GAME: Learning Graphical and Attentive Multi-view Embeddings for Occasional Group Recommendation

Zhixiang He
Department of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
zhixiang.he@my.cityu.edu.hk

Chi-Yin Chow*
Department of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
chiychow@cityu.edu.hk

Jia-Dong Zhang*
Department of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
jzhang26@cityu.edu.hk

ABSTRACT

Group recommendation aims to suggest preferred items to a group of users rather than to an individual user. Most existing methods on group recommendation directly learn the *inherent interests of groups and users* or *inherent features of items*, i.e., independently modeling the inherent embeddings of groups, users or items. However, the independent view severely suffers from the cold-start problem when making recommendations for occasional groups that are temporally formed by a set of users and have few interactions on items. Actually, the groups, users and items are interdependent because they interact with one another. The interdependencies constitute an interaction graph that provides multiple views to model the embeddings of groups, users and items from their interacting counterparts to improve recommendation for occasional groups. To this end, we propose a model, named GAME to learn the *Graphical and Attentive Multi-view Embeddings* (i.e., representations) for the groups, users and items from the independent view and counterpart views based on the interaction graph. In the counterpart views, the embedding of a group, user or item is aggregated from the interacting counterparts based on an attention mechanism that derives the adaptive weight for each counterpart. For instance, a user's embedding may be aggregated from her interacting items or groups. Further, GAME applies neural collaborative filtering to investigate the interactions between the multi-view embeddings of groups (or users) and items for group recommendation. Finally, we conduct extensive experiments on two real datasets. The experimental results show that GAME outperforms other state-of-the-art models, especially on both cold-start groups (i.e., occasional groups) and cold-start items.

CCS CONCEPTS

• **Information systems → Collaborative filtering**.

KEYWORDS

Occasional group; group recommendation; interaction graph; multi-view embeddings; attention mechanism

*The corresponding authors: C.-Y. Chow and J.-D. Zhang.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401064

ACM Reference Format:
Zhixiang He, Chi-Yin Chow, and Jia-Dong Zhang. 2020. GAME: Learning Graphical and Attentive Multi-view Embeddings for Occasional Group Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401064

1 INTRODUCTION

Recent years, group activities are becoming more and more popular with the development of social networks, which has led to the rapid development of group recommender systems. Group recommendation returns the preferred items for a group of users (i.e., members) and is different from traditional recommendation that suggests the interesting items for individual users [11, 16, 44, 45, 52]. Group recommender systems have been applied in various areas such as e-commerce [28, 34], entertainment [6, 12, 27], social media [15, 43], tourism [20, 40], and visualization [41]. In general, there are two types of groups: persistent groups and occasional groups. Persistent groups often have stable members with similar preferences and rich interactions on items whereas occasional groups are formed temporally and have very sparse group-item interactions. Usually, an occasional group has only one interaction on a certain item. This paper focuses on studying recommendation for occasional groups.

The approaches on individual recommendation can be directly applied for group recommendation by treating a group as a virtual user and ignoring the distinct preferences of the members in the group [17, 25, 32]. These approaches only work well for persistent groups with stable members and rich interactions [5, 22, 33], but perform poorly for occasional groups with highly dynamic memberships and little interaction information [36, 47]. Thus, existing works are committed to aggregating the members' preferences as the group preference for making recommendations. A simple way is to aggregate the scores of all members on the same item based on a strategy such as average [2, 3, 49], least misery [1], and maximum satisfaction [4], which predefines the weight of each member and achieves suboptimal performance. A better way is to learn the members' embeddings and aggregate them with dynamic influence weights as the group embedding [5, 18, 36]. Unfortunately, these works separately model the embeddings of users or items from their inherent interests or features by the independent view. That is, they do not exploit the interdependencies among groups, users and items to capture their embeddings and thus they severely suffer from the cold-start problem of occasional groups.

