

# Recommending New Items to Ephemeral Groups Using Contextual User Influence

Elisa Quintarelli  
Politecnico di Milano – Italy  
elisa.quintarelli@polimi.it

Emanuele Rabosio  
Politecnico di Milano – Italy  
emanuele.rabosio@polimi.it

Letizia Tanca  
Politecnico di Milano – Italy  
letizia.tanca@polimi.it

## ABSTRACT

Group recommender systems help groups of users in finding appropriate items to be enjoyed together. Lots of activities, like watching TV or going to the restaurant, are intrinsically group-based, thus making the group recommendation problem very relevant. In this paper we study ephemeral groups, i.e., groups where the members might be together for the first time. Recent approaches have tackled this issue introducing complex models to be learned offline, making them unable to deal with new items; on the contrary, we propose a group recommender able to manage new items too. In more detail, our technique determines the preference of a group for an item by combining the individual preferences of the group members on the basis of their *contextual influence*, where the contextual influence represents the ability of an individual, in a given situation, to direct the group's decision. We conducted an extensive experimental evaluation on a TV dataset containing a log of viewings performed by real groups, showing how our approach outperforms the comparable techniques from the literature.

## Keywords

Group recommendations; influence; context

## 1. INTRODUCTION

Recommender systems suggest items to users [1], and are fundamental in all the countless scenarios where people need to be helped in choosing something. For instance, the users may need to choose a movie to watch at the cinema, a TV program, a restaurant for a dinner, or a book to read. In addition, it is clear that lots of such activities are inherently social, generally carried out by *groups* of users: indeed, people seldom go to the movies or watch TV on their own, and equally rarely eat out alone. Even the choice of a book may be a group activity, if this book is then to be discussed and analyzed by an (online) community. Last, but not least, touristic activities are often organized for groups, thus group recommendations may provide tour organizers with useful

suggestions to compose the daily schedule. Hence the importance of studying *group recommender systems* [27], taking into account the possibility of having multiple users involved in the item choices.

The literature on group recommender systems deals with two different kinds of groups: *persistent* and *ephemeral* [27]. Persistent groups have a significant history of activities together, while ephemeral ones may be constituted by people together for the first time. Since for persistent groups a history of past activities is available, in such a scenario classic recommendation techniques for individual users are employable, treating the whole group as one user. When ephemeral groups are considered, on the contrary, no group history is available, and therefore the group preferences must be computed on the basis of those known for the individual users composing the group. The aim of this work is tackling the more general scenario of ephemeral groups.

To generate ephemeral group recommendations, most approaches propose aggregation functions deriving the preferences of the group by appropriately combining those of the component users, computed exploiting recommendation techniques for individuals. Sometimes this combination is performed using weights based on complex user and item models built from the available feedback provided by the users themselves. The necessity of building these models, however, makes the latter techniques unsuitable for the scenarios characterized by many items which are new in the system, i.e., not associated with past feedback. Since in this work we want to study an approach applicable also to new items, we concentrate on aggregation functions allowing to take them into account. Note that the ability of dealing with new items is essential in several realms, think for instance of the TV domain: many TV programs, e.g. the movies, are often broadcast once and then not any more for a long time, and therefore cannot be associated with past user feedback.

Another interesting point [38] is that, when aggregating the preferences of the members of a group, not all the users should have the same weight: some users might indeed have a greater *influence* in determining the final decision. Consider for instance a family with parents and children in front of the TV: clearly, the parents may be prone to favor the children's wishes in deciding the program to be watched, thus making the children more influential in the decision process.

However, even the latter consideration may be debatable. Indeed, this behavior is likely in the early afternoon, when most children programs are available, while in the late afternoon many other kinds of programs might be chosen and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

RecSys '16, September 15-19, 2016, Boston, MA, USA

© 2016 ACM. ISBN 978-1-4503-4035-9/16/09...\$15.00

DOI: <http://dx.doi.org/10.1145/2959100.2959137>

thus the parents should be more influential, in order to exclude those programs that are inappropriate for children.

For this reason, in this work we propose a novel aggregation function, taking influence into account but considering *contextual influence*, because influence weights may vary with respect to the situation, i.e., the context, that the group is currently living.

### Running Example.

We consider a running example in the TV domain, with users watching TV programs possibly in groups. There are two possible contexts, i.e., *daytime* and *night*, and four users, i.e., Alice (*a*), Bob (*b*), Mark (*m*) and Susan (*s*). Alice has a very strong personality, and is willing to watch her favorite programs in any case. On the contrary, Bob is particularly interested in choosing the programs during daytime, but does not like to create problems in the choice during the evening. Finally, Mark and Susan are more permissive.

### Paper Structure.

The paper is structured as follows. Section 2 presents the related work, while Section 3 explains our recommendation technique. Section 4 describes the experimental evaluation and, finally, Section 5 concludes the paper.

## 2. RELATED WORK

The problem of group recommendations has been widely studied in the last years, since the Polylens system [27] was proposed by O'Connor et al. in 2001. Recommendation techniques to groups, as mentioned in the introduction, can be divided into two categories: recommendations to persistent groups and to ephemeral groups. In the former case it is possible to assume the availability of feedback provided by the group as a whole, while in the latter one group feedback is not available and the preferences of a group must be computed on the basis of those of its members.

Chen et al. [8] propose a recommendation technique aggregating the scores of the group members through a weighted average, where the weights are computed exploiting a genetic algorithm; the fitness function of the algorithm relies on known group preferences, so it is implementable only for persistent groups. Vildjiounaite et al. [34] in the TV domain consider the families as fixed groups, and build a classifier for each family. Seko et al. [29] compute the scores for the items to be recommended to a group exploiting the similarities with the items belonging to the group history. Hu et al. [19] devise a complex model based on restricted Boltzmann machines, learned starting from explicit ratings provided by the groups. We do not consider these approaches, because – as stated in the introduction – in this paper we set ourselves in the more general scenario of ephemeral groups.

