Satisfaction)**.** *Given the user-item ego network* $\Gamma_{u,i} = (F_u, E_{u,i})$*, the* aggregated user satisfaction*,* $\phi(v, i)$ *for user* $v \in F_u$ *and item* i *is defined as the sum of the pairwise satisfaction computed over all its neighbors:*

$$\phi(v,i) = \sum_{w \in F_u} PS(v, w, i) \tag{7}$$

3.1.1. A greedy approximation algorithm to solve the graph-based UI-GF

Greedy is an approximation algorithm to solve the densest k-subgraph problem. It works by repeatedly removing from $\Gamma_{u,i}$ the node w with the minimum value of $\phi(w, i)$ (line 3), and by updating the values $\phi(v, i)$ of its neighbor nodes v accordingly. This process is repeated until exactly k nodes are left (condition in line 2). The pseudo-code of the algorithm is shown in Algorithm 1. It has been introduced by Asahiro *et al.* [29]. Authors prove that the algorithm has tight bounds on the worst case approximation ratio.

Algorithm 1 Greedy algorithm for UI-GF.

Input: User u, item i, $\Gamma_{u,i}$, integer k
Output: $G_{u,i} = (F_u^G, E_{u,i}^G), |F_u^G| = k$

```
1: G_{u,i} ← Γ_{u,i}
2: while |F_u^G| > k do
3:     w ← node with minimum φ(w, i) in G_{u,i} {use a Fibonacci heap to find the node w}
4:     update φ(v, i) of every neighbor v of w
5:     remove w from G_{u,i}
6: end while
7: return G_{u,i}
```

Complexity Analysis. The complexity of the algorithm depends on the values of aggregated user satisfaction, $\phi(\cdot, \cdot)$. As claimed in [20, 30], Greedy can be implemented in linear time $\mathcal{O}(n + m)$, for m edges and n nodes, when the image of the function $\phi(\cdot, \cdot)$ is a subset of $\mathbb{N}_0$. In many real applications, however, this function is not an integer value. In fact, in our case, the aggregated user satisfaction function provides real values. The algorithm, in this case, needs to use a different strategy to efficiently find the node with minimum aggregated user satisfaction and update the values of its neighbors' $\phi(\cdot, \cdot)$. Charikar *et al.* suggested the use of a Fibonacci heap to hold the nodes indexed by their aggregated user satisfaction. In this way, we obtain a final complexity of $\mathcal{O}(m + n \log n)$ [30]. The Fibonacci heap enables us to extract the node associated with the minimum value in $\mathcal{O}(\log n)$ and update the value of a given node in $\Theta(1)$ [31, Chapter 19]. As the algorithm removes at most n nodes and updates at most m neighbors (edges), Greedy with Fibonacci heap has a complexity of $\mathcal{O}(m + n \log n)$, for m edges and n nodes in the user-item ego network.

3.1.2. A k-NN algorithm to solve the graph-based UI-GF

The k Nearest Neighbor technique is a well-known non-parametric algorithm successfully employed in several domains ranging from recommender systems to clustering. Here, we employ k-NN on the user-item ego network (Algorithm 2) to retrieve the k neighbors of target user u having the highest values of aggregated user satisfaction (lines 1–2) to create the set of nodes F_u^G. Then, the algorithm returns the subgraph $G_{u,i}$ induced by set F_u^G.

Algorithm 2 k-NN algorithm for UI-GF.

Input: User u, item i, $\Gamma_{u,i}$, integer k
Output: $G_{u,i} = (F_u^G, E_{u,i}^G), |F_u^G| = k$

$L \leftarrow$ sort $v \in F$ in descending order of $\phi(v, i)$
$F_u^G \leftarrow$ first k nodes of L
$G_{u,i} \leftarrow$ subgraph of $\Gamma_{u,i}$ induced by F_u^G
return $G_{u,i}$

Complexity Analysis. The algorithm sorts all the nodes in $\Gamma_{u,i}$ in $\mathcal{O}(n \log n)$. At most n nodes are selected to create the set F_u^G in $\mathcal{O}(n)$. Finally, the subgraph induced by F_u^G is created in $\mathcal{O}(m)$. Therefore, the final complexity of k-NN is bounded by $\mathcal{O}(m + n \log n)$.

3.2. UI-GF as an item relevance modeling problem

We propose to address the User-Item Group Formation by formulating it as an item relevance modeling task for probabilistic collaborative filtering. Collaborative filtering algorithms exploit the past interactions between users and items to generate personalized suggestions [1]. In contrast to content-based recommenders, they do not require metadata about the items: items are considered black boxes. Since we assess the proposed solutions in the context of LBSNs, we exploit past interactions between users and items, i.e., venues. Additionally, some of these social networks allow users to emit a rating for the venue. We will consider this as an explicit interaction and $r_{u,i}$ will represent the rating that the user u gave to an item i. When ratings are not available, we will rely on the normalized count of check-ins to estimate $r_{u,i}$. In the following, we show our proposal for the UI-GF problem based on an adaptation of an algorithm based on relevance-based language models [5].

Relevance-based language models are a state-of-the-art technique for performing pseudo-relevance feedback in a text retrieval scenario [32]. Even though these methods have originated in the field of Information Retrieval (IR), an emerging trend of applying techniques from IR to recommendation is gaining attention [33, 34]. Following this trend, Parapar *et al.* adapted the relevance-based language modeling framework to the collaborative filtering scenario obtaining high figures of precision [35].

Recently, an item-based relevance modeling framework for collaborative filtering has been proposed in order to deal with a novel recommendation task: the liquidation of long tail items [5]. This task consists in identifying the most suitable users for offering them a given long tail product. This algorithm, called IRM2, creates a relevance model for every long tail item and estimates the probability of relevance of each user under these models. Each relevance model is built upon the users' feedback which can be either explicit (e.g., ratings) or implicit (e.g., check-ins). This technique, which achieved excellent results in the task of liquidating long tail items, can be used to formalize our UI-GF problem in an alternative way. We propose to employ IRM2 to solve the UI-GF problem because, both in the long tail liquidation and in the group formation tasks, we aim to recommend the most appropriate users for a target item. However, its use is not straightforward because both tasks possess their own peculiarities. In particular, when addressing the UI-GF problem, we have to deal with all types of items, not only with long tail ones (i.e., the least popular venues). This is actually not a difficulty since recommending for long tail items is, in principle, harder than recommending for regular ones. The main difference between the long tail liquidation task and the UI-GF problem is that the latter exploits the friendship relationships among users whereas the former does not deal with this kind of information.

Given the item $i \in \mathcal{I}$, IRM2 estimates the relevance of a user $v \in \mathcal{U}$ under the relevance model R_i as follows [5]:

$$p(v|R_i) \propto p(v) \prod_{w \in \mathcal{U}_i} \sum_{j \in J_i} p(w|j) \frac{p(v|j)\, p(j)}{p(v)} \tag{8}$$

where $U_i \subseteq \mathcal{U}$ refers to the set of users who interacted with item i and $J_i \subseteq \mathcal{I}$ denotes the set of similar items to item i.