However, in the group activities, groups, users and items highly interact with one another, where users often participate in a group

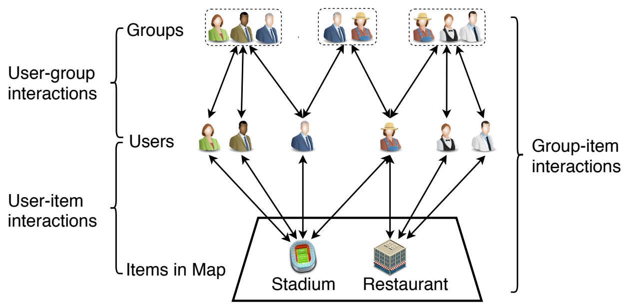


Figure 1: An example on the interaction graph.

for the same preference and the group of users visits some preferred items together. For example, Alice, Bob, and Cathy formed a group and went for the movie "Ne Zha" because they all like cartoon animations. Actually, the groups, users and items are interdependent and constitute an interaction graph, in which we consider *groups, users and items as nodes* and *the interactions among them as edges*, as depicted in Figure 1. In terms of the interaction graph, it is possible to model the embeddings of groups, users and items from multiple views, i.e., from their interaction counterparts. For example, the embedding of a user may be aggregated from her interacting counterparts including items and groups; an item or group can also be modeled based on the similar way. In this paper, we propose a model, called GAME, to learn *Graphical and Attentive Multi-view Embeddings* for occasional group recommendation. **GAME contains two major parts**: (1) **Representation learning.** GAME models the multi-view embeddings for users, items and groups as their representations from the independent view and counterpart views based on the interaction graph. (2) **Neural interaction learning.** GAME deploys neural collaborative filtering (NCF) [17] to learn the interactions between the learned representations of groups (or users) and items in order to predict the preference scores of a group on any items for group recommendation. Specifically, in the first part, GAME learns the multi-view embedding of a user by incorporating the preference information from three views, including her *inherent interests* (i.e., the independent view), *interacting items* and *participating groups* (i.e., two counterpart views). Similarly, GAME captures the multi-view embedding for an item from three views, i.e., its inherent features, interacting users and groups. Besides, GAME derives a group's embedding over its members' multi-view embeddings only, because occasional groups are formed temporally and have sparse interactions on items and it is hard to directly catch the representation of a group from the sparse group-item interaction data. In all the counterpart views, the embedding of a user, item or group is aggregated from the interacting counterparts based on an attention mechanism that adaptively infers the dynamic weight for each counterpart, because a counterpart has the distinct influence on the user, item or group. In particular, the item-specific influence weights of users (i.e., the members in a group) are devised for aggregating the members' representations as the group representation.

In general, we summarize our main contributions as follows:

- We propose GAME for occasional group recommendation to model the embeddings of users, items and groups from multiple views by fully exploiting the interaction graph to capture their comprehensive representations.
- In the counterpart views, we learn the embeddings of users, items and groups from their all interaction counterparts and derive the dynamic weight for each counterpart based on the attention mechanism to improve the performance of group recommendation.
- We conduct extensive experiments on two real datasets. Experimental results show the superiority of GAME by comparing it with other state-of-the-art models, especially on both cold-start (or occasional) groups and cold-start items.

The rest of this paper is organized as follows. Section 2 highlights the related work. Section 3 discusses some preliminaries used in this work. Section 4 describes the proposed GAME for group recommendation. Section 5 presents the experimental results and analysis. Finally, the conclusion is given in Section 6.

2 RELATED WORK

This section highlights related work in *individual recommendation based on embedding learning* and *group recommendation*.

2.1 Individual recommendation based on embedding learning

For individual recommendation [17, 24], each user or item is often represented by an independent embedding (i.e., inherent embedding) that reflects the user's inherent interests or item's inherent features. In the studies [7, 29], a user is also represented as an embedding based on her interacting items (i.e., item-view user embedding) that considers the implicit influences of the interacting items on the user's preference. Symmetrically, an item can be represented as an embedding based on its interacting users (i.e., user-view item embedding) to take into account the item's implicit features indicated by the user preferences [7, 10]. However, most existing methods [7, 10, 17, 24, 29] only model the single-view embedding for users or items.

2.2 Group recommendation

Existing works on group recommendation aim to learn the preference of a group by aggregating its members' preferences. These works can be generally divided into three categories: *score aggregation approaches*, *explicit profile aggregation approaches*, and *implicit embedding aggregation approaches*.