Recommendations to ephemeral groups are performed composing the preferences of the group members by using aggregation functions, the most famous being *average* and *least misery*. The average [24, 36, 31, 4, 30, 6, 26, 7, 2, 12, 21, 17] is the simplest aggregation function: it computes the group preference for an item just as the arithmetic mean of those of the group members. Least misery [27, 24, 4, 30, 6, 26, 33, 18, 7, 12, 21, 17], on the contrary, does not maximize the average satisfaction, but tries to make everyone happy, choosing the items for which the lowest value among the preferences of the group members is the great-

est. Other popular aggregation functions are the *maximum satisfaction* [24, 6, 26, 7, 12, 21], maximizing the greatest preference, and average without misery [24, 32, 12], computing an average but neglecting the items whose preference is below a certain threshold. Works [5, 16, 2] aggregate the group members' preferences through a weighted average, where the weight represents the expertise, computed on the basis of the amount of activity performed by the user in the system. Amer-Yahia et al. [3] introduce an interesting function which, on the one hand, tries to maximize the satisfaction of the group components, but, on the other hand, also tries to minimize the disagreement among them. In the experimental section we carry out comparisons with these techniques. Also the approach of De Campos et al. [11], building a bayesian network and using it to learn both the individual preferences and the aggregation weights, is worth mentioning; this technique, however, needs the construction of a complex model including the items, and therefore cannot be used to recommend new items because these are not part of the model.

The importance of influence in group recommenders has first been noticed by Masthoff and Gatt [25]. They assume the presence of two kinds of influence: *emotional contagion* is the influence of the affective state of a user on the other group members, while *conformity* regards the tendency of a user to adapt to the choices of the others. These factors are incorporated in the group preference, and their values are derived from prior knowledge of the relationships among the group members. Quijano-Sanchez et al. [28] consider *personality* and *trust* to aggregate individual preferences: the personality factor assigns greater weights to the users with stronger personality, while the trust is used to modify the group members' preferences towards those of the users in whom they trust. The personality factor is determined on the basis of psychological tests, while the trust requires social network information. Also Kompan and Bielikova [22] propose a group recommender taking into account the personality, this time represented through a weighted graph indicating how the users influence each other; in this case the computation of the influence values requires the users to undergo psychological tests as well. Christensen and Schiaffino [9] incorporate in their formulas three social factors, named *trust*, *social similarity* and *social centrality*. The first depends on the level of acquaintance between the users and must be declared, while the second and the third rely on the activities in social networks. These approaches exploit social factors, but require to have prior knowledge of their values. On the contrary, our methodology computes influences only on the basis of training data.

Some approaches build complex models of users and items employed to compute both the individual preferences and the influence values to be used in the aggregation. Ye et al. [35] propose a probabilistic model learned with an expectation-maximization procedure, while Liu et al. [23] and Yuan et al. [37] found their methodologies on the latent Dirichlet allocation. Finally, Kim and El Saddik [20] build a graph for each group containing users and items, and use random walks with restarts to compute affinities between users and items, and the influence of the users within the groups. These techniques are not suitable for new items, because they cannot be included in the learned model and thus cannot be recommended.

In this work we propose an aggregation function to com-

pute group preferences starting from individual ones and exploiting the users' influences in a way suitable also for new items. In addition, our influence values are computed relying only on training data, without the need of asking information to the users. Finally, to the best of our knowledge, this is the first proposal defining the influence in a context-aware manner.

### 3. RECOMMENDATION METHODOLOGY

This section describes our proposed group recommendation methodology. The methodology starts with the individual preferences of the users for the items, and provides techniques to aggregate them to obtain group preferences. We set ourselves in the more general framework of context-aware recommender systems, and so suppose that the preference of a user for an item may vary according to the context. In addition, we suppose to know also regular, context-unaware preferences of the users for the items. Users' contextual preferences can be computed through an existing context-aware recommender system for individual users, while to compute context-unaware preferences a traditional one may be employed. We also suppose to have a log recording the history of the items chosen in the past by groups of users. To be more general, we assume that the set of the available items may change depending on the time instant in which it is requested. This is the case of the TV setting of the running example, but also of other scenarios like event recommendations or recommending movies in theaters now.

**PROBLEM STATEMENT.** Let  $\mathcal{U}$  be a set of users,  $\mathcal{I}$  a set of items,  $\mathcal{C}$  a set of contexts. Supposing that the following data are known:

- A context-aware scoring function  $score\_ctx(u, i, c)$ ,  $u \in \mathcal{U}$ ,  $i \in \mathcal{I}$ ,  $c \in \mathcal{C}$ , assigning a score to the items in the various contexts
- A context-unaware scoring function  $score\_no\_ctx(u, i)$ ,  $u \in \mathcal{U}$ ,  $i \in \mathcal{I}$ , assigning a score independent of the context to the items
- A log  $\mathcal{L}$  recording the history of the items chosen by the groups, in the form of 4-ples  $l_j : (t_j, c_j, G_j, i_j)$  with  $c_j \in \mathcal{C}$ ,  $G_j \in \wp(\mathcal{U})$  and  $i_j \in \mathcal{I}$ ;  $t_j$  is the time instant in which the item has been chosen

Then, given a target group  $G \in \wp(\mathcal{U})$ , a context  $c \in \mathcal{C}$  and a time instant  $t$ , the problem of group recommendation is defined as recommending a list of  $K$  items from those in  $\mathcal{I}$  which are available at time instant  $t$  that users in  $G$  may be interested in when in context  $c$ .

**EXAMPLE 1 (LOG).** Table 1 shows a possible log in our TV running example. Each row indicates the time instant, the context, the composition of the group and the chosen item. Since in our scenario, in a given time instant, only a subset of the items is available, i.e., the programs that are on the air, for convenience each row reports also the items that are available in the corresponding time instant.

