Interact and Decide: Medley of Sub-Attention Networks for Effective Group Recommendation

Lucas Vinh Tran
Nanyang Technological University
trandang001@e.ntu.edu.sg

Tuan-Anh Nguyen Pham
Nanyang Technological University
pham0070@e.ntu.edu.sg

Yi Tay
Nanyang Technological University
ytay017@e.ntu.edu.sg

Yiding Liu
Nanyang Technological University
liuy0130@e.ntu.edu.sg

Gao Cong
Nanyang Technological University
gaocong@e.ntu.edu.sg

Xiaoli Li
Institute for Infocomm Research,
A*STAR
xlli@i2r.a-star.edu.sg

ABSTRACT

This paper proposes *Medley of Sub-Attention Networks* (MoSAN), a new novel neural architecture for the *group* recommendation task. Group-level recommendation is known to be a challenging task, in which intricate group dynamics have to be considered. As such, this is to be contrasted with the standard recommendation problem where recommendations are personalized with respect to a single user. Our proposed approach hinges upon the key intuition that the decision making process (in groups) is generally dynamic, i.e., a user's decision is highly dependent on the other group members. All in all, our key motivation manifests in a form of an attentive neural model that captures fine-grained interactions between group members. In our MoSAN model, each sub-attention module is representative of a single member, which models a user's preference with respect to all other group members. Subsequently, a Medley of Sub-Attention modules is then used to collectively make the group's final decision. Overall, our proposed model is both expressive and effective. Via a series of extensive experiments, we show that MoSAN not only achieves state-of-the-art performance but also improves standard baselines by a considerable margin.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Recommender Systems, Group Recommendation, Collaborative Filtering, Neural Attention Mechanism

1 INTRODUCTION

People often participate in activities in groups, e.g., having dinners with colleagues, watching movies with partners, and shopping with friends. This calls for effective techniques for group recommendation. Unfortunately, existing recommendation algorithms designed for individuals are not effective for group recommendation. The availability of group event data further promotes the research interest on how to make effective recommendations for a group of users [2, 4, 8, 15, 18, 20–22, 35, 37] which facilitates groups making decisions, and helps social network services improve user engagement. This paper is concerned with designing highly effective recommender systems that are targeted at modeling group preferences as opposed to individual preferences, where groups are ad-hoc (any combination of individuals) rather than pre-defined.

Group preferences are not straightforward to model, given the inherent complexity of group dynamics. To this end, this work aims to exploit the interactions between group members in order to drive the model towards highly effective group-level recommendations. Moreover, it is natural that collective decisions have a tendency to be dynamic, i.e., a user's preference may be highly influenced by the other members in the group. Group-level agreement tends to require a *consensus* amongst group members, in which this consensus largely depends on each member's roles and expertise.[1] As such, it is crucial to model the *interactions* among group members. However, existing proposals for group recommendation fail to model the interactions of group members well. Most of existing solutions belong to memory based methods that are based on either preference aggregation [18] or score aggregation strategy [4, 22] and do not consider the interactions of group members. These strategies overlook the interactions between group members, and use trivial methods to aggregate members' preferences. Some existing solutions are model-based approaches and try to exploit user interactions for group recommendation. However, they cannot fully utilize the user interactions as to be discussed in Section 2.

In our work, to model the interactions among group members, we propose a new neural architecture for group recommendation. Specifically, our architecture is a new variant of the attention mechanism in which each group member is represented with a single

---

[1]While this is not explicitly captured with any semantic information or meta-data, we hypothesize that this can be implicitly captured with simply interaction data.

sub-attention network. Subsequently, a group of users is then represented as a *'medley'* of sub-attention networks that is responsible for making the overall recommendation. The role of each sub-attention network is to capture the preference of its representative group member, *with respect to* all other members in the group. As such, our proposed model leverages user-user interactions for making group recommendation decisions, and is not only well-aligned with the fundamental intuition of group-level dynamics but also expressive in the sense that it considers the user-user interactions. In fact, our experiments demonstrate that a simple attentive aggregation of user representations is insufficient and has roughly identical performance to that of an average pooling matrix factorization (MF) baseline (More details will be discussed in Section 4). On the other hand, our experiments show that our model is significantly better than seven state-of-the-art baselines. All in all, our core intuition serves as an inductive bias for our model, providing more effective group recommender performance on multiple benchmark datasets.