2.2.1 Score aggregation approaches. The approaches of this category generate the scores of all members in a group on an item by individual recommendation models and aggregate the individual scores to obtain the preference score of the group on the item [1–4]. The popular aggregation strategies include average [2, 3], least misery [1], and maximum satisfaction [4]. The average strategy maximizes the overall satisfactions of a group by taking the mean score of all members; the least misery strategy aims to satisfy the least preference of a member in the group by taking the smallest score of all members; and the maximum satisfaction strategy is committed to reaching the maximal satisfaction of a member in the group by taking the largest score of all members. The work [3] compares the effectiveness of different predefined aggregation strategies and indicates that there is no salient winners or losers. In other words, it is not possible to achieve good performance via a predefined strategy for all groups, because the predefined strategy cannot adapt to different groups and easily falls into local optimal solutions. Further, the study [31] investigates the influence weights of all members on a group at different temporal contexts and applies the weighted sum over the members' scores for TV programs; this method improves the predefined strategies but requires additional temporal context information.

2.2.2 Explicit profile aggregation approaches. Instead of simply aggregating the scores outputted by individual recommendation models, the approaches of this category fuse the profiles of members into a group profile and input the group profile into individual recommendation models to generate recommendation results [27, 49]. The profile of a member is usually represented as an explicit feature vector on items interacted by the member. For example, the work [49] uses the actors, genres and keywords of TV programs watched by a member to construct a feature vector as the profile of the member; then it combines the profiles of members based on the total distance minimization to ensure that the group profile is close to the profiles of most members. Particularly, MusicFX [27] considers the rating vector of a user on musical genres (i.e., items) as her profile and utilizes a simple linear combination and sum of square to aggregate the profiles of users. These profiles consist of explicit features that are not learnable and directly extracted from data. Nonetheless, the explicit features are often limited or unavailable in the data. As a result, these approaches may not effectively represent the profiles of members or groups.

2.2.3 Implicit embedding aggregation approaches. To address the limitation of explicit profile aggregation approaches, the approaches of this category represent each member as an implicit embedding that is learned from the interaction data between groups, users and items. For instance, some probabilistic generative models [26, 46, 50] learn the embedding of each member as one probability distribution over hidden topics and the influence weights of all members as the other probability distribution; then they calculate the group embedding by summing the embeddings of all members with the influence weights. These probabilistic generative models require sufficient group-item interaction data and are not applicable for occasional groups. More sophisticatedly, various deep learning techniques have been developed to learn the implicit embeddings of members for group recommendation [5, 18, 22, 36, 47]. Specifically, AGREE [5] captures the inherent embeddings of members from their inherent interests. MoSAN [36] catches the embedding of each member in a group based on the inherent embeddings of the rest members in the group. Other approaches [18, 22, 47] derive the group-specific embedding of a member from her inherent embedding with the influence of the group on the member based on restricted Boltzmann machine [21], bipartite graph embeddings [48] or multilayer perceptron (MLP) [37]. Further, these works aggregate the embeddings of all members via the simple average or least misery [22], or adaptive attention mechanisms [5, 18, 36, 47]. The attention mechanisms learn the dynamic influence weight of a member in different groups and have been widely applied in other research areas, such as computer vision [8, 39], machine translation [35, 38] and traffic prediction [19]. Nevertheless, the work [47] requires social relationships to estimate the influence weights and is not appropriate for making recommendations to occasional groups without the social relationships among members. Moreover, other works [5, 18, 22, 36] simply model the embeddings of users, items and groups from only one or two views; thus, they fail to take full advantage of the interactions among users, items and groups, and severely suffer from the cold-start problem of occasional groups. To effectively overcome the sparse issue of group-item interaction data, this paper proposes GAME for learning the representations (i.e., multi-view embeddings) of users, items and groups from multiple views based on the interaction graph.