The User-Item Group Formation for the target user u and the recommended item i can be addressed by estimating the probability of relevance of each friend $v \in F_u$ under the relevance model of the target item R_i. The recommended group consists of the k users with the highest estimated relevance. Formally, UI-GF can be defined as the following:

Definition 9 (UI-GF as an Item Relevance Modeling problem)**.** *Given the target user $u \in \mathcal{U}$, the recommended item $i \in \mathcal{I}$ and an integer k, the User-Item Group Formation problem asks to find the set $F_u^G \subseteq \mathcal{U}$ where $|F_u^G| = k$ whose users $v \in F_u^G$ maximize the probability of relevance under the model of the recommended item:*

$$\begin{aligned} F_u^G = \arg\max_{F_u^*} & \quad \sum_{v \in F_u^*} p(v|R_i) \\ \text{s.t.} & \quad F_u^* \subseteq \mathcal{U}, |F_u^*| = k \end{aligned} \tag{9}$$

The pseudocode of IRM2 is shown in Algorithm 3. We consider the set F_u as candidate users which consists of only those users who are friends of the target user. Additionally, to fully specify this technique, we need to provide the details of how to compute the set of similar items as well as the estimates of conditional and prior probabilities. It is worth highlighting that the original formulation of IRM2 considers only the probability of relevance under the model of the recommended item [5]. To introduce the social relationships into the IRM2 model, we extend the model by defining novel prior probability estimators that take into account the social information.

Algorithm 3 IRM2 algorithm for UI-GF.

Input: User u, item i, candidate users F_u, integer k
Output: $F_u^G, |F_u^G| = k$

```
1: F_u^G ← {}
2: H ← build max-heap for each w ∈ F_u with values p(w|R_i)
3: while |F_u^G| < k do
4:     w ← retrieve node with maximum p(w|R_i) in H
5:     add user w to F_u^G
6: end while
7: return F_u^G
```

We build the set of most similar items, J_i, by taking the l most similar items to i according to a pairwise similarity metric. Note that l is one of the parameters of this model. As before we employ the cosine similarity metric to compute the similarity between items. The cosine similarity between two items i and j is given by:

$$s(i,j) = \frac{\sum_{u \in \mathcal{U}_i \cap \mathcal{U}_j} r_{u,i}\, r_{u,j}}{\sqrt{\sum_{u \in \mathcal{U}_i} r_{u,i}^2} \sqrt{\sum_{v \in \mathcal{U}_j} r_{v,j}^2}} \tag{10}$$

The cosine similarity measures the similarity between items computing the dot product over the intersection of users that rated both items. Since our aim is to maximize the satisfaction of the members of the group, the calculation of the similarities between items is mainly based on the agreement among users.

The conditional probability of a user given an item is computed using the maximum likelihood estimate of a multinomial distribution over the count of interactions [5]. However, this estimate suffers heavily from data sparsity. To address this problem, the authors of IRM2 employed absolute discounting smoothing [5]. However, a recently published axiomatic analysis of different smoothing methods for relevance-based language models in recommendation has found that additive smoothing is a better option because it does not demote the IDF effect [26]. Additive smoothing (also referred to as Laplace smoothing) increments all the interactions by a parameter $\gamma > 0$. To the best of our knowledge, this is the first time that additive smoothing is applied to item relevance modeling. In previous studies it was applied only to the user counterpart [26].

Finally, the estimate of the conditional probability of the user u given the item j is given by:

$$p(u|j) = \frac{r_{u,j} + \gamma}{\sum_{v \in \mathcal{U}_j} r_{v,j} + \gamma|\mathcal{U}|} \tag{11}$$

We now provide the details of the prior estimates, $p(j)$ and $p(v)$, used in Eq. 8. Both of them have been considered uniform in [5]. In this article, we use an uniform prior estimator for items, i.e., $p(j)$, while we provide different priors for users, i.e., $p(v)$, because we want to consider also the social graph generated by the friendship relationships to maximize the group satisfaction. Despite this information is not modeled in the original formulation of IRM2, one of the advantages of this relevance modeling framework is its sound statistical foundation which enables us to introduce different types of information in the probability estimates. In fact, previous work on relevance-based language models for recommendation has found that priors different from the uniform estimate can lead to significant improvements [36].

Therefore, we use a uniform item prior and we propose four probability estimates for the user prior. With these non-uniform user priors, IRM2 is able to model a satisfaction function that takes into account both the "strength" of the relationship between users and the relevance of the given item i for those users. We describe below our proposals.

- **Uniform (U).** As a baseline, we studied the uniform estimate for the user prior. Since the set of candidate users of the group recommendation task is F_u (the friends of the target user u), the formulation of this prior is the following:

$$p(v) = \frac{1}{|F_u|} \tag{12}$$

- **Common Friends (CF).** This prior promotes those users who share a large number of common friends with the target user. Since the user prior is in the denominator of (8), we formulate a prior which is inversely proportional to the number of common friends.

$$p(v) \propto \frac{1}{|F_u \cap F_v|} \tag{13}$$

- **Common Group Friends (CGF).** This estimate boosts those users who have more common friends with the members of the current group $G_{u,i}$. Initially, this group is constituted by the target user and this prior behaves as the CF prior. This prior should be updated in the group formation procedure. This modifies Algorithm 3 into Algorithm 4.

$$p(v) \propto \frac{1}{\left|\left(\bigcup_{w \in F_u^G} F_w\right) \cap F_v\right|} \tag{14}$$

- **Group Closeness (GC).** This estimate boosts those users who have more friends in the current group $G_{u,i}$. This prior estimate should also be updated incrementally in the group formation procedure using Algorithm 4.

$$p(v) \propto \frac{1}{|F_u^G \cap F_v|} \tag{15}$$

Complexity Analysis. First, we analyze the complexity of (8) being n the number of users ($n = |\mathcal{U}|$), m the number of items ($m = |\mathcal{I}|$), l the number similar items and v the number of candidates users $v = |F_u|$. Uniform priors can be computed in $\Theta(1)$ and the rest of the user priors can be cached in $\mathcal{O}(n)$. Conditional probabilities, according to (11), can be computed in $\Theta(1)$ if the sum in the denominator is precomputed and cached for each item j. Having cached the priors and the sum of interactions, the cost of evaluating (8) is bounded by $\mathcal{O}(nl)$. Additionally, we have to compute the set J_i of similar items which is $\mathcal{O}(m)$.

Algorithm 4 IRM2 algorithm for UI-GF (modified version for prior CGF).

Input: User u, item i, candidate users F_u, integer k
Output: $F_u^G, |F_u^G| = k$
1: $F_u^G \leftarrow \{\}$
2: **while** $|F_u^G| < k$ **do**
3: $\quad w \leftarrow$ retrieve node with maximum $p(w|R_i)$ in F_u {evaluate $p(w|R_i)\ \forall w \in F_u$}
4: $\quad$ remove user w from F_u
5: $\quad$ add user w to F_u^G
6: $\quad$ update prior with the new set F_u^G
7: **end while**
8: **return** F_u^G

Now, we study Algorithm 3. Building a max-heap is a linear operation. Since we have to compute (8) for each candidate user, line 2 has a complexity of $\mathcal{O}(vnl)$. The while loop runs k times performing a $\mathcal{O}(\log n)$ operation (line 4) and a $\Theta(1)$ operation (line 5). Thus, the loop has a complexity of $\mathcal{O}(k \log n)$. If we take into account the computation of similar items, the complexity of Algorithm 3 is bounded by $\mathcal{O}(m + vnl + k \log n)$.

On the other hand, Algorithm 4 is more costly. Line 3 has a complexity of $\mathcal{O}(vnl)$ because it computes IRM2 for each candidate and lines 4 and 5 runs in constant time. Updating the cached priors is in $\mathcal{O}(v)$. Finally, since the while loop runs for k times, we obtain a final complexity of $\mathcal{O}(m + kvnl)$ for Algorithm 4.

4. Experimental Evaluation

We used five publicly available LBSN datasets to conduct a thorough evaluation of our proposals against state-of-the-art baselines. First, we present the datasets. Since we are dealing with a novel problem, we propose a new evaluation methodology based on ground-truth groups. Next, we detail the baseline algorithms and the metrics used for evaluation. Finally, we describe and discuss the results of the experiments.

