Attention-based Group Recommendation

TRAN DANG QUANG VINH, Nanyang Technological University, Singapore
TUAN-ANH NGUYEN PHAM, Nanyang Technological University, Singapore
GAO CONG, Nanyang Technological University, Singapore
XIAO-LI LI, Institute for Infocomm Research(I2R), A∗STAR, Singapore

Group recommendation aims to recommend items for a group of users, e.g., recommending a restaurant for a group of colleagues. The group recommendation problem is challenging, in that a good model should understand the group decision making process appropriately: users are likely to follow decisions of only a few users, who are group's leaders or experts. To address this challenge, we propose using an attention mechanism to capture the impact of each user in a group. Specifically, our model learns the influence weight of each user in a group and recommends items to the group based on its members' weighted preferences. Moreover, our model can dynamically adjust the weight of each user across the groups; thus, the model provides a new and flexible method to model the complicated group decision making process, which differentiates us from other existing solutions. Through extensive experiments, it has demonstrated that our model significantly outperforms baseline methods for the group recommendation problem.

CCS Concepts: • **Recommender Systems**; • **Group Recommendation**; • **Collaborative Filtering**; • **Attention Mechanism**;

1 INTRODUCTION

The rapid growth of social networking services such as event-based social network services (Meetup), dining and restaurant services (Yelp), and media (Netflix and Last.fm) has made it increasingly easy for people to organize and participate in group activities. Those group activities include, for example, having dinners with colleagues, going to the cinema with spouses, and going picnic with friends. Owning tremendous data from people sharing social relationships and activities online, such social networking services face the challenge of targeting not only individuals but also user groups. As a result, recommendation for groups of users becomes an important task, which is beneficial for both users and social network services.

Traditional recommendation methods targeting individuals, however, cannot be applied in group recommendations. Many recent studies therefore have focused on developing recommender systems for group recommendations [2, 4, 8, 16, 23, 29, 30, 41, 43]. Group recommendation is challenging since the preference differences among users in a group require certain trade-offs in order to balance the different preferences and recommend the most favorable items to the group. Existing group recommendation methods can be categorized into memory-based and

Authors' addresses: Tran Dang Quang Vinh, Nanyang Technological University, Singapore, trandang001@e.ntu.edu.sg; Tuan-Anh Nguyen Pham, Nanyang Technological University, Singapore, pham0070@e.ntu.edu.sg; Gao Cong, Nanyang Technological University, Singapore, gaocong@ntu.edu.sg; Xiao-Li Li, Institute for Infocomm Research(I2R), A∗STAR, Singapore, xlli@i2r.a-star.edu.sg.

model-based approach [21, 36]. The memory-based approach can be further divided into two classes: preference aggregation [21, 25] and score aggregation [4, 15, 24, 27, 31]. The preference aggregation strategy combines all user preferences to create a group profile and then makes recommendations to the group, whereas the score aggregation strategy computes a list of recommendations for each member, and then combines individual lists to generate recommendations, using strategies such as average [4, 5], least misery [2], maximum satisfaction [7], etc. However, the two strategies cannot model the interaction among group members in term of aggregating members′ preferences. In contrast, the model-based approach models the decision making process of the group to recommend items. Nevertheless, existing models are insufficient to replicate the complicated decision making process of the group (more details in Section 2.1).

In this work, we propose a new approach for resolving the group recommendation problem. Our solution results from the key intuition that members in a group tend to follow the opinions of the most important members (leaders/experts) rather than consider opinions from all group members equally. This is because members in a group have different expertise levels on the group′s topic, and a group usually only has a few users who are experts in that topic. Based on this intuition, we aim to model the interaction among users in a group, which helps us to explore the roles of different group members and hence, make more accurate recommendations for the group.

To achieve our goal, we introduce the Attentive Group Recommendation (AGR) model with the use of attention mechanisms in collaborative filtering to address the two mentioned challenges in group recommendation. Employing attention mechanism, AGR is able to learn the influence of other group members′ expertise on one member′s decision in the same group, and therefore, better model the group decision making process. Applying attention mechanism not only helps our model to explore the impact of each user on the group but also capture the various impacts of one user across different groups, which none of existing methods is able to achieve. The key contributions in this paper are:

- The proposed Attentive Group Recommendation model takes a novel deep learning approach to tackle the group recommendation problem. AGR is the first to exploit the attention mechanism technique for group recommendation.
- AGR is able to dynamically adjust the weight for each user across groups, providing a clearer picture of the roles of a user in different groups and better explanation of the group′s final decision.
- We conduct extensive experiments on four datasets and show that AGR consistently achieves better results than state-of-the-art methods.

The rest of the paper is organized as follows: Section 2 overviews existing literature; Section 3 describes the preliminaries, including Attention Mechanism and Bayesian Personalized Ranking (BPR); Section 4 proposes the Attentive Group Recommendation model; Section 5 shows the comparison experiments on four datasets; and Section 6 concludes our work.

2 RELATED WORK

2.1 Group Recommendation

Group recommendation is a relevant problem in many industries such as social media [31, 34], tourism [26], and entertainment [15, 27, 42]. While recommendation techniques targeting individuals are extensively studied, research into group recommendation has been limited [6].

Group recommendation methods can be categorized into memory-based and model-based approach, where the memory-based approach can be further divided into the preference aggregation approach and the score aggregation approach [2]. The preference aggregation approach makes recommendations based on a group profile that combines all user preferences [25, 42], while the score aggregation approach computes a recommendation score of an item for each user, and then aggregates the scores across users to derive a group recommendation

score of the item [4, 15, 24, 27, 31]. The two most popular strategies for score aggregation are the average (AVG) and the least misery (LM) strategies. The AVG strategy takes the average score across individuals in the group as the final recommendation score, thereby maximizing overall group satisfaction [25, 42]. Alternatively, the LM strategy pleases everyone by choosing the lowest among all individuals' scores as the final score [4]. Both score aggregation methods have major drawbacks. The AVG strategy may return items that are favorable to some members but not to the others, while the LM strategy may end up recommending mediocre items that no one either loves or hates. Baltrunas *et al.* [4] point out that the performance of either strategy depends on group size and inner-group similarity. Yahia *et al.* [2] propose the concepts of relevance and disagreement. Arguing that preference disagreements on each item among group members are inevitable, the authors experimentally show that taking into account disagreement significantly improves the recommendation quality of AVG and LM strategies.