The group preference for an item is determined by aggregating the individual preferences of the group members on the basis of their influence. The influence of a given user  $u$

Table 1: Log in our running example

| Time  | Context | Group         | Chosen item | Available items              |
|-------|---------|---------------|-------------|------------------------------|
| $t_1$ | daytime | $\{a, b, m\}$ | $p_1$       | $\{p_1, p_2, p_3\}$          |
| $t_2$ | daytime | $\{a, m, s\}$ | $p_5$       | $\{p_4, p_5, p_6\}$          |
| $t_3$ | daytime | $\{b, s\}$    | $p_8$       | $\{p_7, p_8, p_9\}$          |
| $t_4$ | daytime | $\{a, m\}$    | $p_{11}$    | $\{p_{10}, p_{11}, p_{12}\}$ |
| $t_5$ | night   | $\{a, b, s\}$ | $p_{15}$    | $\{p_{13}, p_{14}, p_{15}\}$ |
| $t_6$ | night   | $\{b, s\}$    | $p_{16}$    | $\{p_{16}, p_{17}, p_{18}\}$ |
| $t_7$ | night   | $\{a, b\}$    | $p_{21}$    | $\{p_{19}, p_{20}, p_{21}\}$ |

is derived by comparing the behavior of  $u$  when alone represented by his/her individual preferences with his/her behavior when in groups represented by the interactions contained in the log  $\mathcal{L}$ . In more detail, the influence of  $u$  evaluates how many times the groups containing  $u$  have selected one of his/her favorite items. Let  $TopK(u, c, t)$  be the list of the  $K$  preferred items of user  $u$  in context  $c$  according to the values of  $score\_ctx(u, i, c)$  for the various  $i \in \mathcal{I}$  available at instant  $t$ ; in the TV domain of our running example, the  $TopK$  at a certain time instant must include just programs on the air at that time instant. The contextual influence is defined as follows:

$$infl\_ctx(u, c) = \frac{|l_j \in \mathcal{L} : c = c_j \wedge u \in G_j \wedge i_j \in TopK(u, c, t_j)|}{|l_j \in \mathcal{L} : c = c_j \wedge u \in G_j|} \quad (1)$$

The value of  $infl\_ctx(u, c)$  quantifies the ability of user  $u$  to direct the group's decision towards his/her own tastes while in context  $c$ .

**EXAMPLE 2 (CONTEXTUAL INFLUENCE).** Suppose we compute the influences using  $K=1$  in Equation (1), i.e., using top-1 lists, and suppose that at the time instants of Example 1 the programs preferred by the users are the following:

- $t_1: a \rightarrow p_2, b \rightarrow p_1, m \rightarrow p_2$
- $t_2: a \rightarrow p_5, m \rightarrow p_5, s \rightarrow p_4$
- $t_3: b \rightarrow p_8, s \rightarrow p_9$
- $t_4: a \rightarrow p_{11}, m \rightarrow p_{10}$
- $t_5: a \rightarrow p_{15}, b \rightarrow p_{13}$
- $t_6: b \rightarrow p_{16}, s \rightarrow p_{16}$
- $t_7: a \rightarrow p_{21}, b \rightarrow p_{20}$

The contextual influence computations for users Alice, Bob and Mark are as follows:

$$\begin{aligned} infl\_ctx(a, daytime) &= 2/3 = 0.667 \\ infl\_ctx(b, daytime) &= 2/2 = 1 \\ infl\_ctx(m, daytime) &= 1/3 = 0.333 \\ infl\_ctx(a, night) &= 2/2 = 1 \\ infl\_ctx(b, night) &= 1/3 = 0.333 \end{aligned}$$

The value of  $infl\_ctx(m, night)$  cannot be computed, because Mark provided no feedback in the context night.

The context-unaware influence of a user can be computed as the context-aware one, but considering the whole log instead of the contextual:

$$infl\_no\_ctx(u) = \frac{|l_j \in \mathcal{L} : u \in G_j \wedge i_j \in TopK(u, c_j, t_j)|}{|l_j \in \mathcal{L} : u \in G_j|} \quad (2)$$

EXAMPLE 3 (NON-CONTEXTUAL INFLUENCE). *The non-contextual influence is a single value for each user, and is computed exploiting the whole log:*

$$\begin{aligned} \text{influence\_no\_ctx}(a) &= 4/5 = 0.8 \\ \text{influence\_no\_ctx}(b) &= 3/5 = 0.6 \\ \text{influence\_no\_ctx}(m) &= 1/3 = 0.333 \end{aligned}$$

Given a group  $G \in \wp(\mathcal{U})$ , its preference  $\text{score\_ctx}(G, i, c)$  for  $i \in \mathcal{I}$  in the context  $c \in \mathcal{C}$  is computed as the mean of the preferences of its members weighted on the basis of each member’s influence in context  $c$ :

$$\text{score\_ctx}(G, i, c) = \frac{\sum_{u \in G} \text{infl\_ctx}(u, c) \cdot \text{score\_ctx}(u, i, c)}{\sum_{u \in G} \text{infl\_ctx}(u, c)} \quad (3)$$

If the number of contexts is high, it may well happen that some users are not associated with feedback in some contexts. In order to deal with such cases, it is possible to compute also a non-contextual score for the items:

$$\text{score\_no\_ctx}(G, i) = \frac{\sum_{u \in G} \text{infl\_no\_ctx}(u) \cdot \text{score\_no\_ctx}(u, i)}{\sum_{u \in G} \text{infl\_no\_ctx}(u)} \quad (4)$$

Algorithm 1 summarizes the way we compute the top-K list of items preferred by a certain group  $G$  in a context  $c$  at time instant  $t$ . If all the members of  $G$  have provided feedback in context  $c$ , then context-aware scores are computed using Equation (3); otherwise, a non-contextual score is obtained using Equation (4).

Note that a first alternative to determine the group preferences for contexts when some of the group members have not provided feedback was to apply Equation (3) anyway, inserting non-contextual values of preference and influence just for those users for whom the contextual ones are not known, thus mixing context-aware and context-unaware quantities in the computation. However, this strategy, which “blends” values measured on different subsets of the log, showed a bad behavior in the experiments. We then did a manual check, sampling users from our dataset and controlling how much their influence varies according to context, and how far these values are from their non-contextual influence. Often, non-contextual influence turned out to be quite distant from the contextual one, thus we explain this somehow unexpected behavior with the fact that putting together contextual and non-contextual influences creates a sort of “unbalanced situation” among the members of the group, which reflects reality worse than keeping the weights uniform.