4.1. Datasets

We employ five publicly available datasets collected from four popular LBSNs: Foursquare, Brightkite, Gowalla and Weeplaces. These datasets record information about the users registered in these social networks and the venues where the users checked-in. All datasets contain entertainment places such as restaurants, cinemas or tourist attractions among other venues. The social links between users are bidirectional friendship relationships.

Foursquare is a popular LBSN where users check-in to inform their friends on the places where they are. Thanks to the authors of [37, 38], we downloaded a dataset containing users check-ins, places, users ratings of the places and the social graph connecting users[5]. Starting from this dataset, which is called hereinafter *Foursquare*, we built a second dataset by selecting only the check-ins falling in the bounding box of New York City[6]. This second dataset is called in the following *Foursquare (New York)*. The rationale of taking a subset of the Foursquare dataset was to evaluate the proposed approaches in a less sparse scenario where all the information is concentrated in a single location. This enables us to test our solutions on a dataset with richer social connections. In addition, we used two other datasets, collected from Brightkite and Gowalla[7] made available by the authors of [39]. These datasets, named in the following *Brightkite* and *Gowalla*, record user check-ins and the social network connecting users but they lack ratings of the visited venues. The appreciation of a user for a venue was thus estimated for the purpose of our work on the basis of the normalized number of check-ins made by the user in that venue: the more the check-ins, the higher the rating. Finally, we used the *Weeplaces* dataset[8] which contains check-ins and friendship relationships of

[5] `https://archive.org/details/201309_foursquare_dataset_umn`
[6] `https://www.flickr.com/places/info/2459115`
[7] Available at `https://snap.stanford.edu/data`
[8] Available at `http://www.yongliu.org/datasets`

Table 1: Statistics regarding the five datasets used in the experiments: Foursquare, Foursquare (New York), Brightkite, Gowalla and Weeplaces

Dataset	**Foursquare**	**Foursquare (NY)**	**Gowalla**	**Brightkite**	**Weeplaces**
# users	2 138 367	103 663	196 591	58 228	15 799
# users w/ check-ins	485 381	82 469	107 092	51 406	15 793
# users w/ friends	1 880 404	55 252	196 591	58 228	15 538
# users w/ all	227 418	34 058	107 092	51 406	15 532
# items	83 999	7813	1 280 969	772 966	971 307
# links	27 098 472	1 890 844	1 900 654	428 156	114 131
# links per user	14.41	34.22	9.67	7.35	7.35
# check-ins	1 021 966	157 064	6 442 892	4 747 281	7 369 712
# check-ins per user	2.10	1.90	60.16	92.35	466.64
# check-ins per item	12.17	20.10	5.03	6.14	7.59
check-ins density (%)	0.003	0.024	0.005	0.012	0.050
# ratings	2 809 580	330 043			
# ratings per user	4.24	3.09			
# ratings per item	33.45	42.24			
ratings density (%)	0.005	0.040			

Foursquare users who used the Weeplaces application.

We used the Foursquare API[9] for all the datasets to obtain the categories for the venues used in the content-based approach described in Section 3.1.

Table 1 shows the main statistics of these datasets. Foursquare is the largest dataset in terms of the number of users, with a very large social network made up of about thirteen million edges. Weeplaces has the largest number of check-ins. The degree distributions of the users in the social networks are shown in Figure 3. As expected, all the datasets present a power-law distribution in the node degrees: the majority of the users have a limited number of friends, while only a few users have thousands or more friends. This is an important consideration as the degree distribution affects the size of the user-item ego network $\Gamma_{u,i}$.

The datasets are extremely sparse in terms of check-ins and ratings. We computed the density as the proportion of ratings/check-ins with respect to the number of users times the number of items. We discarded those users without check-ins to compute the metrics related to check-ins and we did the same for ratings. Even in this way we can observe that rating/check-in density is below 0.01% in all the cases. This poses a challenge for any recommender system and, in particular, for solving the UI-GF problem. In particular, Foursquare collections have an especially low density of ratings and check-ins while the other datasets have very sparse social networks. These particularities affect the performance of our proposals as reported in Section 4.5.

4.2. Evaluation Methodology

To assess the quality of the groups proposed by our solutions to the UI-GF problem, we propose to compare them against ground-truth groups, i.e., groups of friends that actually enjoyed a specific venue together. We extracted these ground-truth groups from the five datasets. In particular, we looked for sets of users who checked-in the same place within a fixed temporal window. We considered a user to be a member of a group only if this person is a friend of at least one of the other group members. In this way, we obtained groups of users who actually enjoyed the place where they checked-in, together with their friends.

[9] `https://developer.foursquare.com`

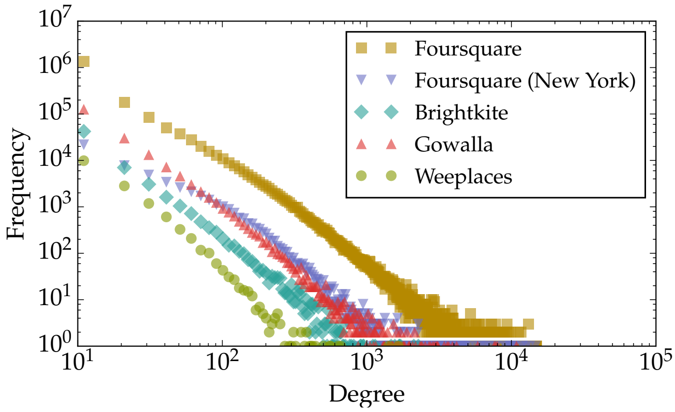

Figure 3: Degree distributions of the social networks of the datasets.

After an empirical analysis of the five datasets, we decided to set a temporal window of 4 hours. Different values of the temporal windows affect the number (and the size) of the ground-truth groups mined. In our experiments we consider only groups with at least 4 members. The 4-hours window allows us to mine $1,495$ ground-truth groups on Foursquare, 258 on Foursquare (New York), $24,996$ on Brightkite, $27,997$ on Gowalla and $39,148$ on Weeplaces. Weeplaces has the largest number of ground-truth groups since it also has the largest number of check-ins (see Table 1).

The evaluation methodology uses these ground-truth groups in the following way: from each of these groups, we select a random member as the target user and the venue where the group registered as an item. Then, we asked our proposals to form a group of k friends for this specific user and venue, with k ranging in $\{4, 6, 8, 10, 12\}$. The members of the ground-truth group are those who we would like to find in the group suggested by the algorithmic solution of the UI-GF problem. In the next section, we present three metrics for assessing the quality of the recommended groups with respect to the ground-truth groups.

4.3. Performance Metrics

We evaluate our proposals by using metrics that exploit the ground-truth groups above discussed. We denote with $\hat{F}_{u,i}$ the ground-truth group for user u and venue i. To evaluate the quality of group F_u^G formed by our techniques and by their competitors, we used set-based information retrieval metrics: *precision*, *recall* and *F-measure* [40]. We averaged these metrics over all the ground-truth groups in each dataset.