Our Contributions. Overall, the key contributions of this work are summarized as follows:

- We propose MoSAN (*Medley of Sub-Attention Networks*), a novel deep learning architecture for the group recommendation problem. Our model distinguishes itself from all prior work in group recommendation based on the fact that it considers user-user interactions using sub-attention networks. To the best of our knowledge, this is the first neural model that explores the usage of user-user interactions for the group recommendation task.
- We conduct extensive experiments on four publicly available benchmark datasets. Our experimental results demonstrate that MoSAN achieves state-of-the-art performance, outperforming a myriad of strong competitors in the task at hand.
- In addition to comparison against well-studied baselines, we conduct ablation studies against two baselines AVG-MF (Average Matrix Factorization) and ATT-AVG (Attentive Aggregation) and observe that our approach significantly outperforms both approaches. This shows that our proposed model provides a more useful inductive bias for the task at hand.
- We show that the attention weights of MoSAN are interpretable, i.e., it is able to discover the different weights of each user across groups, highlighting the impact of each user in different groups.

2 RELATED WORK

Group Recommendation Systems. Group recommendation methods can be characteristically dichotomized into memory-based and model-based approaches, where the memory-based approach can be further divided into the preference aggregation and the score aggregation [2]. The preference aggregation makes recommendations based on a group profile that combines all user preferences [36], while the score aggregation computes a score of an item for each user, and then aggregates the scores across users to derive a group recommendation score of the item [4, 22]. 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 [36]. 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. [4] pointed out that the performance of either strategy depends on group size and inner-group similarity. [2] proposed 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.

Model-based approaches [1, 18, 29, 33] for group recommendation are also notable. [27] proposed 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 [20, 25]. Applying game theory in group recommendation, [8] considered each group event as a non-cooperative game, or a game with competition among members in the group, and suggested 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 applied to solve group recommendation. [20] proposed 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 he/she is an expert in the field. [37] proposed 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. Additionally, [15] assumed 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 suffers from high time complexity, taking several days to run on several datasets as reported by [37]. Owing to its computational prohibitivity, we do not compare with the method in our experiments.

Recently, [17] proposed a deep-architecture model called DLGR that learns high-level comprehensive features of group preferences to avoid the vulnerability of the data. Similar to the work [27], DLGR only focuses on pre-defined groups instead of ad-hoc groups, and thus cannot be applied to our setting. Therefore, we do not compare DLGR with our proposed model in this paper.

Neural Recommender Systems. Neural networks have been extensively applied in recommender systems thanks to their high-quality recommendations [9, 10, 13, 16, 23, 30]. Particularly, 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 [38]. However, there is very little work that exploits neural techniques into group recommendation.

Neural Attention. Attention mechanism is one of the most exciting recent advancements in deep learning [3, 12, 14, 31, 32]. The usage of neural attention in recommender systems has also gained considerable interest, with many works that exploit this recent advance for standard recommendation tasks [11, 30, 34]. Notably, a recent work exploits neural attention for a group recommendation setup called AGREE [7]. However, AGREE is different from our model as AGREE also focuses on pre-defined groups where it requires additional group preference representation information as one component to learn the final representation of a group. In addition, if we separate the group preference representation from AGREE, the architecture becomes completely different from the original design and it similarly becomes one of our baselines called Attentive Aggregation (ATT-AVG) model, which will be discussed in Section 4. Hence, their framework cannot be applied to our setting. Moreover, our overall intuition and framework (usage of sub-attention and user-user interactions) significantly distinguish our work from theirs.

3 OUR PROPOSED FRAMEWORK

Generally speaking, our proposed MoSAN model consists of two levels: 1) user interaction learning through sub-attention network modules in which each group member is represented with a single sub-attention network module to simulate the decision making process of other members under the influential of that group member; and 2) the medley of sub-group decisions which concludes the final decision to recommend items for the group. Next, we first present the input encoding of the group recommendation problem in Section 3.1. We then introduce the two key components of our proposed model in Section 3.2. Lastly, we discuss the optimization method in Section 3.3.

3.1 Input Encoding

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 (i.e., training instances) as $H = \{\langle g_1, s_1 \rangle, \langle g_2, s_2 \rangle, \ldots, \langle g_L, s_L \rangle\}$, where $g_l \subset U$ denotes an ad-hoc group and $s_l = i_k$ 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 $f_{g_t} : I \rightarrow \mathbb{R}$.
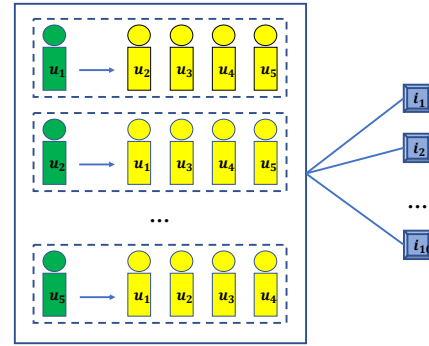

Figure 1: Illustration of group decision making process of MoSAN, in which green users represent *user-context* and yellow users represent *user-latent*.