As a consequence, we decided to adopt the technique described in Algorithm 1.<sup>1</sup>

EXAMPLE 4 (SCORE COMPUTATION). *Suppose that at instant  $t_8$ , during daytime, a group composed of Bob and Mark is watching TV, and it is needed to recommend one program among the available  $p_{22}$ ,  $p_{23}$  and  $p_{24}$ . Note that this group is never recorded in the log: it is the first time that these two users watch TV together without other people. The following individual preference values are given:  $\text{score\_ctx}(b, p_{22}$ ,*

---

**Algorithm 1** Top-K computation for a group of users

---

```

1: function COMPUTETOPK( $G \in \wp(\mathcal{U})$ ,  $c \in \mathcal{C}$ , time instant  $t$ )
2:   if all users in  $G$  provided feedback in context  $c$  then
3:     USE_CONTEXT = true
4:   else
5:     USE_CONTEXT = false
6:   end if
7:   for all  $i \in \mathcal{I}$  do
8:     if USE_CONTEXT then
9:        $\text{score}(G, i, c) = \text{score\_ctx}(G, i, c)$ 
10:    else
11:       $\text{score}(G, i, c) = \text{score\_no\_ctx}(G, i)$ 
12:    end if
13:  end for
14:  return the  $K$  items  $i \in \mathcal{I}$  with the greatest values of
15:     $\text{score}(G, i, c)$ , among those available at time  $t$ 
16: end function

```

---

*daytime) = 0.8,  $\text{score\_ctx}(b, p_{23}, \text{daytime}) = 0.2$ ,  $\text{score\_ctx}(b, p_{24}, \text{daytime}) = 0.3$ ,  $\text{score\_ctx}(m, p_{22}, \text{daytime}) = 0.2$ ,  $\text{score\_ctx}(m, p_{23}, \text{daytime}) = 0.5$ ,  $\text{score\_ctx}(m, p_{24}, \text{daytime}) = 0.9$ . The group scores for the three available items are computed as weighted averages. Since both the users have provided feedback in the context daytime Algorithm 1 prescribes to employ Equation (3), where we use the influence values computed in Example 2:*

$$\begin{aligned} \text{score\_ctx}(\{b, m\}, p_{22}, \text{daytime}) &= \frac{1 \cdot 0.8 + 0.333 \cdot 0.2}{1 + 0.333} = 0.650 \\ \text{score\_ctx}(\{b, m\}, p_{23}, \text{daytime}) &= \frac{1 \cdot 0.2 + 0.333 \cdot 0.5}{1 + 0.333} = 0.275 \\ \text{score\_ctx}(\{b, m\}, p_{24}, \text{daytime}) &= \frac{1 \cdot 0.3 + 0.333 \cdot 0.9}{1 + 0.333} = 0.450 \end{aligned}$$

*Therefore, the system recommends  $p_{22}$  to the group. Note that the choice is directed by the heavier influence of Bob.*

*Suppose now that at instant  $t_9$ , in the context night, a group with Alice, Bob and Mark is watching TV and needs one recommendation, choosing among  $p_{25}$ ,  $p_{26}$  and  $p_{27}$ . Note that Mark provided no feedback in the context night, and therefore it is not possible to utilize contextual preferences and influences. As described in Algorithm 1, non-contextual preferences and influences are used. The following non-contextual preferences are given:  $\text{score\_no\_ctx}(a, p_{25}) = 0.9$ ,  $\text{score\_no\_ctx}(a, p_{26}) = 0.1$ ,  $\text{score\_no\_ctx}(a, p_{27}) = 0.6$ ,  $\text{score\_no\_ctx}(b, p_{25}) = 0.2$ ,  $\text{score\_no\_ctx}(b, p_{26}) = 0.9$ ,  $\text{score\_no\_ctx}(b, p_{27}) = 0.3$ ,  $\text{score\_no\_ctx}(m, p_{25}) = 0.3$ ,  $\text{score\_no\_ctx}(m, p_{26}) = 0.7$ ,  $\text{score\_no\_ctx}(m, p_{27}) = 0.5$ . The group scores for the three available items are computed using Equation (4) and the non-contextual influences of Example 3:*

$$\begin{aligned} \text{score\_no\_ctx}(\{a, b, m\}, p_{25}) &= \frac{0.8 \cdot 0.9 + 0.6 \cdot 0.2 + 0.333 \cdot 0.3}{0.8 + 0.6 + 0.333} = 0.542 \\ \text{score\_no\_ctx}(\{a, b, m\}, p_{26}) &= \frac{0.8 \cdot 0.1 + 0.6 \cdot 0.9 + 0.333 \cdot 0.7}{0.8 + 0.6 + 0.333} = 0.492 \\ \text{score\_no\_ctx}(\{a, b, m\}, p_{27}) &= \frac{0.8 \cdot 0.6 + 0.6 \cdot 0.3 + 0.333 \cdot 0.5}{0.8 + 0.6 + 0.333} = 0.477 \end{aligned}$$

*Therefore, the system recommends program  $p_{25}$  to the group.*

## 4. EXPERIMENTS

Our methodology was tested using a dataset in the TV domain, containing implicit feedback and context information. The TV scenario is actually a perfect setting for our approach: in fact, watching TV is inherently a group activity. Moreover, many kinds of TV programs – like the movies – are broadcast once and then not anymore for a long time, thus making the new item problem extremely relevant.

We remark that the dataset we employ contains implicit feedback by real groups constituted by people really watching TV together. Several approaches from the literature (e.g., [27, 4]), on the contrary, use datasets containing just individual feedback, like Movielens, and then build groups

<sup>1</sup>Please notice that, in scenarios where efficiency is critical, the online top-K computation, where each item’s score is derived aggregating the preferences of the single users, may be executed by using fast algorithms from the literature, like the Threshold Algorithm proposed by Fagin et al. [15]. Such algorithms would derive the top-K by applying Equations (3) and (4) to just a subset of the items.