4.3.1. Precision

This metric computes the fraction of members in F_u^G that also appear in the ground-truth group $\hat{F}_{u,i}$:

$$precision(F_u^G) = \frac{|\hat{F}_{u,i} \cap F_u^G|}{|F_u^G|} \tag{16}$$

4.3.2. Recall

This metric computes the fraction of actual group members in $\hat{F}_{u,i}$ that are present in the suggested group F_u^G:

$$recall(F_u^G) = \frac{|\hat{F}_{u,i} \cap F_u^G|}{|\hat{F}_{u,i}|} \tag{17}$$

4.3.3. F-measure

The F-measure or F_1 score is the harmonic mean of precision and recall. This metric ranges from 0 to 1 and shows a high value when both precision and recall are high:

$$F_1(F_u^G) = \frac{2 \times precision(F_u^G) \times recall(F_u^G)}{precision(F_u^G) + recall(F_u^G)} \tag{18}$$

4.4. Baselines

We compare the performance of the solutions proposed with two baselines: Top k-Nodes and Densest k-Subgraph.

4.4.1. Top k-Nodes (k-Top)

Top k-Nodes is a heuristic that computes a dense k-subgraph without considering the edges. It forms the group by retrieving the k nodes of the user-item ego network with the highest value of $R(\cdot, i)$. Note that in this approach the relationships among the users are not considered. Consequently, it does not use any pairwise satisfaction measure.

4.4.2. Densest k-Subgraph (DkSP)

Densest k-Subgraph (DkSP) is a well-known heuristic that aims at approximating the densest k-subgraph of a graph G [41]. It works by first identifying three candidate k-subgraphs by applying the following three procedures:

- *Procedure 1.* Select $k/2$ arbitrary edges from the graph, then return the set of nodes incident with these edges, adding arbitrary nodes to this set if its size is lower than k.
- *Procedure 2.* Create two disjoint sets H and C. The set H includes the $k/2$ nodes with the highest aggregated user satisfaction in the input graph G. The set C is created by selecting $k/2$ nodes from $G \setminus H$ with the highest aggregated user satisfaction with respect to the nodes in H. Return the subgraph induced by the set $H \cup C$.
- *Procedure 3.* Let $W_2(u, v)$ be the function that returns the number of paths of length 2 between two nodes u and v and H be the set with $k/2$ nodes with the highest aggregated user satisfaction in the input graph G. For every node v in H, compute $W_2(v, w)$ for all $w \in G$ and create a set H^v with $k/2$ nodes with the highest $W_2(v, w)$. Then, create the set B^v with the $k/2$ neighbors x of v with the highest aggregated user satisfaction with respect to the set H^v. Finally, return the subgraph G'_v induced by the set $H^v \cup B^v$, adding arbitrary nodes to this set if its size is smaller than k.

Each one of the previous procedures generates a candidate k-subgraph. The DkSP algorithm returns the densest k-subgraph among these three candidates.

4.5. Effectiveness

We now evaluate the proposed algorithms against the baselines using the performance metrics defined above. Our goal is to check if the groups formed by our proposed techniques are really relevant with respect to the ground-truth groups mined from the data. Figures 4, 5 and 6 depict the results for precision, recall and F-measure achieved on the five datasets for all the proposed algorithms and the baselines. We varied the parameter k, which controls the group size of the solution, from 4 to 12 people.

Both Greedy and k-NN outperform k-Top and DkSP in terms of precision for both PAV and PLM metrics. We can observe that on average Greedy achieves better results on Foursquare datasets, while k-NN demonstrates a better performance on the Brightkite, Gowalla and Weeplaces datasets. It is worth highlighting that the improvement is higher for smaller values of k, while for larger groups the difference decreases. Moreover, Greedy and k-NN are able to suggest more precise groups when using the PLM user-item relevance. As shown in Table 2, the precision measured for k-NN using PLM results to be up to 14%, 3%, 5%, 6% higher than the one with PAV for Foursquare, Foursquare (New York), Brighkite and Gowalla datasets, respectively. This result can be interpreted by observing that users tend to invite the friends who are expected to like the venue, while they rarely invite a friend when they know she would not like it. This behavior is captured specifically by the pairwise least misery relevance that considers the minimum among the user-item relevance scores for forming the group.

On the other hand, IRM2 outperforms all the algorithms on the Brightkite and Weeplaces datasets using any prior (see Table 3). After an initial exploratory analysis for IRM2, we set the number of similar items

Table 2: Improvements (%) of *Precision* (p) and *Recall* (r) by varying k for Greedy and k-NN when using PLM instead of PAV.

Algorithm	k	FS		FS (NY)		Gowalla		Brighkite		Weeplaces	
		p	r	p	r	p	r	p	r	p	r
Greedy	4	6.2	6.6	7.2	7.1	5.3	4.8	0.9	2.0	1.4	1.4
	6	7.6	7.5	3.9	5.6	5.6	4.8	−1.0	0.4	1.0	−1.6
	8	7.9	7.8	4.3	6.7	6.5	5.5	−1.7	−0.1	1.1	0.9
	10	7.5	7.3	3.7	5.4	5.9	4.4	−1.5	−0.2	1.6	1.6
	12	6.8	6.6	5.9	5.7	6.3	5.1	−1.8	−0.3	1.8	1.9
k-NN	4	5.2	5.5	2.7	2.6	6.2	6.5	14.6	11.0	1.6	1.5
	6	4.9	4.9	2.0	3.8	5.6	4.9	7.5	6.4	0.8	0.4
	8	5.4	5.3	2.1	4.2	4.6	4.0	4.1	3.7	1.2	0.9
	10	5.0	4.9	3.0	2.8	4.3	3.8	4.4	3.7	1.5	1.5
	12	4.4	4.4	3.5	2.9	4.3	3.8	2.8	2.6	1.7	1.7

Table 3: Improvements (%) of *Precision* (p) and *Recall* (r) of IRM2 (using different prior estimates) over GREEDY and k-NN using PAV when $k = 4$.

Baseline	Prior	Foursquare		Foursquare (NY)		Gowalla		Brighkite		Weeplaces	
		p	r	p	r	p	r	p	r	p	r
GREEDY	U	−60.2	−60.9	−4.4	4.8	−9.5	1.2	74.9	68.3	47.6	52.0
	CF	−58.5	−59.1	2.3	13.6	−8.8	1.9	77.2	69.2	48.5	52.5
	CGF	−53.5	−54.1	−2.5	8.1	8.3	20.3	93.4	85.6	48.0	52.1
	GC	−30.1	−30.4	0.5	10.6	59.4	76.5	90.8	84.4	48.9	53.3
k-NN	U	−61.0	−61.8	0.2	7.4	−16.2	−6.0	54.9	48.2	38.5	42.8
	CF	−59.3	−60.0	7.3	16.4	−15.6	−5.4	57.0	49.0	39.3	43.3
	CGF	−54.5	−55.1	2.2	10.8	0.3	11.7	71.4	63.4	38.9	42.9
	GC	−31.5	−31.9	5.3	13.3	47.5	63.9	69.1	62.3	39.7	44.1

l to 400. In the same way, we set the smoothing parameter γ to 0.001. The proposed priors (CF, CGF and GC) demonstrates a better performance than the original uniform prior (U). In particular, GC constitutes the best estimate and outperforms also all the algorithms on the Gowalla dataset. Also, it provides a significant improvement in performance on the Foursquare datasets. Nevertheless, on the Foursquare datasets, $GREEDY$ is a much better option.

A similar behavior is confirmed when we take into account the recall metric. The plot shows that Greedy and k-NN when using PLM achieve higher recall figures on Foursquare datasets. Interestingly, when PAV is used, the k-Top baseline exhibits better recall than Greedy and k-NN on the Foursquare datasets. The advantage of Greedy using PLM instead of PAV is up to 7% on both Foursquare datasets. Moreover, it is up to 5% and 4% for k-NN on Foursquare and Foursquare (New York), respectively (see Table 2). These results confirm the previous findings from the analysis employing the precision metric. For the Brightkite and Weeplaces datasets, IRM2 provides the best results independently of the prior. Again, the GC prior is the best estimate outperforming also all the algorithms on Gowalla dataset. The relatively high values of precision and recall achieved by our solutions demonstrate that they are indeed able to suggest meaningful and relevant groups of friends with whom to enjoy a given venue.