For each training instance, our model accepts a list of group members (users) and an item. Each user and item are represented as one-hot vectors, which map onto a dense low-dimensional vector by looking up an user and item embedding matrix, respectively. These user and item embeddings are referred to as *user-latent* and *item-latent* vectors (denoted as $\mathbf{u}_m$ and $\mathbf{v}_j$, respectively) in the rest of the paper. Moreover, we introduce an additional user embedding matrix, referred to as *user-context* embedding (denoted as $\mathbf{c}_l$) which is specifically designed to denote the owner of each sub-attention network. Embedding matrices are trainable parameters of the model's architecture network and are trained end-to-end with the rest of the parameters. Finally, our model trains by pairwise ranking, which essentially requires negative item samples. We define the negative items as the items that are not selected by all members in the group.

Figure 1 illustrates the group decision making process we simulate in this paper. The solid rectangle represents the final decision of the group with each item. The inner dashed rectangles represent user interactions within the group, in which each dashed rectangle illustrates how a user (*user-context*) influences other users (*user-latent*) in the decision making process.

3.2 Medley of Sub-Attention Networks

This subsection introduces our proposed neural framework for group recommendation. To recapitulate, the key motivation behind our neural architecture is to enable group-level recommendations by modeling interactions between group members. Figure 2 illustrates the architecture of MoSAN.

Motivation. It is not straightforward to model the group preferences, given the inherent complexity of group dynamics. Therefore, typical aggregation algorithms may be insufficient for the task. Different aggregation strategies have been proposed such as average [4, 5], least misery [2], or maximum satisfaction [6]. In general, these aggregation strategies are also known as pre-defined strategies, where they first predict the scores across individuals for candidate items, and then aggregate those predicted scores of each member in a group via the strategies to obtain the group's preferences.
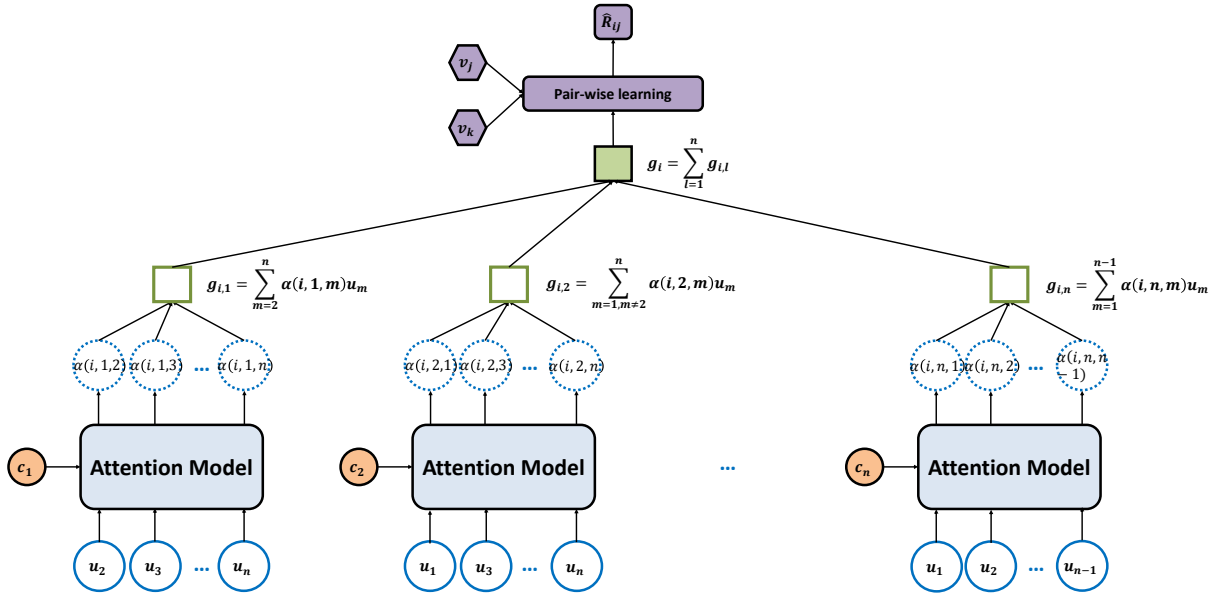
Figure 2: High level overview of our proposed Medley of Sub-Attention Networks (MoSAN) model. Each sub-attention networks is representative of a single group member, interacting with all other group members in order to learn its preference score.

We argue that these existing aggregation strategies are not sufficient to model the complexity and dynamics of the group due to its inflexibility in adjusting the weights of members in the group. For example, user A may have higher impact weight than user B in a given group when the group makes decision on which movie to watch, but have lower weight than user B when the group makes decision on which restaurant to dine at. Recently, neural attention mechanism has been proposed as one of the most exciting advancements in deep learning [3, 12, 32]. 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. Our model adopts the attention mechanism to learn attentive and dynamic weight of each user, in which higher weights indicate the corresponding users are more important; thus, their contributions are more important for the group's final decision.