Table 1: Descriptions of key symbols

Symbols	Descriptions
$\mathcal{G}$	The interaction graph
U, V, G	The sets of users, items, and groups
u, v, g	User u, item v, and group g
$\mathtt{M}, \mathtt{N}, \mathtt{L}$	Numbers of users, items, and groups
$\mathcal{G}_{UG}, \mathcal{G}_{GV}, \mathcal{G}_{UV}$	User-group, group-item, and user-item graphs
E_{UG}, E_{GV}, E_{UV}	Edges in $\mathcal{G}$
$R_V(u)$	The set of u's interacting items in $\mathcal{G}_{UV}$
$R_G(u)$	The set of u's interacting groups in $\mathcal{G}_{UG}$
$R_U(v)$	The set of v's interacting users in $\mathcal{G}_{UV}$
$R_G(v)$	The set of v's interacting groups in $\mathcal{G}_{GV}$
$\mathbf{p}_u, \mathbf{q}_v, \mathbf{r}_g$	Inherent embeddings of u, v, and g
$\hat{\mathbf{p}}_u, \hat{\mathbf{q}}_v, \hat{\mathbf{r}}_g^v$	Representations of u, v, and g
$\mathbf{p}_u^{(v)}$	Item-view user embedding of user u
$\mathbf{p}_u^{(g)}$	Group-view user embedding of user u
$\mathbf{q}_v^{(u)}$	User-view item embedding of item v
$\mathbf{q}_v^{(g)}$	Group-view item embedding of item v
$\alpha_j^{(v)}$	Attention of item j in contributing to $\mathbf{p}_u^{(v)}$
$\alpha_l^{(g)}$	Attention of group l in contributing to $\mathbf{p}_u^{(g)}$
$\beta_i^{(u)}$	Attention of user i in contributing to $\mathbf{q}_v^{(u)}$
$\beta_l^{(g)}$	Attention of group l in contributing to $\mathbf{q}_v^{(g)}$
γ_u^v	Item-specific influence weight of user u for $\hat{\mathbf{r}}_g^v$
$\mathbf{W}, \mathbf{b}$	The weight and bias in neural networks

3 PRELIMINARIES

This section presents the preliminaries for our proposed GAME which models the representations of users, items and groups from multiple views based on the interaction graph. Note that we usually refer the term of *representation* to the final multi-view embedding of a user, item or group. For conveniently understanding, Table 1 lists the key symbols used in this work. In group recommendation, there are three sets of entities: a user set U ($|U| = \mathtt{M}$), an item set V ($|V| = \mathtt{N}$), and a group set G ($|G| = \mathtt{L}$). We introduce some definitions based on these three sets.

Definition 1. *Interaction graph.* An interaction graph describes heterogeneous connections among users, groups and items in group recommendation. It is denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of nodes $\{U \cup G \cup V\}$ and $\mathcal{E}$ is the set of edges $\{E_{UG} \cup E_{GV} \cup E_{UV}\}$, where E_{UG} is the subset of edges between users and groups, E_{GV} is the subset of edges between groups and items, and E_{UV} is the subset of edges between users and items. Note that the whole interaction graph $\mathcal{G}$ may contain three bipartite subgraphs, user-group interaction graph $\mathcal{G}_{UG} = \{U \cup G, E_{UG}\}$, group-item interaction graph $\mathcal{G}_{GV} = \{G \cup V, E_{GV}\}$, and user-item interaction graph $\mathcal{G}_{UV} = \{U \cup V, E_{UV}\}$.

From $\mathcal{G}_{UG}$, we define $R_G(u)$ as the set of groups that user u has participated in (i.e., user u's interacting groups in $\mathcal{G}_{UG}$). Similarly, $R_V(u)$ denotes the set of items that user u has interacted with (i.e., user u's interacting items in $\mathcal{G}_{UV}$); $R_U(v)$ denotes the set of users who have interacted with item v (i.e., item v's interacting users in $\mathcal{G}_{UV}$); and $R_G(v)$ denotes the set of groups that have interacted with item v (i.e., item v's interacting groups in $\mathcal{G}_{GV}$).

Definition 2. *Inherent embedding.* Each user $u \in U$, item $v \in V$, or group $g \in G$ is inherently embedded as a vector, denoted as $\mathbf{p}_u \in \mathbb{R}^{\mathtt{D}}$, $\mathbf{q}_v \in \mathbb{R}^{\mathtt{D}}$ or $\mathbf{r}_g \in \mathbb{R}^{\mathtt{D}}$, respectively, where $\mathtt{D}$ denotes the dimension of a vector.

An inherent embedding reflects the inherent interests of a user (or group) or the inherent features of an item. Our GAME will model *item/group-view user embeddings* and *user/group-view item embeddings* based on the inherent embeddings and interaction graph in Section 4.