**Table 2: Time slots**

| Start time | End time | Description    |
|------------|----------|----------------|
| 02:00:00   | 07:00:00 | Graveyard slot |
| 07:00:00   | 09:00:00 | Early morning  |
| 09:00:00   | 12:00:00 | Morning        |
| 12:00:00   | 15:00:00 | Daytime        |
| 15:00:00   | 18:00:00 | Early fringe   |
| 18:00:00   | 20:30:00 | Prime access   |
| 20:30:00   | 22:30:00 | Prime time     |
| 22:30:00   | 02:00:00 | Late fringe    |

artificially. We found two publicly available datasets envisaging feedback by real groups: Meetup, employed in [23], and Plancast, employed in [37]. Unfortunately, both contain just group feedback, and this makes them unsuitable to test our methodology. The influence computation, indeed, is based on a comparison between the behavior of the users when alone and when in groups, and therefore does need group feedback but requires individual feedback as well.

The preference functions adopted to determine the contextual and non-contextual scores of the items for the single users in the TV dataset are those recently proposed in [10], which represents a state-of-the-art technique to generate TV program recommendations for individual users relying on implicit feedback. The recommendation methodology defined in [10] is content-based, thus allowing to compute users' preference values also for new items.

Note that, as underlined in the running example, in this special domain all the recommendation lists are generated, rather than as subsets of the whole set of items (i.e., available TV programs), just as subsets of the programs aired at the time instant in which the recommendation is required.

## 4.1 Dataset

The employed dataset contains TV viewing information related to 7921 users and 119 channels, broadcast both over the air and by satellite. The dataset is composed of an electronic program guide (EPG) containing the description of 21194 distinct programs, and a log containing both individual and group viewings performed by the users. The attributes available for each program in the EPG are its genre and the channel on which it is transmitted.

The log spans from December 3rd, 2013 to March 1st, 2014 and contains 4968231 entries, among which we retained just the syntonizations longer than three minutes. 3519167 viewings are performed by individual users, and are those used to feed the methodology [10] for the computation of the individual preferences of the group members. The remaining 1449064 viewings envisage more than one person. Each log row specifies the identifier of the user and that of the program he/she watched, along with start time and end time. Start and end times were employed to derive the day of the week and the time slot, which are the context dimensions considered in the experiments; the available values for the time slot are shown in Table 2.

The group viewings were split into a training set, including the syntonizations between December 3rd, 2013 and February 15th, 2014 (1210316 entries), and a test set, containing the remaining ones (238748 entries). The former was used as the log  $\mathcal{L}$  in input to the formulas proposed in this paper, while the latter was employed to assess the quality of the recommendations.

## 4.2 Evaluation Metrics

The performance of our recommendation algorithm was evaluated using *Recall@K*, describing the number of test items included in recommendation lists of length  $K$  computed at the instant in which the viewing of the test items started and in the context in which they were watched.

More formally, let  $v$  be a program view in the test set,  $v_t$  the start time of the view,  $v_G$  the group of users that watched the program,  $v_i$  the program watched and  $v_c$  the context in which the view took place.  $TopK(G, c, t)$  indicates the set of top- $K$  items for the group  $G$  in context  $c$  among those on the air at time instant  $t$ , determined using the recommendation methodology to be evaluated. *Recall@K* is computed as follows:

$$Recall@K = \frac{|v \in \text{Test Set} : v_i \in TopK(v_G, v_c, v_t)|}{|v \in \text{Test Set}|} \quad (5)$$

We executed experiments for  $K=1$ ,  $K=2$  and  $K=3$ .

## 4.3 Compared Methods

The methodology introduced in this paper, from now on dubbed *CtxInfl*, was compared with several other techniques, among those presented in Section 2 which allow to recommend new items to ephemeral groups:

- Average (AVG)
- Least misery (LM)
- Maximum satisfaction (MS)
- The approach of [3], taking into account the disagreement between the group members (DIS)
- Weighted average using the expertise, where the expertise of a user in our experiments is defined as the number of program viewings which he/she performed (EXP)

In addition, our methodology was compared with a version of itself exploiting only non-contextual influences, i.e., computing influences always using Equation (2); this approach is named *Infl*. Such a comparison is important to verify whether the usage of the context to model the influence may actually provide a performance gain.

Besides the evaluation on the whole the test set, we also executed experiments considering different subsets of the test items on the basis of the group size and heterogeneity. About the group sizes, we measured the recall related to the groups containing two (192378 test items), three (35216 test items), four (9739 test items) and five (1151 test items) members. The heterogeneity of the group was measured using the Spearman's footrule distance with location parameter  $\ell$  [14] between top- $K$  lists<sup>2</sup>, employing  $K = 3$  and  $\ell = K + 1$ . The distance was measured between the top-3 lists of favorite items of all the pairs of group members and then averaged. The values of this distance range from 0 if the top-3 lists of the users contain the same items identically ordered, to 12 if the top-3 lists of the users are completely disjoint. We measured the recall related to the groups with average distance  $h$  such that  $h \leq 4$  (46935 test items),  $4 < h \leq 8$  (99333 test items), and  $h > 8$  (92480 test items).

<sup>2</sup>The Spearman's footrule distance is the  $L_1$  distance between two permutations of the elements of a domain. The Spearman's footrule distance with location parameter  $\ell$  is a variation allowing to deal with top- $K$  lists, where all the elements of the domain not included in the top- $K$  list are placed by default at position  $\ell$ .

Finally, in Section 3 we have explained that CtxInfl uses non-contextual preferences and influences for the groups containing at least one member whose feedback in the context of interest is unknown. As a consequence, CtxInfl and Infl apply exactly the same formulas on all the test items  $v$  such that at least one user in  $v_G$  provided no feedback in the context  $v_c$ , thus making scarcely significant the comparison of CtxInfl and Infl on these items. Therefore, to better analyze the relevance of the context in influence modeling we compared CtxInfl and Infl also on further restrictions of the test sets listed above, containing just the test items  $v$  such that all the users in  $v_G$  provided feedback in the context  $v_c$ ; on these test items, CtxInfl never executes the ELSE branch at Line 10 of Algorithm 1. The test sets obtained through this further restriction contain about 3/4 test items with respect to the original ones.