We next review the model-based approaches [1, 21, 36, 40] for group recommendation. For example, Seko *et al.* [35] propose a model that incorporates item categories into recommendation, arguing that item categories influence the group's decision and items of different categories are not strictly comparable. The method, however, only applies to pre-defined groups such as couples, which can be treated as pseudo-users and apply single user recommendation techniques, while real-life groups are often ad-hoc and formed just for one-off or few activities [23, 32]. Applying game theory in group recommendation, Carvalho *et al.* [8] consider each group event as a non-cooperative game, or a game with competition among members in the group, and suggest that the recommendation goal should be the game's Nash equilibrium. However, since a Nash equilibrium can be a set of items, the game theory approach may fail to recommend one specific item.

Probabilistic models have also been widely applied to solve group recommendation. Liu *et al.* [23] propose a personal impact topic (PIT) model for group recommendation, assuming that the most influential user should represent the group and have big impact on the group's decisions. However, such an assumption does not reflect the reality that a user's influence only contributes to the group's final decision if she is an expert in the field. For example, a movie expert may determine which movie the group should watch when she goes to the cinema with her friends, but she may not be the one to choose which restaurant they are dining afterwards. Yuan *et al.* [43] propose a consensus model (COM) for group recommendation. The model assumes (i) that a user's influence depends on the topic of decision, and (ii) that the group decision making process is subject to both the topic of the group's preferences and each user's personal preferences. Despite such assumptions, COM suffers from a drawback similar to that of PIT: COM assumes that a user has the same probability to follow the group's decisions across different groups. Alternatively, Gorla *et al.* [16] assume that the score of a candidate item depends not only on its relevance to each member in a group but also its relevance to the whole group. They develop an information-matching based model for group recommendation, but the model seriously suffers from high time complexity, $O(|U|^2|I|^2)$, where $|U|$ and $|I|$ are number of users and number of items, respectively. According to the experimental study [43], the method [16] cannot run to finish within days on datasets as we use in our experiments, and thus we do not compare with it in our experiments. Recently, Hu *et al.* [19] develop a deep-architecture model called DLGR that learns high-level comprehensive features of group preferences to avoid the vulnerability of the data. DLGR only focuses on pre-defined groups instead of ad-hoc groups. Therefore, we do not compare DLGR with our proposed model in this paper.

2.2 Deep Learning based Recommender Systems

Deep learning techniques have been extensively applied in recommender systems thanks to their high-quality recommendations [9–11, 13, 17, 18, 28, 37, 38]. Deep learning is able to capture non-linear and non-trivial relationships between users and items, which provides in-depth understanding of user demands and item characteristics, as well as the interactions between them. A tremendous part of literature has focused

on integrating deep learning into recommender systems to perform various recommendation tasks, where a comprehensive review can be found at [20, 44]. However, there is very little work that exploits deep learning techniques into group recommendation.

Our proposed AGR model makes recommendations using deep learning techniques based on user rating history. Specifically, AGR leverages attention mechanism to adapt the representation of the group, where more details will be shown in Section 3 and Section 4.

3 PRELIMINARIES

3.1 Problem Formulation

Let $U = \{u_1, u_2, \ldots, u_M\}$ and $I = \{i_1, i_2, \ldots, i_N\}$ be the sets of M users and N items, respectively. We denote a history log as $H = \{\langle g_1, s_1 \rangle, \langle g_2, s_2 \rangle, \ldots, \langle g_N, s_N \rangle\}$, where $g_n \subset U$ denotes an ad-hoc group and $s_n = i_k$ denotes the selected item by the group.

Given a target group g_t, we aim to generate a recommendation list of items that group members $u_{g_t, i}$ in the group g_t may be interested in. Note that the target group can be an ad-hoc group. The group recommendation problem can be defined as follows:

Input: Users U, items I, historical log H, and a target group g_t.

Output: A function that maps an item to a real value to represent the item score for the target group $f_{g_t} : I \to \mathbb{R}$.

3.2 Bayesian Personalized Ranking (BPR)

Based on the matrix factorization method, the Bayesian personalized ranking (BPR) method aims to address the challenge of implicit feedback recommendation [33]. Although the method was previously developed for the personalized ranking task, one can also leverage BPR for optimizing the group recommendation task. Specifically, we model a triplet of one group and two items: the positive item, which is observed, and the negative item, which is not. Our triplet model assumes that if group i has viewed an item j (positive item), the group must prefer this item over all other unobserved item k (negative item). The model thus ranks positive items higher than the negative ones.

Specifically, our objective function is formulated as follows:

$$\arg\min_{\Theta} \sum_{(i,j,k) \in \mathcal{D}_s} -\ln \sigma(\hat{R}_{ij}(\Theta) - \hat{R}_{ik}(\Theta)) + \lambda \|\Theta\|^2 \tag{1}$$

in which the set $\mathcal{D}_s$ contains all pairs of positive and negative items for each group; Θ represents the model parameters; $\hat{R}_{ij}(\Theta)$ is the predicted score for group i and item j; σ is the logistic sigmoid function; and λ is the regularization parameter.

Since groups' real-life data are usually implicit [23, 43], BPR is the suitable learning method for our group recommendation model.