Finally, in terms of F-measure we can see that the trends are very similar to those from the previous plots. IRM2 with the Group Closeness prior is the best option on Gowalla, Brightkite, and Weeplaces dataset. Meanwhile, on both Foursquare datasets, GREEDY using PLM is the most effective method.

By relating these results to the properties of the datasets (see Table 1), we can argue that, IRM2 works better on datasets with sparse social networks but with more dense check-in data, while graph-based approaches as GREEDY and k-NN perform very well on Foursquare datasets which present high sparsity on the ratings and check-ins but a higher number of links among users. A possible explanation of this

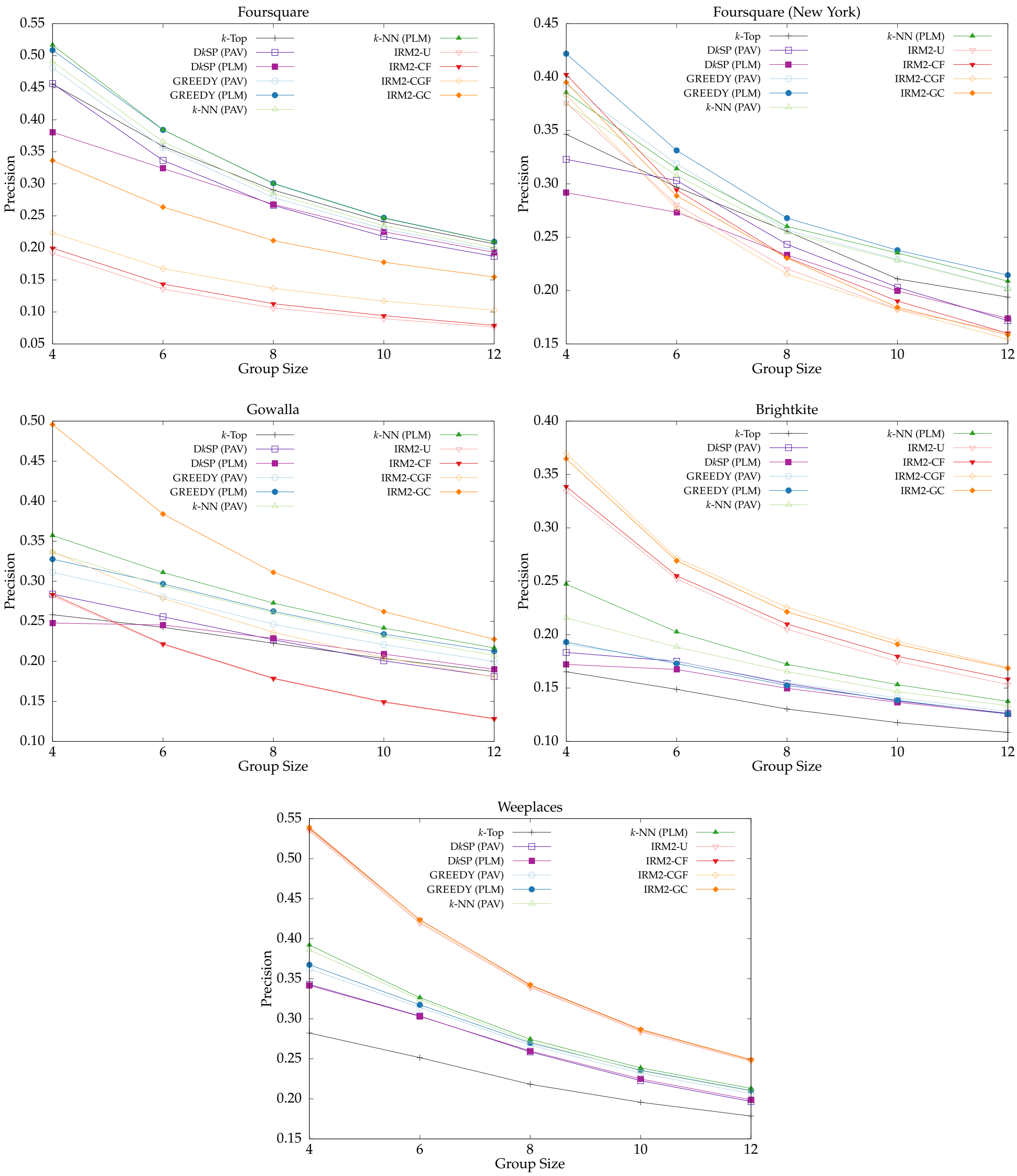

Figure 4: Values of precision of the different algorithms w.r.t. the ground-truth groups on the five datasets.

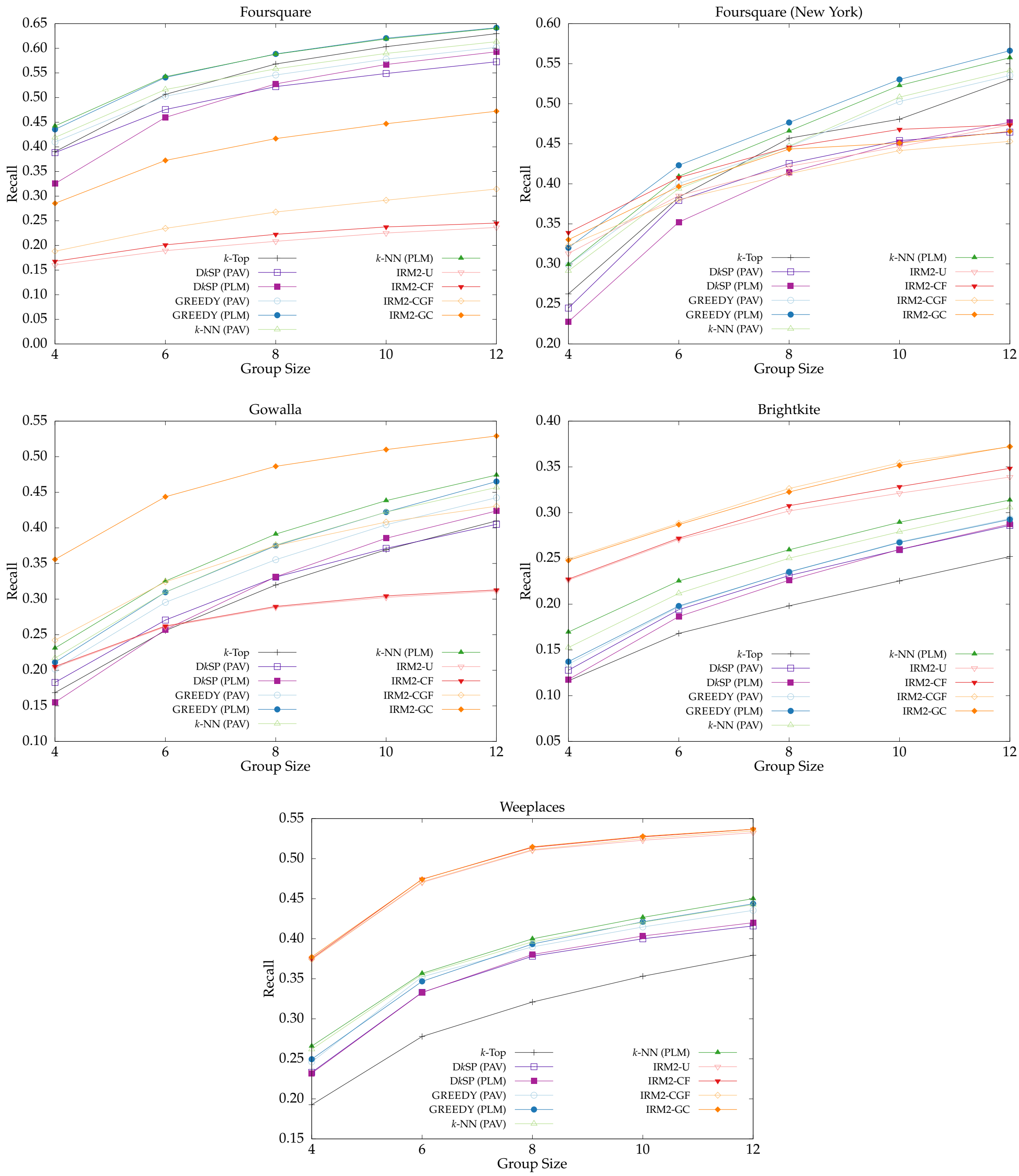

Figure 5: Values of recall of the different algorithms w.r.t. the ground-truth groups on the five datasets.