Definition 3. *Problem statement.* Given a group g, the objective is to recommend top-K items that are new to g and g should be interested in. Note that a new item to g, is defined as that g has not interacted with the item in history.

4 OUR PROPOSED GAME MODEL

This section describes GAME for group recommendation. First, we present the architecture of GAME in Section 4.1. Then, we describe the representation learning in Section 4.2 and neural interaction learning between groups (or users) and items in Section 4.3. Finally, the training optimization of GAME is presented in Section 4.4.

4.1 Model architecture

Figure 2 presents the architecture of GAME for group recommendation with two major parts. (1) *Representation learning*. This part consists of the first three layers: the inherent embedding layer, the layer with multiple Graph-Aware Single-view Embeddings (i.e., GASEs layer) to model the representations of users and items, and the attention layer which learns the representations of groups. (2) *Neural interaction learning*. This part contains the interaction layer and prediction layer, where the former investigates the interaction between the obtained representations of a group (or a user) and an item, and the later predicts the preference score for the group on the item.

To be more specific: (1) The inherent embedding layer takes the interaction graph consisting of groups, users and items as the input. (2) The GASEs layer learns representations of users and items by modeling the interactions on their counterparts in $\mathcal{G}$, as shown in Figures 3 and 4. (3) In the attention layer, the aggregation model learns the representations of groups by aggregating their members' representations. (4) In the interaction layer, the NCF [17] combining matrix factorization [25] with an MLP is applied to learn a non-linear interaction function with respect to groups (or users) and items, and generate the hidden interaction vector. (5) The prediction layer with a fully connected layer (FCL) takes the hidden interaction vector as the input and returns the final prediction score.

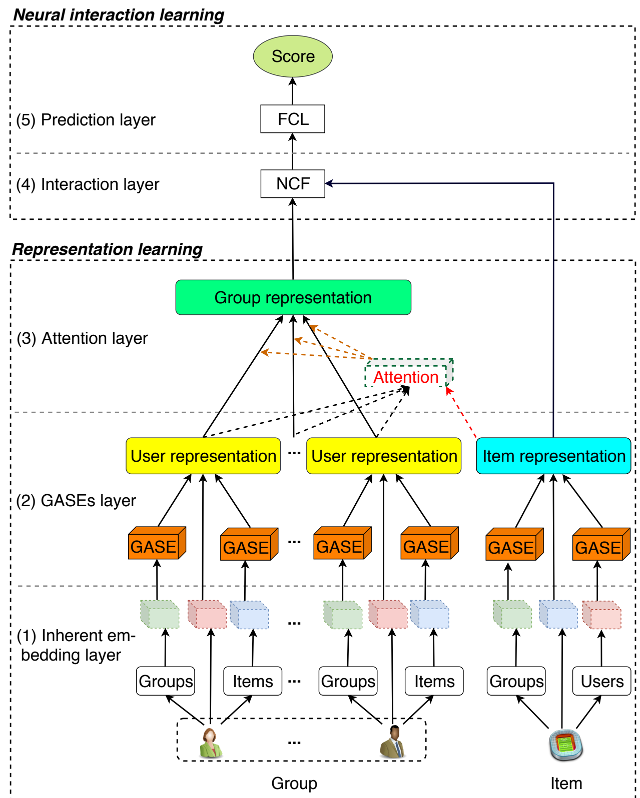


Figure 2: The architecture of GAME consists of two major parts (with five layers): *representation learning* and *neural interaction learning*. The representation learning part includes the first three layers that work for learning representations of users, groups and items. The neural interaction learning part consists of the last two layers which predict the preference score of the group on the target item.

4.2 Representation learning

To learn the representations for users, groups, and items, we first introduce their inherent embeddings which indicate inherent interests of users (or groups) or inherent features of items as shown in Definition 2. These embeddings will be updated during training.