3.3 Attention Mechanism

Attention mechanism is one of the most exciting among recent advancements in deep learning [3, 12, 39]. It has been successfully applied in various machine learning tasks such as machine translation [3], image and video captioning [39], and speech recognition [12]. The concept of attention is that when people visually access an object, we tend to focus on (pay attention to) certain important parts of the object instead of the whole object in order to come up with a response. Figure 1 represents an example of the attention mechanism. The attention model takes n arguments $x_1, x_2, \ldots, x_n$ and a context c. It then returns a vector y, which is the combination of x_i, given the information linked to the context c. Specifically, given a context c, the model returns a weighted arithmetic mean of each x_i, where the weight α_i is chosen according to the relevance of x_i.
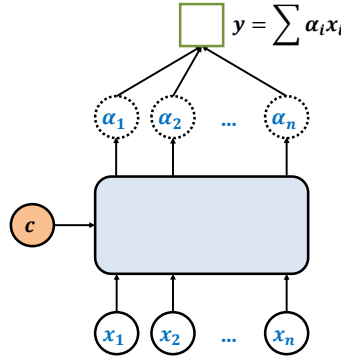
Fig. 1. Attention Mechanism

Our model adopts the attention mechanism with the main idea is to learn attentive weights of a user for other users in a group, in which higher weights indicate the corresponding users are more important; thus, their contributions are more important for the group's final decision. Aiming to expand the use of attention mechanism to group recommendation, the proposed AGR to the best of our knowledge is the first attention-based group recommendation model.

4 ATTENTIVE GROUP RECOMMENDATION

This section introduces our Attentive Group Recommendation (AGR) model. We first present the key intuition behind our model and then describe the general framework.

4.1 Model Intuitions

The AGR model aims to simulate group decision making based on the key intuition that users in a group usually come from diverse backgrounds with different expertise; therefore, when it comes to make the group's decision, each person's opinion has different influence (**Intuition 1**). Moreover, users usually appreciate opinions of users who are experts in the group's topic more than those of other users (**Intuition 2**). For example, in a group of travel planning for upcoming vacation, opinion from those who have traveled to Canada should have higher impact if the group is considering Canada as the destination for the trip. In addition, a user contributes differently in different groups depending on her relevant expertise (**Intuition 3**). For example, a user who has solid knowledge about movies may dominate the decision making of choosing a movie for a group to watch, but may not make any contribution to the decision making of choosing a camping site.

Overall, we observe that interactions between members are important to the group's decision making: users always discuss their opinions with one another before making a group decision. The importance of member interactions has not been examined in existing methods of group recommendation. A major contribution of the proposed AGR is that the model explores how a user affects other users in the group decision making and how the group decision changes accordingly.

Intuitively, we argue that during group decision making, (i) each user nominates certain users as the main decision makers of the group, and (ii) the most voted users then choose an item for the group. Such a voting scheme implies that the expert users in the fields relevant to the group's preferences usually receive high votes. With that in mind, for a group i of n users, we can consider the voting scheme (or the group decision making process) as n sub-processes that happen simultaneously. That is, the first user votes for other users, the second user votes for other users, ..., and the n-th user votes for other users. Therefore, we create n sub-groups from a
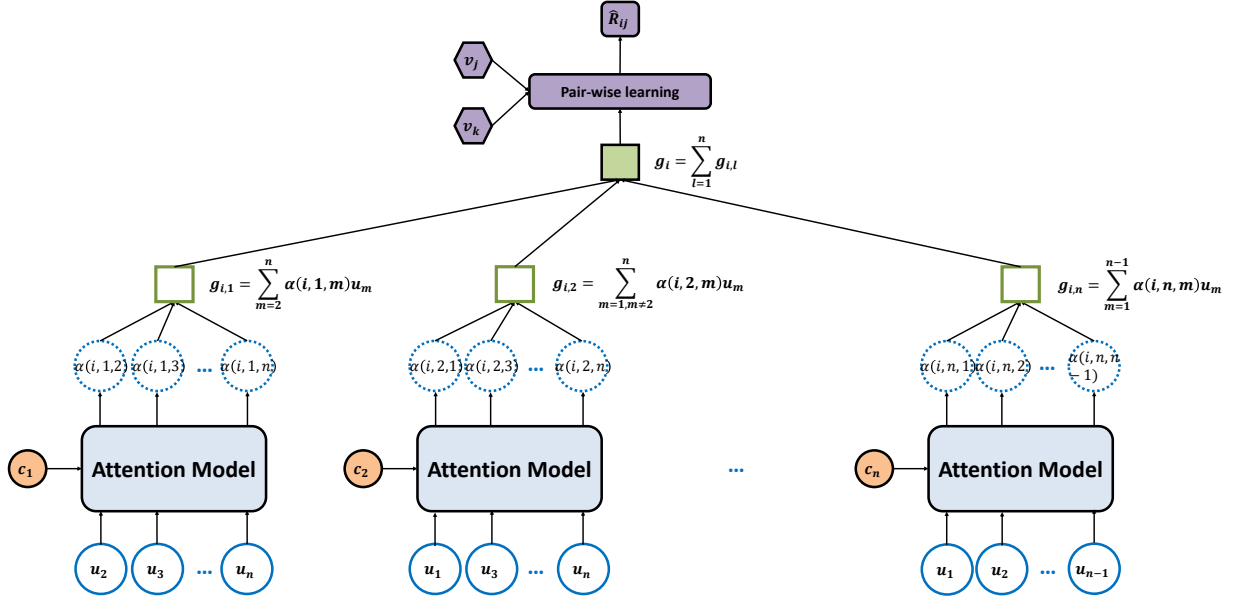
Fig. 2. Attentive Group Recommendation

group i of n users to explore the influence of each user as perceived by the remaining members (more details in Section 4.2).

4.2 General Framework

AGR models the group's preference score with respect to a candidate item using neural network. We next simulate the two steps of the group decision making process as mentioned.

To simulate step (i) of group decision making, we can consider an influence weight $\alpha(i, l, m)$, which represents the preference vote of user l with respect to user m in group i. It is expected that if user m is an important user in the group, she will receive high weight $\alpha(i, l, m)$ from user l, which supports our **Intuition 2**. One can estimate the influence weight $\alpha(*, l, m)$ for every (l, m)-pair of users assuming that $\alpha(*, l, m)$ is constant across all groups in which user l and user m participate. However, such an assumption contradicts to our **Intuition 3** because user l can perceive the influence of user m differently in different groups, which contradicts to our intuition. According to **Intuition 3**, user m should have larger influence in groups with topics relevant to her expertise than she does in groups with topics she is unfamiliar with. Thus, when it comes to voting, user l tends to nominate user m as the main decision maker in the group where user m is an expert, but not in other groups that user m is not familiar. The influence weight $\alpha(*, l, m)$ therefore should be calculated dynamically according to changing group preferences. Therefore, by using the attention mechanism to learn $\alpha(*, l, m)$ dynamically, we are able to detect the most important users in the group whose weights are higher than others; thus, the step (ii) of the group decision making process can be obtained easily.