Our Method. We focus on designing novel and effective neural architectures for group recommendation under representation learning framework. Specifically, we propose novel representation learning technique for learning-to-rank group-of-users and item pairs. Under the representation learning paradigm, we are able to model group representation via an end-to-end representation learning.

Let $\mathbf{u}_m$ and $\mathbf{v}_j$ be the embedding vector for user m and item j, respectively. We aim to obtain an embedding vector $\mathbf{g}_i$ for each group to estimate the group's preference on the item j. Formally, it can be defined as:

$$\mathbf{g}_i = f_a(\{\mathbf{u}_m\}_{m \in I_i}, \mathbf{v}_j) \qquad (1)$$

in which $\mathbf{g}_i$ denotes the representation learning of group i which represents its preference on item j; I_i contains the user indexes

of group i; and f_a is the aggregation function to be specified. In MoSAN, our model architecture presenting group embedding consists of two levels: 1) Sub-Attention Network Module, and 2) the Medley of Sub-Attention Networks.

We next elaborate the two-level structure of MoSAN.

Sub-Attention Network Module. For a given group $\mathbf{g}_i$, we create n attention sub-networks. Each attention sub-network l takes the user-context[2] 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) as input, and then returns the attention weight $\alpha(i, l, m)$ of each user m ($m \neq l$) (dashed blue circles) of sub-network l in group $\mathbf{g}_i$.

Intuitively, 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. Recall that this satisfies our key intuition and desiderata set out in the exposition of this paper. 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:

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

in which the matrices $\mathbf{W}_c$ and $\mathbf{W}_u$ are weight matrices of the attention network that convert user-context embedding and user-latent embedding to hidden layer respectively, and $\mathbf{b}$ is the bias vector of the hidden layer; the weight vector $\mathbf{w}$ and bias d are the parameters of the second layer that we use to project the hidden layer to the score $a(i, l, m)$. We simply use a linear $\phi(x) = x$ as an activation function, but one can also use other functions like a ReLU function $\phi(x) = \max(0, x)$.

---

[2]Note that the user-context vector is mainly used to differentiate the ownership of the current attention sub-network.

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

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

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 $\mathbf{c}_l$. As such, $\mathbf{g}_{i,l}$ can be treated as the l-th representation of the group i, which captures the decisions of each member given that these group members' decisions are influenced by the decision of member l.

Medley of Sub-Attention Networks. 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). This final representation can be interpreted as the feature representation of the group of users, which can be easily matched with the item embedding to determine the recommendation score. More concretely, 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:

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

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

One may argue that such a simple summation could fail to consider the different number of group members, e.g., the summation of $\mathbf{g}_i$ over a 3-member group tends to be much smaller than a 10-member group. As a result, according to Eq. (4), the group preference score on item v of the 3-member group will be much less than that of the 10-member group due to the huge number of members, even if the 10-member group may not like this item v as much as the 3-member group. However, since a large group (i.e., 10-member group) has large representation, it produces larger loss value in terms of absolute value. Hence, its gradient tend to be larger, and their group members representations are updated more during the training. If we choose appropriate learning rate, the performance will not much be affected. Moreover, in the testing phrase, we only consider and compare items to a group, so we do not have to concern about the large/small group problem. Usually the number of members in a group is small (i.e., less than 20 members), so this concern can be negligible. The normalization can also be taken into account at this layer; however, it will also be considered as an average pooling of the sub-attention networks, which makes the distinguish observation between each sub-attention network becomes less important.

Alternatively, one could consider an additional attentive aggregation at the second layer. However, our preliminary experiments showed this yielded no improvements in performance. This is intuitive, as our model already 'reasons' over group members using the sub-attention network modules at earlier layers. Specifically, regarding the summation, the difference of each member is already captured by the sub-attention network. This is because each sub-attention network already captures the relationship of one member against the group, which aligns well with this intuition. As such, an additional attention layer did not provide any benefit to the overall network structure.

3.3 Optimization and Learning

Objective function. Our proposed MoSAN leverages BPR [26] pair-wise learning objective to optimize the pair-wise ranking between the positive and negative items. The objective function can be rewritten as follows:

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

in which Θ represents the model parameters; and $\alpha(i, l, m)$ is the weight of user l votes for user m in group i. Subsequently, the model can be trained end-to-end with an optimizer such as Adaptive Moment Estimation (Adam).

Learning details. We next describe some details for learning our proposed model which are useful to replicate.

Mini-batch training. We perform mini-batch training. Each mini-batch contains interactions of group members and the item adopted by the group. Specifically, we shuffle all the observed interactions, and then sample a mini-batch of those observed ones. Lastly, we form the training instances by sampling a fixed number of negative instances for each observed interaction.