## 4.4 Results

In this subsection we present in tabular form the results obtained in the experiments. Tables 3, 4 and 5 show, respectively, Recall@1, Recall@2 and Recall@3, expressed in percentage, measured on the whole test set and on the test sets restricted, as explained above, on the basis of the group size and heterogeneity. Table 6, on the other hand, compares only CtxInfl and Infl showing Recall@1, Recall@2 and Recall@3 on the test sets further restricted by removing the test items  $v$  such that at least one user in  $v_G$  did not provide feedback in the context  $v_c$ . The tables report on the rows the test set composition, and on the columns the employed technique. Our algorithm CtxInfl and its variant Infl were run setting  $K=1$  in Equations (1) and (2), since this is the value achieving the best results.

## 4.5 Result Analysis

In the following the results reported in the tables are analyzed in detail. Subsection 4.5.1 focuses on the comparison of CtxInfl with the competitors from the literature, while Subsection 4.5.2 examines the differences between CtxInfl and its non-contextual variant.

### 4.5.1 Comparison with the Competitors from the Literature

The comparison starts with the whole test set, and then considers those derived on the basis of the group size and the group heterogeneity.

#### Full Test Set.

The measured recall values highlight how our strategy (CtxInfl) dominates its competitors for all the experimented lengths of the recommendation list. It is also possible to notice that, among the other approaches, the best-performing one is EXP, and this suggests the goodness of the strategies relying on a weighted average with respect to plain average, least misery, maximum satisfaction, and also the approach of [3] taking disagreement into account. Another aspect that can be noticed from the results is that the difference between our methodology and the baselines is larger when recommending few items: 2.81% for Recall@1, 2.29% for Recall@2 and 1.81% for Recall@3. This happens because many users watch just a limited number of channels, and therefore even simple strategies are able to identify the proper program in lists containing more items.

## Results Varying the Group Size.

In these experiments CtxInfl is also the best one, and its main competitor in most cases remains EXP. It is immediately evident that the advantage obtained by CtxInfl grows when the group size is bigger: considering Recall@1, the difference is 2.43% when the group size is 2, 4.31% when the group size is 3, 4.47% when the group size is 4, and 5.65% when the group size is 5. This result is easily interpreted: when groups are larger social factors are more important.

## Results Varying the Group Homogeneity.

Once again, CtxInfl is the leading technique and EXP the second one. The first evidence is that lower heterogeneity corresponds to higher recall; therefore, as it is intuitive, recommending to homogeneous groups is a simpler problem. Moreover, the gain brought by CtxInfl with respect to EXP grows with the heterogeneity of the groups: considering Recall@1, the gain is 1.84% for  $h \leq 4$ , 2.43% for  $4 < h \leq 8$  and 3.68% for  $h > 8$ . Also this outcome is very intuitive: if the group is homogeneous the members have the same tastes, and therefore it is not important who are the most influencing ones.

### 4.5.2 Relevance of the Context in Influence Computation

The results clearly show that adding the context to the influence provides performance improvements, for all the experimented test sets and lengths of the recommendation list. The improvements brought by CtxInfl with respect to Infl are more evident in the results shown in Table 6 than on those shown in Tables 3, 4 and 5, because the former consider just the test items on which the two methodologies apply different formulas. In more detail, in Table 6 we observe that the gain seems to be bigger when the recommendation lists are shorter: considering all the test items  $v$  such that all the members of  $v_G$  provided feedback in the context  $v_c$ , the benefit provided by the context is 1.68% for Recall@1, 1.48% for Recall@2 and 1.20% for Recall@3. In addition, the performance improvement grows with group heterogeneity: considering Recall@1, the gain is 0.56% for  $h \leq 4$ , 1.65% for  $4 < h \leq 8$  and 2.26% for  $h > 8$ .

## 4.6 Accuracy Improvement Evaluation

To validate the statistical significance of the accuracy improvement provided by our algorithm, we used McNemar's test [13], comparing CtxInfl with the other experimented algorithms. The statistical test aims at verifying that the performance differences between two approaches are significant and not just due to chance.

McNemar's test was executed considering the whole test set for Recall@1, Recall@2 and Recall@3. Table 7 shows the values of the McNemar's test statistic obtained by comparing CtxInfl with its competitors: the greater the value of the statistic, the more significant the results. The measured values are very high, corresponding to p-values extremely close to 0 and thus providing strong evidence for rejecting the hypothesis of no improvement. Note that a p-value of 0.05 corresponds to a value 3.8415 of the test statistic.

## 4.7 Summary of the Evaluation

The evaluation assessed the quality of the group recommendation technique proposed in this paper using a dataset in the TV domain, a domain particularly suitable to carry

**Table 3: Recall@1**

| Test set       | CtxInfl | Infl   | AVG    | LM     | MS     | DIS    | EXP    |
|----------------|---------|--------|--------|--------|--------|--------|--------|
| All            | 34.12%  | 32.85% | 30.24% | 28.93% | 28.95% | 30.56% | 31.31% |
| Group size = 2 | 34.81%  | 33.66% | 31.47% | 30.14% | 30.30% | 31.72% | 32.38% |
| Group size = 3 | 31.18%  | 29.39% | 25.49% | 24.93% | 23.95% | 26.21% | 26.87% |
| Group size = 4 | 31.71%  | 29.98% | 23.87% | 20.89% | 21.30% | 24.17% | 27.24% |
| Group size = 5 | 31.89%  | 30.58% | 26.67% | 19.64% | 23.81% | 27.02% | 26.24% |
| $h \leq 4$     | 44.44%  | 44.03% | 42.11% | 41.57% | 41.63% | 42.15% | 42.60% |
| $4 < h \leq 8$ | 36.13%  | 34.88% | 32.50% | 32.03% | 30.43% | 33.04% | 33.70% |
| $h > 8$        | 26.71%  | 24.99% | 21.78% | 19.18% | 20.92% | 22.01% | 23.03% |