We propose estimating influence weights for users in a group using an attention-based model. Particularly, given group members in a group i, when a user l look into other users in the group, she tends to only focuses on few expert users, and ignores other users, which is similar with the idea of attention mechanism introduced in Section 3.3. This motivates us to use an attention network to compute $\alpha(i, l, m)$ to measure how much important

user m is in group i under user l's opinion. Consequently, combining outputs from all sub-groups, we can estimate the importance weights of all users in group i (**Intuition 1**), and accordingly make the recommendation for the group.

Formally, AGR models each user l by two factor vectors: the user-latent vector $\mathbf{u}_l$ and the user-context vector $\mathbf{c}_l$. The attention model uses the context vector $\mathbf{c}_l$ to estimate the impact of the remaining users ($\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{l-1}, \mathbf{u}_{l+1}, \ldots, \mathbf{u}_n\}$) on user l. To be more specific, we first divide group $\mathbf{g}_i$ into n sub-groups, where each sub-group $\mathbf{g}_{i,l}$ include all users in the group $\mathbf{g}_i$ except user l. We then can apply an attention sub-network for each sub-group, whose output is the representation for that sub-group. This representation serves as the outcome of the sub-process voting under the view of user l. The representation (or the output) of each attention sub-networks is calculated as $\mathbf{g}_{i,l} = \sum_{m \neq l} \alpha(i, l, m) \mathbf{u}_m$. Therefore, the representation of a group i is $\mathbf{g}_i = \sum_l \mathbf{g}_{i,l}$ where $\mathbf{g}_{i,l} = \sum_{m \neq l} \alpha(i, l, m) \mathbf{u}_m$. We consider $\mathbf{g}_i = \sum_l \mathbf{g}_{i,l}$, that is, the voting scheme counts the votes of all users equally.

Figure 2 illustrates the architecture of AGR. For a given group $\mathbf{g}_i$, we create n attention sub-networks. Each attention sub-network l takes the user-context vector $\mathbf{c}_l$ (filled orange circle) and the set of member user-latent vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{l-1}, \mathbf{u}_{l+1}, \ldots, \mathbf{u}_n\}$ (solid blue circles), and then returns the attention weight $\alpha(i, l, m)$ of each user m ($m \neq l$) (dashed blue circles). Finally, the output of each attention sub-network l is calculated as the weighted sum $\mathbf{g}_{i,l} = \sum \alpha(i, l, m) \mathbf{u}_m$ (solid green square), which represents the group given the user-context $\mathbf{c}_l$. The final representation of the group $\mathbf{g}_i$ is then computed as the summation $\mathbf{g}_i = \sum_l \mathbf{g}_{i,l}$ (filled green square). Lastly, we employ BPR for optimizing the group recommendation task as mentioned in Section 3.2.

Attention Sub-networks. Each attention sub-network models the interactions between each member l and the rest of the group to learn the preference votes of user l for other members in the group. Given a user-context vector $\mathbf{c}_l$ and a set of user-latent vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\} \setminus \{\mathbf{u}_l\}$, we use a two-layer network to compute the attention score $a(i, l, m)$ as:

$$a(i, l, m) = \mathbf{w}^T \phi(\mathbf{W}_c \mathbf{c}_l + \mathbf{W}_u \mathbf{u}_{m, m \neq l} + \mathbf{b}) + \mathbf{c}, \quad l, m = \overline{1, n} \tag{2}$$

in which the matrices $\mathbf{W}_c$, $\mathbf{W}_u$ and bias $\mathbf{b}$ are the first layer parameters, and the vector $\mathbf{w}$ and bias $\mathbf{c}$ are the second layer parameters. We simply use a linear $\phi(x) = x$, but one can also use a ReLU function $\phi(x) = \max(0, x)$.

We normalize $a(i, l, m)$ using the Softmax function to obtain the final attention weights:

$$\alpha(i, l, m) = \frac{\exp(a(i, l, m))}{\sum_{m=1, m \neq l}^{n} \exp(a(i, l, m))} \tag{3}$$

Predicted Score. After we obtain the representation of the group i, the predicted score $\hat{R}_{ij}$ for group i and item j is computed as follows:

$$\begin{aligned} \hat{R}_{ij} &= \mathbf{g}_i^T \mathbf{v}_j \\ &= \left(\sum_l \mathbf{g}_{i,l} \right)^T \mathbf{v}_j \\ &= \left(\sum_l \sum_{m \neq l} \alpha(i, l, m) \mathbf{u}_m \right)^T \mathbf{v}_j, \end{aligned} \tag{4}$$

in which $\mathbf{v}_j$ is the item latent vector for item j.

Objective Function. AGR leverages BPR pair-wise learning objective to optimize the pair-wise ranking between the positive and negative items. The objective function Eq. (1) can be rewritten as follows:

$$\arg\min_{\Theta} \sum_{(i,j,k)\in\mathcal{D}_s} -\ln\sigma\left\{\left(\sum_l\sum_{m\neq l}\alpha(i,l,m)\mathbf{u}_m\right)^T\mathbf{v}_j \quad - \quad \left(\sum_l\sum_{m\neq l}\alpha(i,l,m)\mathbf{u}_m\right)^T\mathbf{v}_k\right\} \quad + \quad \lambda(\|\Theta\|^2), \quad (5)$$

in which Θ represents the model parameters; and $\alpha(i,l,m)$ is the weight of user l vote for user m in group i. We then use Adaptive Moment Estimation (Adam) to optimize our objective function.

5 EXPERIMENTS

In this section, we report experimental results of comparing AGR and six baseline techniques on four datasets. We also report the learned attention weights to evaluate the dominant users in group decision making. Such results will offer explanation for group recommendation result, which is an additional advantage of AGR. In general, our experiments aim to answer the following research questions (RQ):

- **RQ 1:** How does AGR perform as compared to existing state-of-the-art methods?
- **RQ 2:** Are the dynamic weights learned by AGR more preferable than the fixed weights learned by existing methods? How effective is our attention model?
- **RQ 3:** How does AGR perform with different group sizes?