Dropout. We also employ dropout [28] to improve our proposed model's performance. Specifically, we drop with the dropout rate of ρ on the first layer of our neural sub-attention network. We empirically found that applying dropout on the hidden layer of the neural attention network did boost our generalization performance. On a side note, we only apply dropout during the training phrase and disable it during the testing phase.

4 EXPERIMENTS

In this section, we report experimental results of comparing MoSAN and seven state-of-the-art 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 MoSAN. In general, our experiments aim to answer the following research questions (RQ):

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

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/User	7.44	5.22	26.03	47.37
Avg. Record/Item	1.88	4.95	11.24	16.11

Table 1: Dataset Statistics

4.1 Experimental Settings

Datasets. We conduct extensive experiments on four real-world datasets. The first dataset is from an event-based social network (EBSN), Plancast,[3] which is used in [19]. 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 dataset crawled from the EBSN Meetup,[4] which is from the work [24]. 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. The statistics of this dataset is different from which reported in [20] due to the difference in the period of crawling.

The final two datasets are obtained from the MovieLens 1M Data.[5] 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. MovieLens-Simi contains groups with high user-user similarity. We select top 33% in terms of similarity of all possible pairs to form groups from there. MovieLens-Rand contains groups that are formed without the restrictions above. For a given group in both cases, if every member gives 4 stars or above to a movie, we assume that the movie is adopted by the group. 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. Note that the groups in our setting are ad-hoc, i.e., it is possible that a group appears only in test data, but not in training data.

Evaluation Metrics. Following previous work [4, 20, 24, 37], we evaluate model performance using three widely used evaluation metrics: *precision* (*prec@K*), *recall* (*rec@K*), and normalized discounted cumulative gain (NDCG) (*ndcg@K*). Here K is the number

---
[3]https://www.plancast.com
[4]https://www.meetup.com/
[5]https://grouplens.org/datasets/movielens/

of recommendations. We evaluate recommendation accuracy with $K = \{5, 10, 20\}$. *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.

Compared Baselines. We compare with seven state-of-the-art baselines in our experiments: CF-AVG, CF-LM, CF-RD [2], PIT [20], COM [37], MF-AVG and ATT-AVG.

- **User-based CF (CF-AVG, CF-LM, CF-RD)** [2]: The baselines are standard user-based collaborative filtering methods by integrating pre-defined aggregation strategies which are averaging strategy (CF-AVG), least-misery strategy (CF-LM) and relevance and disagreement strategy (CF-RD).
- **Personal impact topic model (PIT)** [20]: 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)** [37]: 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.
- **Average Matrix Factorization (MF-AVG)**: This baseline is a simplified version of MoSAN and considers the average embedding of all users in the group. All users are weighted equally. We represent a group as $g = \sum_i w_i u_i$ where $w_i = \frac{1}{n}$, and we optimize the BPR objective to predict group recommendation scores.
- **Attentive Aggregation (ATT-AVG)**: This baseline is also a simplified version of MoSAN, and represents a group embedding using a vanilla attentive aggregation over all the user embeddings in the group. The user embeddings are optimized with the BPR objective function.

Hyperparameter Settings. For PIT and COM, we tuned the number of topics and kept other hyperparameters as default. With regard to the MF-AVG/ATT-AVG and MoSAN 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/ATT-AVG and MoSAN 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. We randomly set dropout rate $\rho = 0.5$ and sample 3 negative items for each training instance.