**Table 4: Recall@2**

| Test set       | CtxInfl | Infl   | AVG    | LM     | MS     | DIS    | EXP    |
|----------------|---------|--------|--------|--------|--------|--------|--------|
| All            | 51.35%  | 50.24% | 48.47% | 45.87% | 46.35% | 48.77% | 49.06% |
| Group size = 2 | 52.14%  | 51.18% | 49.92% | 47.34% | 47.68% | 50.21% | 50.38% |
| Group size = 3 | 48.29%  | 46.61% | 43.11% | 40.56% | 41.29% | 43.37% | 43.86% |
| Group size = 4 | 47.86%  | 45.93% | 40.55% | 37.00% | 39.17% | 41.00% | 43.14% |
| Group size = 5 | 47.96%  | 45.78% | 42.66% | 42.57% | 42.05% | 42.75% | 43.44% |
| $h \leq 4$     | 64.44%  | 64.27% | 63.63% | 62.25% | 60.46% | 63.35% | 63.84% |
| $4 < h \leq 8$ | 54.17%  | 53.30% | 51.74% | 49.15% | 49.40% | 52.17% | 52.50% |
| $h > 8$        | 41.68%  | 39.82% | 37.26% | 34.03% | 35.91% | 37.61% | 37.87% |

**Table 5: Recall@3**

| Test set       | CtxInfl | Infl   | AVG    | LM     | MS     | DIS    | EXP    |
|----------------|---------|--------|--------|--------|--------|--------|--------|
| All            | 62.48%  | 61.58% | 60.12% | 57.35% | 58.18% | 60.34% | 60.67% |
| Group size = 2 | 63.36%  | 62.61% | 61.48% | 58.71% | 59.58% | 61.73% | 61.90% |
| Group size = 3 | 59.25%  | 57.72% | 55.30% | 52.25% | 52.85% | 55.46% | 56.11% |
| Group size = 4 | 57.88%  | 56.31% | 52.03% | 49.83% | 50.82% | 51.89% | 53.84% |
| Group size = 5 | 58.91%  | 56.21% | 53.87% | 56.21% | 54.21% | 54.39% | 55.52% |
| $h \leq 4$     | 75.25%  | 75.08% | 74.48% | 72.98% | 71.87% | 74.53% | 74.72% |
| $4 < h \leq 8$ | 65.41%  | 64.84% | 63.78% | 60.61% | 61.57% | 63.86% | 64.16% |
| $h > 8$        | 52.86%  | 51.21% | 48.90% | 45.92% | 47.59% | 49.37% | 49.77% |

**Table 6: Recall measured considering only the test items  $v$  such that all the users in  $v_G$  provided feedback in the context  $v_c$** 

| Test set       | Recall@1 |        | Recall@2 |        | Recall@3 |        |
|----------------|----------|--------|----------|--------|----------|--------|
|                | CtxInfl  | Infl   | CtxInfl  | Infl   | CtxInfl  | Infl   |
| All            | 35.58%   | 33.90% | 53.36%   | 51.88% | 64.60%   | 63.40% |
| Group size = 2 | 36.14%   | 34.64% | 54.02%   | 52.77% | 65.35%   | 64.38% |
| Group size = 3 | 33.27%   | 30.78% | 50.65%   | 48.30% | 61.90%   | 59.77% |
| Group size = 4 | 32.90%   | 30.41% | 50.03%   | 47.25% | 59.44%   | 57.18% |
| Group size = 5 | 31.95%   | 30.00% | 50.39%   | 47.14% | 61.95%   | 57.92% |
| $h \leq 4$     | 47.40%   | 46.84% | 68.33%   | 68.08% | 79.06%   | 78.83% |
| $4 < h \leq 8$ | 37.40%   | 35.75% | 56.12%   | 54.98% | 67.44%   | 66.70% |
| $h > 8$        | 27.85%   | 25.59% | 43.07%   | 40.62% | 54.48%   | 52.30% |

**Table 7: Values of the McNemar’s test statistic**

|          | Infl  | AVG    | LM     | MS     | DIS    | EXP    |
|----------|-------|--------|--------|--------|--------|--------|
| Recall@1 | 912.3 | 3870.6 | 2786.5 | 4870.5 | 3001.9 | 2160.4 |
| Recall@2 | 868.9 | 2498.1 | 2775.6 | 4201.3 | 1849.5 | 1628.6 |
| Recall@3 | 765.1 | 1942.4 | 2601.8 | 3092.9 | 1389.5 | 1221.8 |

out this evaluation, because we are interested in new items and in such a domain most of the items are new. The experiments proved that our algorithm outperforms the existing approaches in terms of recall and, in more detail, the performance gain grows with the size and the heterogeneity of the group. The experiments showed also the relevance of em-

ploying the context in the influence computation. Finally, the significance of the improvements was confirmed through a statistical test.

## 5. CONCLUSION

This paper has studied a group recommendation methodology for ephemeral groups, able to recommend also items that are new in the system. The proposed approach derives group preferences by aggregating individual ones relying on the contextual influence, that is the ability of a user, in a given situation, to direct the group’s decision. The influence values are determined exclusively from log data, without

asking information from the users. The effectiveness of the proposed approach has been proven through experiments on a dataset in the TV domain, envisaging real groups.

Interesting developments of the present work regard the analysis of novelty and serendipity problems for group recommendations.

## Acknowledgments

This research has been partially supported by the IT2Rail project funded by European Union's Horizon 2020 research and innovation program under grant agreement No: 636078, and by the Italian project SHELL CTN01-00128.111357.