5.1 Experimental Settings

5.1.1 Datasets. We conduct our experiments on four real-world datasets. The first dataset is from an event-based social network (EBSN), Plancast[1], which is used in [22]. Plancast allows users to directly follow the event calendars of other users. An event in Plancast consists of a user group and a venue. We therefore consider an event a group, and each user in the event a group member. Members in the group will select a venue (the candidate item) to host the event. Our goal is to recommend a venue for the group event.

Our second dataset is the crawled dataset from the EBSN Meetup[2], which is from the work [30]. We select the NYC data, which contains events held in New York City, as the dataset for our experiments. Similar to Plancast, we aim to recommend a venue for a given group to host an event.

We derive the last two datasets from the MovieLens 1M Data[3]. The MovieLens 1M Data contains one million movie ratings from over 6,000 users on approximately 4,000 movies. Following the approach in [4], we extract from the MovieLens 1M Data two datasets: MovieLens-Simi and MovieLens-Rand. Users in the MovieLens-Simi data are assigned into the same group when they have high inner group similarity, while users in the MovieLens-Rand data are grouped randomly. MovieLens-Simi and MovieLens-Rand groups thereby resemble two typical real life situations: groups can either include people with similar preferences, or form between unrelated people. For example, a group of close friends has high inner group similarity, whereas people on the same bus can be considered a random group.

Table 1 reports descriptive statistics of the four datasets. We randomly split each dataset into training, tuning and testing data with the ratio of 70%, 10% and 20% respectively.

5.1.2 Evaluation Metrics. Following the literature [4, 14, 16, 23, 30, 43], we evaluate model performance using three metrics: *precision* (*prec@K*), *recall* (*rec@K*), and normalized discounted cumulative gain (NDCG) (*ndcg@K*). Here K is the number of recommendations. We evaluate recommendation accuracy with $K = \{5, 10, 20\}$.
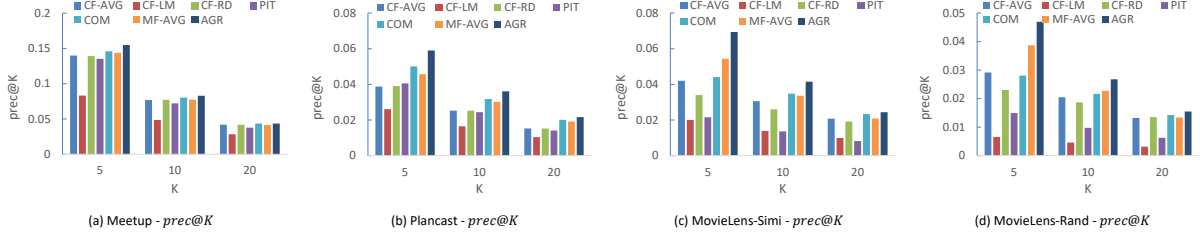
---

[1]https://www.plancast.com

[2]https://www.meetup.com/

[3]https://grouplens.org/datasets/movielens/

Fig. 3. Performance of Group Recommendation Methods in terms of *prec@K* ($p < 0.0001$)
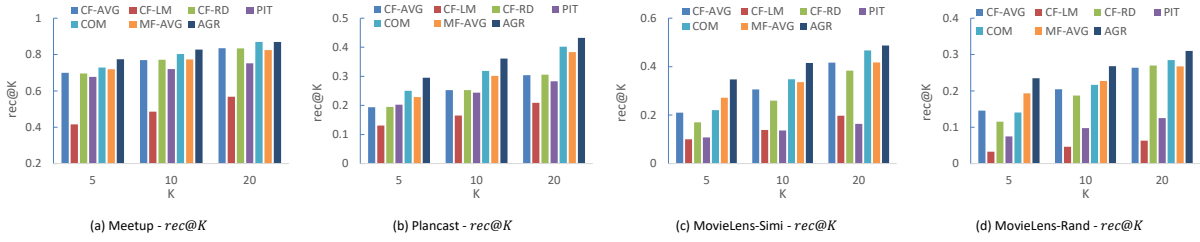

Fig. 4. Performance of Group Recommendation Methods in terms of *rec@K* ($p < 0.0001$)
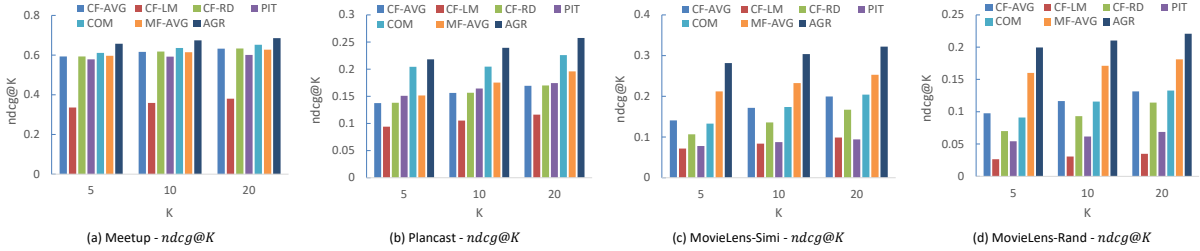

Fig. 5. Performance of Group Recommendation Methods in terms of *ndcg@K* ($p < 0.0001$)

precision@K is the fraction of top-*K* recommendations selected by the group, while *recall@K* is the fraction of relevant items (true items) that have been retrieved in the top *K* relevant items. We average the *precision@K* and *recall@K* values across all testing groups to calculate *prec@K* and *rec@K*. We also use the *NDCG* metric to evaluate the rankings of true items in the recommendation list. We average the *NDCG* values across all testing groups to obtain the *ndcg@K* metric.

For all of the three metrics, a larger metric value indicates better recommendations.