4.2 Overall Performance Comparison (RQ 1)

This subsection compares the recommendation results from MoSAN to those from the baseline models. Figure 3, Figure 4 and Figure 5
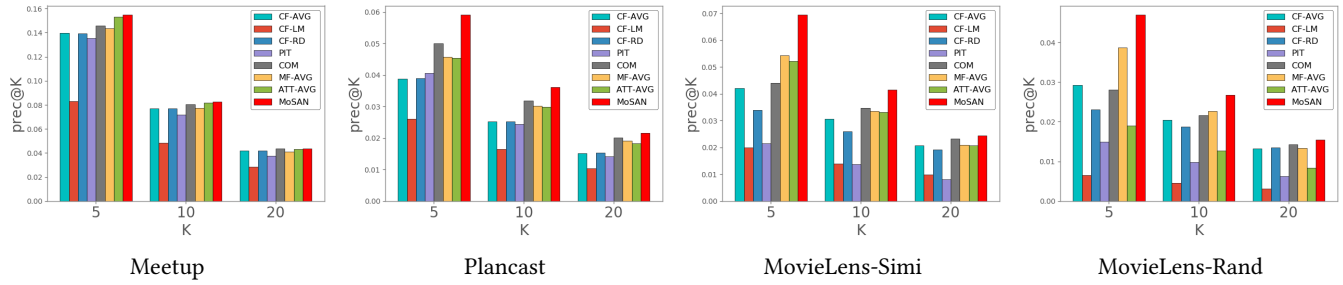
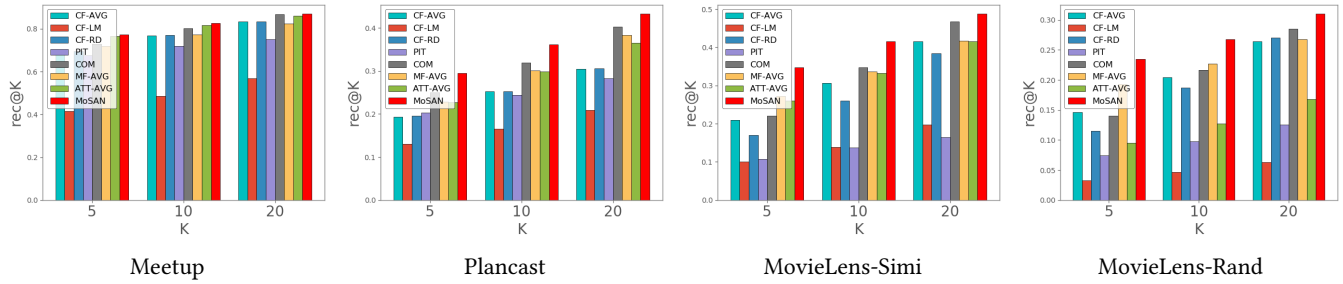**Figure 3: Performance of Group Recommendation Methods in terms of prec@K** ($p < 0.0001$) (*Best viewed in color*).

Figure 4: Performance of Group Recommendation Methods in terms of rec@K ($p < 0.0001$) (*Best viewed in color*).
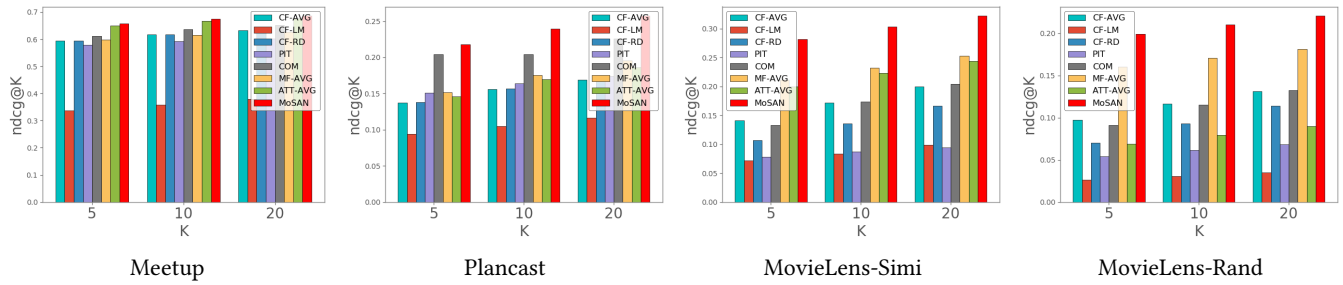

Figure 5: Performance of Group Recommendation Methods in terms of ndcg@K ($p < 0.0001$) (*Best viewed in color*).

Model	Meetup		Plancast		MovieLens -Simi		MovieLens -Rand	
	rec@5	ndcg@5	rec@5	ndcg@5	rec@5	ndcg@5	rec@5	ndcg@5
MF-AVG	0.851962	0.632753	0.467176	0.211513	0.321079	0.191142	0.530135	0.274007
ATT-AVG	0.883432	0.681998	0.439959	0.200346	0.221269	0.099581	0.520669	0.263710
MoSAN	**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

Table 2: Performance comparison between MF-AVG and ATT-AVG (Ablation study) on four datasets. Results show that our proposed attention mechanism is significantly better than a standard attention-based aggregation.

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:

- MoSAN 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 is generated 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.

- MoSAN 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 MovieLens-Simi and MovieLens-Rand datasets in terms of *ndcg@K*, but not on the Meetup and Plancast datasets. One explanation is that the simplistic setup of MF-AVG cannot model the complexity of real life group interactions.
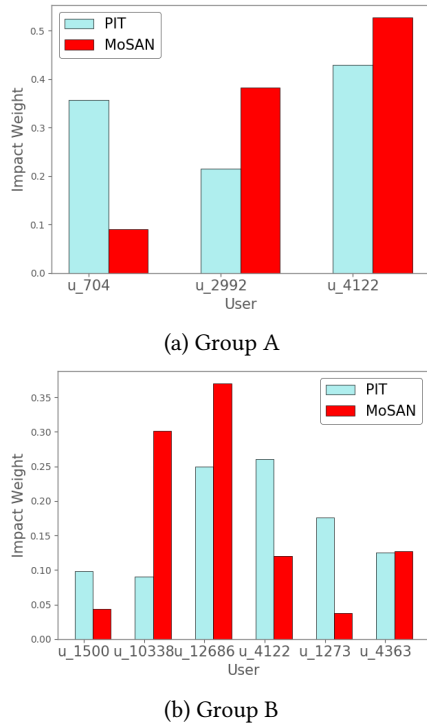
(a) Group A

(b) Group B

Figure 6: Attention Weights Learned by PIT and MoSAN.