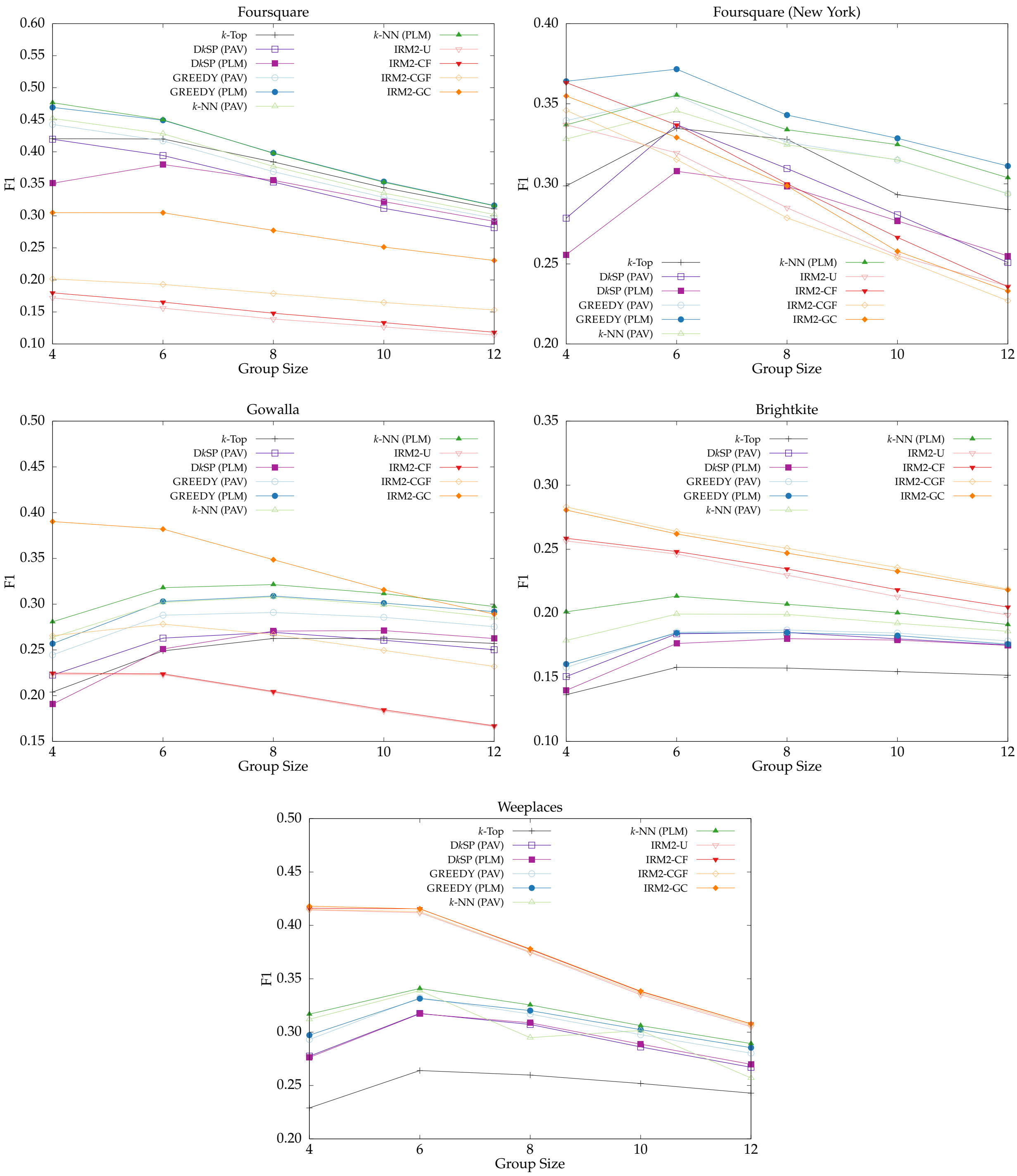

Figure 6: Values of F-measure of the different algorithms w.r.t. the ground-truth groups on the five datasets.

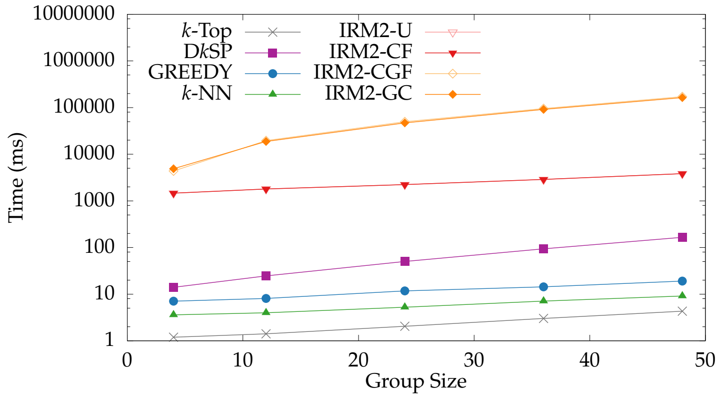

Figure 7: Average execution time per recommendation using k-Top, k-NN, Greedy, DkSP and IRM2 algorithms as a function of group size k on the Foursquare (New York) dataset.

phenomenon rely on the robustness of graph-based approaches in capturing group dynamics analyzing the user-item ego network. In contrast, the original formulation of IRM2 does not consider social relationships among users [5]. We introduced this information—the friendship relationships—into IRM2 by defining novel user prior estimators. Since GREEDY and k-NN outperformed IRM2 on Foursquare datasets, which are the collections with the densest social connections, we believe that there is still room for improvement in the formulation of prior probability estimators that model social information.

Additionally, note that GREEDY exploits the user-item relevance scores computed by a content-based technique meanwhile IRM2 follows a collaborative filtering approach. This result is consistent with the literature from the field of Recommender Systems: content-based approaches tend to work better on sparse collections whereas collaborative filtering algorithms perform very well on less sparse datasets [42]. Moreover, previous work has shown that IRM2 is a probabilistic collaborative filtering technique that tends to work better on dense datasets compared to other similar probabilistic approaches [43].

4.6. Efficiency

In this section, we report the results of an experimental evaluation of the computational efficiency of k-Top, DkSP, k-NN, Greedy and IRM2 algorithms. The purpose of this evaluation is to assess the scalability of these algorithms in real-world applications.