5.1.3 Recommendation Methods. This section experimentally compares six recommendation methods: CF-AVG, CF-LM, CF-RD [2], PIT [23], COM [43], MF-AVG and the proposed AGR. Among the baseline models,

Table 1. Dataset Statistics

Dataset	Plancast	Meetup	MovieLens -Simi	MovieLens -Rand
Total Users	41,065	42,747	5,759	5,802
Total Groups	25,447	13,390	29,975	54,969
Total Items	13,514	2,705	2,667	3,413
Avg. Group Size	12.01	16.66	5.00	5.00
Avg. Record for a User	7.44	5.22	26.03	47.37
Avg. Record for an Item	1.88	4.95	11.24	16.11

CF-AVG, CF-LM, CF-RD are score-aggregation approaches; PIT and COM are state-of-the-art probabilistic models; and MF-AVG is a matrix factorization model.

User-based CF with averaging strategy (CF-AVG): CF-AVG applies user-based CF to calculate a preference score for each user with respect to a candidate item i, and then averages the preference scores across all users to obtain the group recommendation score of item i.

User-based CF with least-misery strategy (CF-LM): Similar to CF-AVG, CF-LM first applies user-based CF to calculate a score for each user with respect to a candidate item i. However, the recommendation score of item i is taken as the item's lowest preference score across all users.

User-based CF with relevance and disagreement strategy (CF-RD) [2]: CF-RD also performs user-based CF to calculate a score for each user with respect to a candidate item i. The group recommendation score is calculated as the sum of the relevance and the disagreement scores of the item. The relevance score is obtained using CF-AVG or CF-LM, whereas the disagreement score is either the average pair-wise relevance difference for the item across group members (the average pair-wise disagreement method), or the mathematical variance of the relevance of the item across group members (the disagreement variance method).

Personal impact topic model (PIT) [23]: PIT is an author-topic model. Assuming that each user has an impact weight that represents the influence of the user to the final decision of the group, PIT chooses a user with a relatively large impact score as the group's representative. The selected user then chooses a topic based on her preference, and then the topic generates a recommended item for the group.

Consensus model (COM) [43]: COM relies on two assumptions: (i) the personal impacts are topic-dependent, and (ii) both the group's topic preferences and individuals' preferences influence the final group decision. The first assumption allows COM to derive topic-dependent personal impacts from group-topic and topic-user distributions. The second assumption allows COM to aggregate the group's topic preferences and individual's preferences subjective to weights derived from users' personal variables.

Average Matrix Factorization (MF-AVG): MF-AVG is a matrix factorization model. MF-AVG takes the average score of an item across all group members as the group recommendation score of the item. MF-AVG thereby considers all personal impact weights equal, assuming that all members contribute equally to the group's decision making. In other words, MF-AVG can be considered as our model with uniform weights setting. We represent a group as $g = \sum_i w_i u_i$ where $w_i = \frac{1}{n}$, we optimize the BPR pair-wise ranking objective function to predict group recommendation scores.

Attentive Group Recommendation (AGR): Our proposed model combines attention sub-networks to create an attentive model that learns the dynamic personal impact weights of each user.
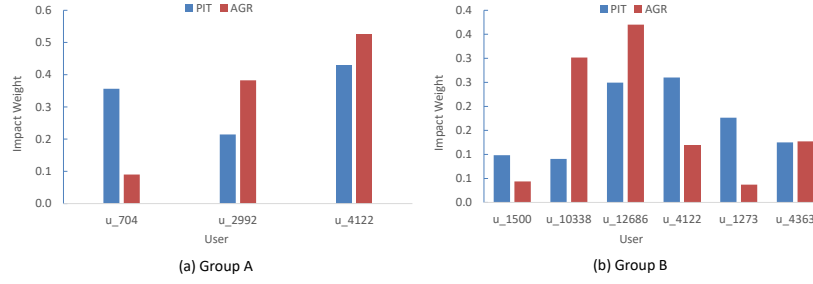
Fig. 6. Attention Weights Learned by PIT and AGR

Parameter Settings. For PIT and COM, we tuned the number of topics and kept other hyper-parameters as default. As regards to the MF-AVG and AGR models, we first randomly initialize the parameters using the Gaussian distribution with mean of 0 and standard deviation of 0.05, and then use Adaptive Moment Estimation (Adam) to optimize our objective functions. We also tested the batch size of $[128, 256, 512]$, the learning rate of $[0.001, 0.005, 0.01, 0.05, 0.1]$, and different regularizers of $[0.001, 0.01, 0.1, 0]$. We empirically set the embedding size of MF-AVG and AGR with the dimension of 50. We obtain the optimal setting with the batch size of 256, learning rate of 0.001, and regularizers of 0.01.

5.2 Overall Performance Comparison (RQ 1)

This section compares the recommendation results from AGR to those from the baseline models. Figure 3, Figure 4 and Figure 5 report the *prec@K*, *rec@K* and *ndcg@K* values for the four datasets with $K = \{5, 10, 20\}$. We observe from the three figures that:

- AGR consistently achieves the best performance across all methods, including score-aggregation approaches (CF-AVG, CF-LM, CF-RD) and probabilistic model approaches (PIT, COM).
- Although the group information of the two MovieLens datasets are conducted manually instead of already observed as Meetup and Plancast, our model can still be able to show the recommendation flexibility in adopting to randomness datasets.
- AGR and MF-AVG models produce good results in comparison to the previous state-of-the-art probabilistic models. MF-AVG performs better than PIT and COM on the Meetup and Plancast datasets, but not on the MovieLens-Simi and MovieLens-Rand dataset. One explanation is that the simplistic setup of MF-AVG cannot model the complexity of real life group interactions.

The *prec@5* metric values show that AGR outperforms the best baseline method by 6.22%, 17.91%, 27.71%, 21.36% on Meetup, Plancast, MovieLens-Simi and MovieLens-Rand, respectively. We observe the same improvements for *rec@5*. In general, AGR's recommendations are consistently better than the baseline methods', with the *p*-value less than 0.0001 for all the results.

It is noticeable that PIT's performance is not compare with those of other baseline models. One possible reason is that as a Meetup or Plancast group usually has a large number of participants, many of whom only join a few groups and thus have very limited historical data, the user impacts learned by PIT for such participants are not reliable. Another possible reason for PIT's poor performance is that the assumptions underlying PIT do not hold in the context of MovieLens data: since MovieLens users select movies independently from one another, there is no representative user in a MovieLens group.