Number of users K to be removed	Meetup		Plancast	
	MoSAN	PIT	MoSAN	PIT
$K = 1$	0.643932	0.56832	0.197189	0.14193
$K = 3$	0.593263	0.56690	0.118760	0.14085
$K = 5$	0.500849	0.56424	0.081018	0.14013
$K = 7$	0.377586	0.56238	0.058164	0.13954
$K = 9$	0.284848	0.56163	0.046473	0.13950

Table 3: Performance comparison between MoSAN and PIT on Meetup and Plancast datasets in terms of ndcg@5 by removing top-K high weight users.

more diverse than Meetup groups, and Plancast event venues are not as easily predicted as Meetup event venues.

In addition, one may concern the ability of MoSAN to deal with large groups. However, it should not be an issue as we observe that typically most of groups have fewer than 10 users. For larger groups, we can always reduce the number of users to a smaller number (e.g., 20 users). Then, we can use our model to firstly compute the attention weights of users and remove the users with very small weights. We can always perform this step because in reality, the number of group leaders/experts who participates in making the final decision for the group is always very small.

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

4.3 The Role of Attention Mechanism (RQ 2)

Ablation Study. We further conduct paired t-tests on the performances of MoSAN against MF-AVG and ATT-AVG models to verify that MoSAN's improvements over MF-AVG and ATT-AVG are statistically significant at the five percent significance level. While MoSAN employs a user-user attention mechanism to calculate the weights, MF-AVG assigns a normalized constant weight to each group member. Similarly, ATT-AVG is a simple attentive pooling over all users. Even though it also adopts an attention mechanism, it does not consider user-user interactions. We conduct paired t-tests on the performance metrics of top-K recommended lists ($K \in [10, 100]$) for MoSAN, MF-AVG and ATT-AVG to see whether MoSAN statistically significantly outperforms MF-AVG and ATT-AVG. We also observe that ATT-AVG does not always outperform MF-AVG, signifying that the standard attention mechanism is insufficient.

Table 2 compares the performances of MoSAN and MF-AVG. We observe that the mean pooling strategy of MF-AVG always performs worse than the attention mechanism of MoSAN. 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 MoSAN's performances are statistically much stronger than MF-AVG's performances.

Attention Weight Visualization. As noted previously, an advantage of MoSAN is that the method allows us to calculate the attention weight values for explanation of group recommendation results. Figure 6 visualizes the attention weights learned by MoSAN and PIT for two randomly-chosen groups from our experiments. We compare with the weights learned by PIT because similar to

We observe that there is no obvious winner among the baseline solutions. For each dataset, we define the baseline that has the greatest performance among the proposed ones as *the best baseline* method. The *prec@5* metric values show that MoSAN outperforms the best baseline method significantly by 0.94%, 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, MoSAN's recommendations are consistently better than the baseline methods', with the p-value less than 0.0001 for all the results and thus statistically significant.

It is noticeable that PIT's performance is not comparable 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. Hence, 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 MoSAN 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, and group members tend to choose the place they are most familiar with, making it relatively easy to recommend the venue to the group [24]. While Meetup users form a big group before hosting an event and choosing the venue, Plancast allow users to follow other users' event calendars and choose to participate in existing events. Plancast groups therefore tend to be

MoSAN, PIT is able to learn the personal impact weight for each user [20]. Since COM does not learn the personal impact weight for each user [37], we do not compare MoSAN with COM here.

Figure 6(a) and Figure 6(b) show user attention weights learned by PIT and MoSAN 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 MoSAN 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 MoSAN can capture the dynamic user impacts across groups in group decision making.

The importance of high weight users. In addition, we also conduct another ablation experiment to analyze the importance of users by comparing the recommendation results of MoSAN and PIT with and without high weight users on Meetup and Plancast datasets. We rank users by their impact weight in decreasing order and remove top-K users from group recommendation, and report the change in MoSAN's and PIT's performance, in which $K \in \{1, 3, 5, 7, 9\}$.