We run these experiments on a single machine using a single-thread implementation. The machine has two Intel E5620 @ 2.4GHz and 108 GB of RAM. We run the experiments 5 times and we report the average running time per user-item recommendation when varying the size of the groups. Although we think that it is unlikely that users demand recommendations that involve large groups of people, we vary the group sizes from 4 to 48 members to study the scalability of our proposals. Figure 7 shows the measured times on the Foursquare (New York) dataset. For the sake of space, we do not report the results on the other datasets as they present similar trends. Apart from finding the same efficiency trends, we found that the magnitude of difference in time is proportional to the size of the dataset measured in terms of the number of check-ins, ratings and social links.

The results show that the most efficient algorithm is k-Top since this technique does not exploit the relationships among users. The other baseline, DkSP, exhibits acceptable execution times with an average runtime lower than 100 milliseconds for values of k up to 48. DkSP uses three procedures to compute its solution and this process affects its efficiency compared to k-NN and GREEDY. In fact, our graph-based proposed solutions show low running times as they need around 10 milliseconds for forming a group.

In contrast, IRM2 is the most expensive technique depending on the size of the group k and the prior estimate. We report the time needed to form a group of size k without any precomputed data so to provide the reader with the the worst case scenario for the efficiency. It is worth highlighting that the performance of IRM2 can be notably improved by caching the computation of similar items. Experiments reveal that the use of CGF and GC priors requires substantially more computing time than U or CF priors because priors should be recomputed after we add a new candidate user to the group. Moreover, the experimentation explores a wide range of values for k. For values of k that are crucial for real-world applications, i.e., from 4 to 8 members per-group, IRM2 with U and CF priors requires around 1 second to form a group. In addition, when using the CGF and GC priors, IRM2 needs from 4 to 10 seconds to build groups of the same kind. As we saw in the previous sections, those priors provided huge improvements in terms of effectiveness in three out of five datasets. To conclude, on the one hand, GREEDY and k-NN algorithms are very fast and provide good results on sparse datasets with good social network information. On the other hand, with more check-ins or rating data, IRM2 may provide much better results at the expense of an increase in the computational cost.

5. Conclusions and Future Work

Finding the best group of companions with whom to enjoy an item or a destination is the motivation that inspired our novel recommendation task. The definition of such novel task, formalized as User-Item Group Formation problem, poses several theoretical challenges and provides important practical implications. UI-GF differs from traditional group recommendation and group formation tasks since it asks for maximizing two orthogonal aspects: i) the relevance of the recommended item for every member of the group, and ii) the intra-group social relationships. In our formulation, we focused on the maximization of the *satisfaction* of the members of the group without constraining the scope of the concept of *satisfaction* on purpose. Our aim was in fact to easy the definition of different models addressing this problem. In particular we proposed two different models for the UI-GF task. Our first model uses a graph-based technique that exploits the user-item ego network to maximize the group satisfaction by means of two different measures (pairwise aggregated voting and pairwise least misery). The second one employs probabilistic collaborative filtering which is able to model the concept of group satisfaction through different user priors.

Another contribution of the paper is the definition of an evaluation methodology for assessing the performance of the proposed solutions. To this end, we designed a methodology based on publicly available data from location-based social networks. From that data, we extract ground-truth groups that enable us to assess the quality of the recommendations by using traditional information retrieval metrics such as Precision, Recall and F-measure. We evaluated both algorithmic proposals by using five publicly available LBSN datasets built with the above methodology. The results of extensive experiments showed that our solutions outperform the baselines and are able to effectively find groups of friends who can jointly appreciate a suggested location. Graph-based algorithms (GREEDY and k-NN) yielded the best results on the Foursquare datasets while, on the other collections employed, the relevance modeling approach (IRM2) provided better solutions at the expense of an increased computational cost. In general, when we have high-quality social data and sparse rating or check-in feedback, GREEDY and k-NN tend to work better. In contrast, if we have large amounts of ratings or check-ins, IRM2 may provide superior recommendations. In terms of efficiency, results confirm that the both the proposed techniques can be applied in real-world scenarios.

Our work has practical implications and interesting applications from an industrial point of view. To the best of our knowledge, we are in fact not aware of any commercial service that suggests groups of friends with whom enjoying a recommended/purchased item. We believe that our techniques are very general and flexible and can be easily adapted to different domains where this kind of additional recommendation service can be provided, e.g., e-commerce, multimedia streaming, e-tourism, etc.

This work opens the way for further research. For example, it would be interesting to automatically compute the optimal size of the recommended group. One approach to address this task consists in modeling the UI-GF task as an instance of the well-known "densest at most k-subgraph" problem. It would be also interesting to investigate an extension of the probabilistic model of IRM2 to estimate automatically the optimal value of k. This is a more complex formulation which also needs more experiments in real world applications. Additionally, we envision to study the suitability of different pairwise similarities for the graph-based algorithms as well as design new prior estimates for IRM2. Finally, the identification of proper index structures to speed up the computation of the solution of UI-GF and the extension of our recommendation problem to other scenarios in addition to LBSNs deserves to be investigated.

Acknowledgments

Franco Maria Nardini, Raffaele Perego, and Chiara Renso acknowledge the support of the projects H2020-INFRAIA-2014-2015 SoBigData (Grant no. 654024) and the Marie Slodowska-Curie MASTER (Grant no. 777695) that received funding from the European Union's Horizon 2020 research and innovation programme. Daniel Valcarce acknowledges the support of the grant FPU014/01724 (MECD) and projects TIN2015-64282-R (MINECO/ERDF), GPC ED431B 2016/035 (Xunta de Galicia) and ED431G/01 (Xunta de Galicia/ERDF).

References

[1] X. Ning, C. Desrosiers, G. Karypis, A Comprehensive Survey of Neighborhood-Based Recommendation Methods, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, 2nd Edition, Springer US, Boston, MA, USA, 2015, pp. 37–76. doi:10.1007/978-1-4899-7637-6_2.

[2] M. de Gemmis, P. Lops, C. Musto, F. Narducci, G. Semeraro, Semantics-Aware Content-Based Recommender Systems, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, 2nd Edition, Springer US, Boston, MA, USA, 2015, pp. 119–159. doi:10.1007/978-1-4899-7637-6_4.

[3] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, C. Yu, Group recommendation: Semantics and efficiency, Proceedings of the VLDB Endowment 2 (1) (2009) 754–765. doi:10.14778/1687627.1687713.

[4] I. Brilhante, F. M. Nardini, R. Perego, C. Renso, J. Macedo, Group finder: an item-driven group formation framework, in: Proceedings of the 17th IEEE International Conference on Mobile Data Management, MDM '16, IEEE Computer Society, 2016, pp. 8–17. doi:10.1109/MDM.2016.16.

[5] D. Valcarce, J. Parapar, A. Barreiro, Item-based relevance modelling of recommendations for getting rid of long tail products, Knowledge Based Systems 103 (C) (2016) 41–51. doi:10.1016/j.knosys.2016.03.021.

[6] S. Basu Roy, L. V. Lakshmanan, R. Liu, From group recommendations to group formation, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, SIGMOD '15, ACM, New York, NY, USA, 2015, pp. 1603–1616. doi:10.1145/2723372.2749448.

[7] J. Masthoff, Group Recommender Systems: Aggregation, Satisfaction and Group Attributes, Springer US, Boston, MA, 2015, pp. 743–776. doi:10.1007/978-1-4899-7637-6_22.
URL https://doi.org/10.1007/978-1-4899-7637-6{_}22

[8] F. Ortega, J. Bobadilla, A. Hernando, A. Gutiérrez, Incorporating group recommendations to recommender systems: Alternatives and performance, Information Processing & Management 49 (4) (2013) 895–901. doi:10.1016/j.ipm.2013.02.003.