We also observe that the AGR does not outperform the baselines models on the Meetup dataset as significantly as it does on the other three datasets. One explanation is that since a Meetup group often has few venue options,

Table 2. Performance Comparison between MF-AVG and AGR

Model	Meetup		Plancast		MovieLens -Simi		MovieLens -Rand	
	REC	NDCG	REC	NDCG	REC	NDCG	REC	NDCG
MF-AVG	0.851962	0.632753	0.467176	0.211513	0.321079	0.191142	0.530135	0.274007
AGR	0.888513	0.689149	0.502620	0.270515	0.360627	0.230099	0.578485	0.338959
p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001

group members tend to choose the place they are most familiar with, making it relatively easy to recommend the venue to the group [30]. While Meetup users form a big group before hosting an event and choosing the venue, such services as Plancast allow users to follow other users' event calendars and choose to participate in existing events. Plancast groups therefore tend to be more diverse than Meetup groups, and Plancast event venues are not as easily predicted as Meetup event venues.

In general, AGR achieves remarkable recommendation results on the variety of datasets. Our experiments show the flexibility of AGR in making group recommendation given different data types.

5.3 The Role of Attention Mechanism (RQ 2)

We further conduct paired t-tests on the performances of AGR and MF-AVG models to verify that AGR's improvements over MF-AVG are statistically significant at the five percent significance level. While AGR employs attention mechanism to calculate the weights, MF-AVG assigns a normalized constant weight to each group member. We conduct paired t-tests on the performance metrics of top-K recommended lists ($K \in [10, 100]$) for the two models to see whether AGR statistically significantly outperforms MF-AVG.

Table 2 compares the performances of AGR and MF-AVG. We observe that the mean pooling strategy of MF-AVG always performs worse than the attention mechanism of AGR, except in the MovieLens-Rand case. The good performance of MF-AVG on MovieLens-Rand data is expected because MovieLens-Rand groups satisfy MF-AVG assumptions: randomly-grouped users in MovieLens-Rand data tend to equally contribute to the groups' final decisions. The reported p-values are nominal for all tests, indicating that AGR's performances are statistically much stronger than MF-AVG's performances.

Attention Weight Visualization. As noted previously, an advantage of AGR is that the method allows us to calculate the attention weight values for further analyses. Figure 6 visualizes the attention weights learned by AGR and PIT for two randomly-chosen groups from our experiments. We look at the weights learned by PIT because similar to AGR, PIT learns the personal impact weight for each user [23]. Since COM does not learn the personal impact weight for each user [43], we do not compare AGR with COM in this Section.

Figure 6(a) and Figure 6(b) show a few user attention weights learned by PIT and AGR of two randomly-chosen groups that share a user no. 4122 ("u_4122"). Figure 6(a) reports the personal impact weights of three users in the first group (group A). According to both models, the user no. 4122 has the highest weights in group A and therefore dominates group A's decision making. Figure 6(b), however, shows that PIT continues to assume that user no. 4122 is the most influential user in group B, whereas AGR is able to detect other users who play more important roles in group B's decision making than user no. 4122 does. While PIT's personal impact parameter cannot differentiate the roles of one user in different groups and thus may fail to recognize influential members in a group, the attention mechanism of AGR can capture the dynamic user impacts in group decision making.
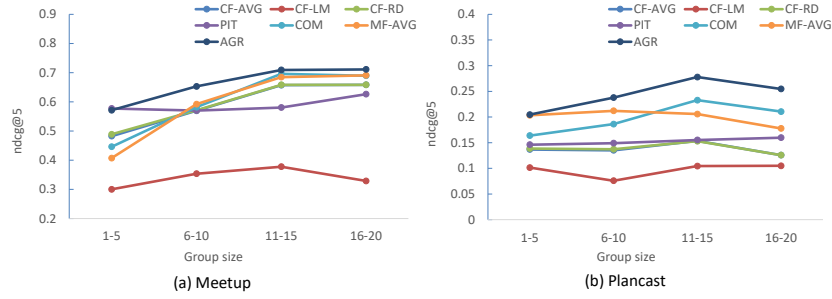
Fig. 7. Performance on Different Group Sizes

5.4 Model Performances for Different Levels of Group Sizes (RQ 3)

To study the performance of each recommendation method on different group sizes, we run the experiments for four levels of group size (1-5 members, 6-10 members, 11-15 members, and 16-20 members) using Meetup and Plancast groups. We keep the same setting as illustrated in Section 5.1 for all the models, and classify the groups into bins based on group size. Since the number of groups with more than 20 members is small, we decide to exclude these groups in this experiment. Figure 7 plots the resulting *rec@5* and *ndcg@5* curves. Note that since the group size of the MovieLens-Simi and MovieLens-Rand dataset is fixed, we do not study the different levels of group sizes on these two datasets.

Figure 7 shows that AGR achieves better performances than other baseline methods across different group sizes. We have the following observations: 1) AGR shows clear improvements in recommendations for groups of larger sizes. AGR improves 1.90% and 2.93% on the Meetup dataset over the best baseline for groups of 11-15 members and 16-20 members respectively, while these numbers are 19.30% and 20.97% for the Plancast dataset. This indicates the significance of AGR in addressing groups of larger sizes. 2) CF approaches often deliver good performance when the group size is small, as low diversity within a group facilitates smooth aggregations. As the number of members in the group increases, we need relatively complex methods such as probabilistic models or neural networks models to make adequate recommendations.

6 CONCLUSIONS

Aiming to solve the group recommendation problem through a deep learning approach, the Attentive Group Recommendation model (AGR) utilizes an attention mechanism to learn the influence weight of each user in a group in order to make group recommendations. The major contributions of AGR are that the model not only assumes and dynamically learns different impact weights of each given user for different groups, but it also also considers the interactions between users in the group. Thus, AGR is able to better model the group decision making process. To the best of our knowledge, AGR is the first model to exploit attention mechanism in group recommendation. Conducting extensive experiments on four real-world datasets, we show the consistent performance of AGR as compared to those of existing group recommendation methods.