Table 3 shows the performance comparison between MoSAN and PIT in terms of *ndcg@5* by removing top-K high weight users. It is noticeable that the performance of PIT is stable as the number of removed users increases, whereas the performance of MoSAN drops dramatically. Specifically, the average performance drop of MoSAN for removing every additional two users is 18.15% for Meetup dataset and 29.97% for Plancast dataset, while the performance of PIT is nearly unchanged. The decreasing performance of MoSAN shows that the removed users are important users in the group while the top-ranked users in PIT are not really important users; and thus, the attention mechanism in MoSAN can really capture the dynamic user impacts better than PIT. It reflects the effectiveness of MoSAN in analyzing and weighting high/low impact users in different groups, which is critical for group recommendation systems. All in all, the ablation experiment of the importance of users shows the preciseness of MoSAN over PIT in learning the attentive weight for each user.

4.4 Model Performances for Different 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 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 very small, we 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.
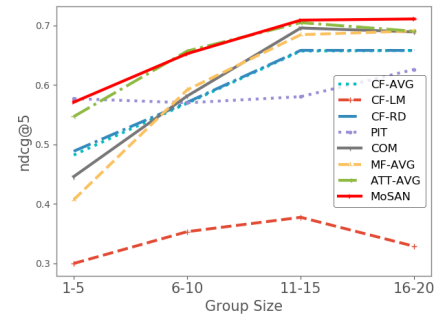
(a) Meetup
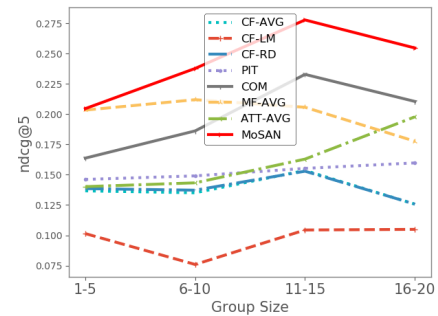
(b) Plancast

Figure 7: Performance on Different Group Sizes (*Best viewed in color***).**

Figure 7 shows that MoSAN achieves better performance than other baseline methods across different group sizes. We have the following observations: 1) MoSAN shows clear improvements in recommendations for groups of larger sizes. MoSAN improves 0.59% and 2.93% on the Meetup dataset over the best baseline method 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 MoSAN 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.

5 CONCLUSION

In this paper, we propose a new effective neural recommender for group recommendation. The major contributions of MoSAN are that the model not only dynamically learns different impact weights of each given user for different groups, but it also considers the interactions between users in the group. Thus, MoSAN is able to better model the group decision making process. In addition, MoSAN can tell the relative importance of users in a group, and thus make explainable recommendation possible. We conduct extensive experiments on four real-world datasets, demonstrating that MoSAN is capable of outperforming a myriad of strong state-of-the-art baselines.

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. https://doi.org/10.1145/1718487.1718499

[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. https://doi.org/10.14778/1687627.1687713

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 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. https://doi.org/10.1145/1864708.1864733

[5] Shlomo Berkovsky and Jill Freyne. 2010. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*. 111–118. https://doi.org/10.1145/1864708.1864732

[6] Ludovico Boratto and Salvatore Carta. 2011. State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups. In *Information Retrieval and Mining in Distributed Environments*. 1–20. https://doi.org/10.1007/978-3-642-16089-9_1

[7] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. 2018. Attentive Group Recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 645–654. https://doi.org/10.1145/3209978.3209998

[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. https://doi.org/10.1145/2487788.2488090

[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. https://doi.org/10.1145/3077136.3080797

[10] 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. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016, Boston, MA, USA, September 15, 2016*. 7–10. https://doi.org/10.1145/2988450.2988454

[11] Jin Yao Chin, Kaiqi Zhao, Shafiq R. Joty, and Gao Cong. 2018. ANR: Aspect-based Neural Recommender. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. 147–156.

[12] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-Based Models for Speech Recognition. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 577–585.

[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] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805

[15] 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.

[16] 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.

[17] 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.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37. https://doi.org/10.1109/MC.2009.263

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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. https://doi.org/10.1145/3097983.3098108

[24] 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. https://doi.org/10.1109/TKDE.2016.2601091

[25] 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)*.

[26] 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.

[27] 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. https://doi.org/10.1145/2043932.2043953

[28] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[29] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Adv. Artificial Intelligence* 2009 (2009), 421425:1–421425:19. https://doi.org/10.1155/2009/421425

[30] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. 2309–2318. https://doi.org/10.1145/3219819.3220086

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 6000–6010.

[32] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 3156–3164.

[33] 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. https://doi.org/10.1145/2020408.2020480

[34] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. AAAI Press, 3119–3125.

[35] 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.

[36] 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.

[37] 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. https://doi.org/10.1145/2623330.2623616

[38] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1 (2019), 5:1–5:38. https://doi.org/10.1145/3285029