## 6. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.
- [2] I. Ali and S. Kim. Group recommendations: Approaches and evaluation. In *Proc. IMCOM*, page 105. ACM, 2014.
- [3] S. Amer-Yahia, S. Basu Roy, A. Chawla, G. Das, and C. Yu. Group recommendation: Semantics and efficiency. *PVLDB*, 2(1):754–765, 2009.
- [4] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proc. RecSys*, pages 119–126. ACM, 2010.
- [5] S. Berkovsky and J. Freyne. Group-based recipe recommendations: Analysis of data aggregation strategies. In *Proc. RecSys*, pages 111–118. ACM, 2010.
- [6] S. Bourke, K. McCarthy, and B. Smyth. Using social ties in group recommendation. In *Proc. 22nd Irish Conference on Artificial Intelligence and Cognitive Science (AICS)*, 2011.
- [7] A. J. Chaney, M. Gartrell, J. M. Hofman, J. Guiver, N. Koenigstein, P. Kohli, and U. Paquet. A large-scale exploration of group viewing patterns. In *Proc. TVX*, pages 31–38. ACM, 2014.
- [8] Y. Chen, L. Cheng, and C. Chuang. A group recommendation system with consideration of interactions among group members. *Expert. Syst. Appl.*, 34(3):2082–2090, 2008.
- [9] I. A. Christensen and S. N. Schiaffino. Social influence in group recommender systems. *Online Information Review*, 38(4):524–542, 2014.
- [10] P. Cremonesi, P. Modica, R. Pagano, E. Rabosio, and L. Tanca. Personalized and context-aware TV program recommendations based on implicit feedback. In *Proc. EC-Web*, pages 57–68. Springer, 2015.
- [11] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales. Group recommending: A methodological approach based on bayesian networks. In *Proc. ICDE Workshops*, pages 835–844. IEEE, 2007.
- [12] T. De Pessemier, S. Dooms, and L. Martens. Comparison of group recommendation algorithms. *Multimedia Tools Appl.*, 72(3):2497–2541, 2014.
- [13] T. G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.
- [14] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. *SIAM J. Discrete Math.*, 17(1):134–160, 2003.
- [15] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *Proc. PODS*, pages 102–113. ACM, 2001.
- [16] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada. Enhancing group recommendation by incorporating social relationship interactions. In *Proc. GROUP*, pages 97–106. ACM, 2010.
- [17] S. Ghazarian and M. Nematbakhsh. Enhancing memory based collaborative filtering for group recommender systems. *Expert Syst. Appl.*, 42(7):3801–3812, 2015.
- [18] J. Gorla, N. Lathia, S. Robertson, and J. Wang. Probabilistic group recommendation via information matching. In *Proc. WWW*, pages 495–504. ACM, 2013.
- [19] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and W. Cao. Deep modeling of group preferences for group-based recommendation. In *Proc. AAAI*, pages 1861–1867. AAAI Press, 2014.
- [20] H. Kim and A. El-Saddik. A stochastic approach to group recommendations in social media systems. *Inf. Syst.*, 50:76–93, 2015.
- [21] N. Kim and J. Lee. Group recommendation system: Focusing on home group user in TV domain. In *Proc. SCIS*, pages 985–988. IEEE, 2014.
- [22] M. Kompan and M. Bieliková. Social structure and personality enhanced group recommendation. In *Proc. UMAP Workshops*. CEUR-WS.org, 2014.
- [23] X. Liu, Y. Tian, M. Ye, and W. Lee. Exploring personal impact for group recommendation. In *Proc. CIKM*, pages 674–683. ACM, 2012.
- [24] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Model. User-Adapt. Int.*, 14(1):37–85, 2004.
- [25] J. Masthoff and A. Gatt. In pursuit of satisfaction and the prevention of embarrassment: Affective state in group recommender systems. *User Model. User-Adapt. Int.*, 16(3-4):281–319, 2006.
- [26] E. Ntoutsi, K. Stefanidis, K. Nørnvåg, and H. Kriegel. Fast group recommendations by applying user clustering. In *Proc. ER*, pages 126–140. Springer, 2012.
- [27] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. PolyLens: A recommender system for groups of user. In *Proc. ECSCW*, pages 199–218. Kluwer, 2001.
- [28] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, and G. Jiménez-Díaz. Social factors in group recommender systems. *ACM TIST*, 4(1):8, 2013.
- [29] S. Seko, T. Yagi, M. Motegi, and S. Muto. Group recommendation using feature space representing behavioral tendency and power balance among members. In *Proc. RecSys*, pages 101–108. ACM, 2011.
- [30] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier. Analysis of strategies for building group profiles. In *Proc. UMAP*, pages 40–51. Springer, 2010.
- [31] C. Shin and W. Woo. Socially aware TV program recommender for multiple viewers. *IEEE Trans. Consumer Electronics*, 55(2):927–932, 2009.
- [32] R. Sotelo, Y. Blanco-Fernández, M. López-Nores, A. Gil-Solla, and J. J. Pazos-Arias. TV program recommendation for groups based on multidimensional tv-anytime classifications. *IEEE Trans. Consumer Electronics*, 55(1):248–256, 2009.
- [33] M. O. van Deventer, J. de Wit, J. Vanattenhoven, and M. Guelbahar. Group recommendation in an hybrid broadcast broadband television context. In *Proc. UMAP Workshops*. CEUR-WS.org, 2013.
- [34] E. Vildjiounaite, V. Kyllönen, T. Hannula, and P. Alahuhta. Unobtrusive dynamic modelling of TV programme preferences in a Finnish household. *Multimedia Syst.*, 15(3):143–157, 2009.
- [35] M. Ye, X. Liu, and W. Lee. Exploring social influence for recommendation: a generative model approach. In *Proc. SIGIR*, pages 671–680. ACM, 2012.
- [36] Z. Yu, X. Zhou, Y. Hao, and J. Gu. TV program recommendation for multiple viewers based on user profile merging. *User Model. User-Adapt. Int.*, 16(1):63–82, 2006.
- [37] Q. Yuan, G. Cong, and C. Lin. COM: A generative model for group recommendation. In *Proc. KDD*, pages 163–172. ACM, 2014.
- [38] H. Zhu, B. A. Huberman, and Y. Luon. To switch or not to switch: Understanding social influence in online choices. In *Proc. CHI*, pages 2257–2266. ACM, 2012.