A promising development direction for AGR is to incorporate side information such as social connections, text information (for example event description) or time information into the context to learn the attention model.

REFERENCES

[1] Deepak Agarwal and Bee-Chung Chen. 2010. fLDA: matrix factorization through latent dirichlet allocation. In *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*. 91–100.

[2] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawla, Gautam Das, and Cong Yu. 2009. Group Recommendation: Semantics and Efficiency. *PVLDB* 2, 1 (2009), 754–765.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* abs/1409.0473 (2014). http://arxiv.org/abs/1409.0473

[4] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. 2010. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*. 119–126.

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

[6] Ludovico Boratto. 2016. Group Recommender Systems. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. 427–428.

[7] Ludovico Boratto and Salvatore Carta. 2011. *State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups*.

[8] Lucas Augusto Montalvão Costa Carvalho and Hendrik Teixeira Macedo. 2013. Users' satisfaction in recommendation systems for groups: an approach based on noncooperative games. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*. 951–958.

[9] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. 335–344.

[10] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. In *SIGIR*.

[11] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. *CoRR* abs/1606.07792 (2016). http://arxiv.org/abs/1606.07792

[12] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua Bengio. 2015. Attention-Based Models for Speech Recognition. *CoRR* abs/1506.07503 (2015). http://arxiv.org/abs/1506.07503

[13] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*. 191–198.

[14] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*.

[15] Andrew Crossen, Jay Budzik, and Kristian J. Hammond. 2002. Flytrap: intelligent group music recommendation. In *Proceedings of the 7th International Conference on Intelligent User Interfaces, IUI 2002, San Francisco, California, USA, January 13-16, 2002*. 184–185.

[16] Jagadeesh Gorla, Neal Lathia, Stephen Robertson, and Jun Wang. 2013. Probabilistic group recommendation via information matching. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*. 495–504.

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*. 173–182.

[18] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based Recommendations with Recurrent Neural Networks. *CoRR* abs/1511.06939 (2015).

[19] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Wei Cao. 2014. Deep Modeling of Group Preferences for Group-Based Recommendation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*. 1861–1867.

[20] Alexandros Karatzoglou and Balázs Hidasi. 2017. Deep Learning for Recommender Systems. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*.

[21] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.

[22] Xingjie Liu, Qi He, Yuanyuan Tian, Wang-Chien Lee, John McPherson, and Jiawei Han. 2012. Event-based social networks: linking the online and offline social worlds. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*. 1032–1040.

[23] Xingjie Liu, Yuan Tian, Mao Ye, and Wang-Chien Lee. 2012. Exploring personal impact for group recommendation. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*. 674–683.

[24] Joseph F. McCarthy. 2002. Pocket Restaurant Finder: A situated recommender systems for groups. In *Proceeding of Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*.

[25] Joseph F. McCarthy and Theodore D. Anagnost. 1998. MusicFX: An Arbiter of Group Preferences for Computer Aupported Collaborative Workouts. In *CSCW '98, Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work, Seattle, WA, USA, November 14-18, 1998.* 363–372.

[26] Kevin McCarthy, Maria Salamó, Lorcan Coyle, Lorraine McGinty, Barry Smyth, and Paddy Nixon. 2006. CATS: A Synchronous Approach to Collaborative Group Recommendation. In *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11-13, 2006.* 86–91.

[27] Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. 2001. PolyLens: A recommender system for groups of user. In *Proceedings of the Seventh European Conference on Computer Supported Cooperative Work, 16-20 September 2001, Bonn, Germany.* 199–218.

[28] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based News Recommendation for Millions of Users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017.* 1933–1942.

[29] Tuan-Anh Nguyen Pham, Xutao Li, and Gao Cong. 2017. A General Model for Out-of-town Region Recommendation. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17).* 401–410.

[30] Tuan-Anh Nguyen Pham, Xutao Li, Gao Cong, and Zhenjie Zhang. 2016. A General Recommendation Model for Heterogeneous Networks. *IEEE Trans. on Knowl. and Data Eng.* 28, 12 (Dec. 2016), 3140–3153.

[31] Sebastiano Pizzutilo, Berardina De Carolis, Giovanni Cozzolongo, and Francesco Ambruoso. 2005. Group Modeling in a Public Space: Methods, Techniques, Experiences. In *Proceedings of the 5th WSEAS International Conference on Applied Informatics and Communications (AIC'05).* 175–180.

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

[33] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009.* 452–461.

[34] Amirali Salehi-Abari and Craig Boutilier. 2015. Preference-oriented Social Networks: Group Recommendation and Inference. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15).*

[35] Shunichi Seko, Takashi Yagi, Manabu Motegi, and Shin-yo Muto. 2011. Group recommendation using feature space representing behavioral tendency and power balance among members. In *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011.* 101–108.

[36] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Adv. Artificial Intelligence* 2009 (2009), 421425:1–421425:19.

[37] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2017. Translational Recommender Networks. *CoRR* abs/1707.05176 (2017). http://arxiv.org/abs/1707.05176

[38] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. *CoRR* abs/1801.09251 (2018). http://arxiv.org/abs/1801.09251

[39] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and Tell: A Neural Image Caption Generator. *CoRR* abs/1411.4555 (2014). http://arxiv.org/abs/1411.4555

[40] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011.* 448–456.

[41] Mao Ye, Xingjie Liu, and Wang-Chien Lee. 2012. Exploring social influence for recommendation: a generative model approach. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012.* 671–680.

[42] Zhiwen Yu, Xingshe Zhou, Yanbin Hao, and Jianhua Gu. 2006. TV Program Recommendation for Multiple Viewers Based on user Profile Merging. *User Model. User-Adapt. Interact.* 16, 1 (2006), 63–82.

[43] Quan Yuan, Gao Cong, and Chin-Yew Lin. 2014. COM: a generative model for group recommendation. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014.* 163–172.

[44] Shuai Zhang, Lina Yao, and Aixin Sun. 2017. Deep Learning based Recommender System: A Survey and New Perspectives. *CoRR* abs/1707.07435 (2017). http://arxiv.org/abs/1707.07435