[9] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, W. Cao, Deep modeling of group preferences for group-based recommendation, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14, AAAI Press, 2014, pp. 1861–1867.

[10] I. Garcia, L. Sebastia, E. Onaindia, On the design of individual and group recommender systems for tourism, Expert Systems with Applications 38 (6) (2011) 7683–7692. doi:10.1016/j.eswa.2010.12.143.

[11] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, K. Seada, Enhancing group recommendation by incorporating social relationship interactions, in: Proceedings of the 16th ACM International Conference on Supporting Group Work, GROUP '10, ACM, New York, NY, USA, 2010, pp. 97–106. doi:10.1145/1880071.1880087.

[12] S. Amer-Yahia, B. Omidvar-Tehrani, S. Basu Roy, N. Shabib, Group Recommendation with Temporal Affinities, in: Proceedings of the 18th International Conference on Extending Database Technology, EDBT '15, 2015, pp. 421–432. doi:10.5441/002/edbt.2015.37.

[13] O. Kaššák, M. Kompan, M. Bieliková, Personalized hybrid recommendation for group of users, Information Processing & Management 52 (3) (2016) 459–477. doi:10.1016/j.ipm.2015.10.001.

[14] M. S. Pera, Y.-K. Ng, A group recommender for movies based on content similarity and popularity, Information Processing & Management 49 (3) (2013) 673–687. doi:10.1016/j.ipm.2012.07.007.

[15] A. Anagnostopoulos, R. Atassi, L. Becchetti, A. Fazzone, F. Silvestri, Tour recommendation for groups, Data Mining and Knowledge Discovery (2016) 1–32. doi:10.1007/s10618-016-0477-7.

[16] S. J. G. Chen, L. Lin, Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering, IEEE Transactions on Engineering Management 51 (2) (2004) 111–124. doi:10.1109/TEM.2004.826011.

[17] A. Baykasoglu, T. Dereli, S. Das, Project team selection using fuzzy optimization approach, Cybernetics and Systems 38 (2) (2007) 155–185. doi:10.1080/01969720601139041.

[18] R. Andersen, K. Chellapilla, Finding dense subgraphs with size bounds, in: Proceedings of the 6th International Workshop on Algorithms and Models for the Web-Graph, WAW '09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 25–37. doi:10.1007/978-3-540-95995-3_3.

[19] T. Lappas, K. Liu, E. Terzi, Finding a team of experts in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, ACM, New York, NY, USA, 2009, pp. 467–476. doi:10.1145/1557019.1557074.

[20] M. Sozio, A. Gionis, The community-search problem and how to plan a successful cocktail party, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, ACM, New York, NY, USA, 2010, pp. 939–948. doi:10.1145/1835804.1835923.
[21] M. Coscia, F. Giannotti, D. Pedreschi, A classification for community discovery methods in complex networks, Statistical Analysis and Data Mining 4 (5) (2011) 512–546. doi:10.1002/sam.10133.
[22] D. H. Lee, P. Brusilovsky, Improving personalized recommendations using community membership information, Information Processing & Management 53 (5) (2017) 1201–1214. doi:10.1016/j.ipm.2017.05.005.
[23] D.-N. Yang, C.-Y. Shen, W.-C. Lee, M.-S. Chen, On socio-spatial group query for location-based social networks, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, ACM, New York, NY, USA, 2012, pp. 949–957. doi:10.1145/2339530.2339679.
[24] W. Liu, W. Sun, C. Chen, Y. Huang, Y. Jing, K. Chen, Circle of friend query in geo-social networks, in: Proceedings of the 17th International Conference on Database Systems for Advanced Applications - Volume Part II, DASFAA'12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 126–137. doi:10.1007/978-3-642-29035-0_9.
[25] I. Brilhante, J. A. Macedo, F. M. Nardini, R. Perego, C. Renso, Where shall we go today?: Planning touristic tours with tripbuilder, in: Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management, CIKM '13, ACM, New York, NY, USA, 2013, pp. 757–762. doi:10.1145/2505515.2505643.
[26] D. Valcarce, J. Parapar, A. Barreiro, Additive smoothing for relevance-based language modelling of recommender systems, in: Proceedings of the 4th Spanish Conference on Information Retrieval, CERI '16, ACM, New York, NY, USA, 2016, pp. 9:1–9:8. doi:10.1145/2934732.2934737.
[27] M. O'Connor, D. Cosley, J. A. Konstan, J. Riedl, Polylens: A recommender system for groups of users, in: Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work, ECSCW'01, Kluwer Academic Publishers, Norwell, MA, USA, 2001, pp. 199–218.
[28] A. Jameson, B. Smyth, Recommendation to groups, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 596–627.
[29] Y. Asahiro, K. Iwama, H. Tamaki, T. Tokuyama, Greedily finding a dense subgraph, Journal of Algorithms 34 (2) (2000) 203–221. doi:10.1006/jagm.1999.1062.
[30] M. Charikar, Greedy approximation algorithms for finding dense components in a graph, in: Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization, APPROX '00, Springer-Verlag, London, UK, UK, 2000, pp. 84–95.
[31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, Third Edition, 3rd Edition, The MIT Press, 2009.
[32] V. Lavrenko, W. B. Croft, Relevance based language models, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01, ACM, New York, NY, USA, 2001, pp. 120–127. doi:10.1145/383952.383972.
[33] A. Bellogín, J. Wang, P. Castells, Bridging memory-based collaborative filtering and text retrieval, Information Retrieval 16 (6) (2013) 697–724. doi:10.1007/s10791-012-9214-z.
[34] D. Valcarce, Exploring statistical language models for recommender systems, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, ACM, New York, NY, USA, 2015, pp. 375–378. doi:10.1145/2792838.2796547.
[35] J. Parapar, A. Bellogín, P. Castells, A. Barreiro, Relevance-based language modelling for recommender systems, Information Processing & Managemnent 49 (4) (2013) 966–980. doi:10.1016/j.ipm.2013.03.001.
[36] D. Valcarce, J. Parapar, A. Barreiro, A study of priors for relevance-based language modelling of recommender systems, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, ACM, New York, NY, USA, 2015, pp. 237–240. doi:10.1145/2792838.2799677.
[37] J. J. Levandoski, M. Sarwat, A. Eldawy, M. F. Mokbel, Lars: A location-aware recommender system, in: Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, ICDE '12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 450–461. doi:10.1109/ICDE.2012.54.
[38] M. Sarwat, J. J. Levandoski, A. Eldawy, M. F. Mokbel, Lars*: An efficient and scalable location-aware recommender system, IEEE Transactions on Knowledge and Data Engineering 26 (6) (2014) 1384–1399. doi:10.1109/TKDE.2013.29.
[39] E. Cho, S. A. Myers, J. Leskovec, Friendship and mobility: User movement in location-based social networks, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, ACM, New York, NY, USA, 2011, pp. 1082–1090. doi:10.1145/2020408.2020579.
[40] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval: The Concepts and Technology Behind Search, Addison Wesley, 2011.
[41] U. Feige, D. Peleg, G. Kortsarz, The dense k-subgraph problem, Algorithmica 29 (3) (2001) 410–421. doi:10.1007/s004530010050.
[42] F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, 2nd Edition, Springer US, Boston, MA, USA, 2015. doi:10.1007/978-1-4899-7637-6.
[43] D. Valcarce, J. Parapar, Á. Barreiro, Efficient Pseudo-Relevance Feedback Methods for Collaborative Filtering Recommendation, in: Proceedings of the 38th European Conference on Information Retrieval, ECIR '16, Springer International Publishing, Berlin, Heidelberg, 2016, pp. 602–613. doi:10.1007/978-3-319-30671-1_44.