4.2.1 User representation modeling. GAME models a user's representation from three views including her inherent interests, interacting items, and participating groups. Figure 3 depicts the components for learning user representations: the left GASE learns the item-view user embedding from items based on the user-item interaction graph $\mathcal{G}_{UV}$, the right GASE learns the group-view user embedding from groups based on the user-group interaction graph

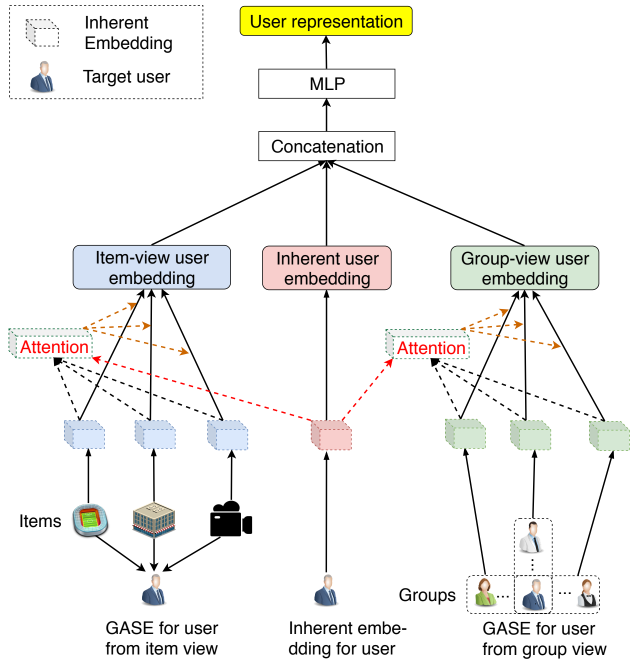


Figure 3: GAME for users. The user representation is modeled from three views, i.e., learning an embedding from each view. Specifically, they are the *inherent user embedding* to describe the user's inherent interests, *item-view user embedding* to capture items' influences based on the user-item interaction graph $\mathcal{G}_{UV}$, and *group-view user embedding* to reflect groups' influences based on the user-group interaction graph $\mathcal{G}_{UG}$. The final user representation is modeled via an MLP with the input of the concatenation of the three types of user embeddings.

$\mathcal{G}_{GV}$, and an MLP combines the item-view and group-view user embeddings with the inherent user embedding for obtaining the final user representation. We present the details as follows.

Item-view user embedding. The preference of a user is often indicated by her interacting items. For example, a foodie likes visiting restaurants to taste various foods and a tourism enthusiast prefers traveling all over the world to view tourism attractions [51]. Thus, it is helpful to derive the user embedding from the item view based on the user-item interaction graph $\mathcal{G}_{UV}$. To this end, GAME develops a GASE to model the item-view user embedding $\mathbf{p}_u^{(v)}$ as the weighted sum over the inherent embeddings of interacting items $R_V(u)$ in the graph $\mathcal{G}_{UV}$, given by

$$\mathbf{p}_u^{(v)} = GASE(\mathcal{G}_{UV}, u) = \sum_{j \in R_V(u)} \alpha_j^{(v)} \mathbf{q}_j, \tag{1}$$

where $\mathbf{q}_j$ is the inherent embedding of item j, and $\alpha_j^{(v)}$ is the dynamic attention weight of the interaction with item j in contributing to user u's item-view user embedding when characterizing her preference from her user-item interaction history $R_V(u)$. The attention weight $\alpha_j^{(v)}$ is learned by an attentive neural network with the inputs of inherent user embedding $\mathbf{p}_u$ and inherent item embedding $\mathbf{q}_j$ for each item $j \in R_V(u)$, given by

$$\alpha_j^{(v)} = Softmax[\mathbf{h}^{(v)\top} \sigma(\mathbf{W}^{(v)}([\mathbf{p}_u, \mathbf{q}_j]) + \mathbf{b}^{(v)})], \tag{2}$$

where the Softmax function is used to normalize values; $\mathbf{h}^{(v)}$, $\mathbf{W}^{(v)}$, and $\mathbf{b}^{(v)}$ are learnable parameters, and σ is an activation function.

Group-view user embedding. A group usually has its own themes that have impacts on its members. For instance, the members in a group for sports may prefer hiking and rock climbing, whereas the members in a group for entertainment would like to watch movies and play games [18]. Therefore, GAME also captures the user embedding from the group view based on the user-group interaction graph $\mathcal{G}_{UG}$. Analogously, GAME exploits another GASE to model the group-view user embedding $\mathbf{p}_u^{(g)}$ as the weighted sum over the inherent embeddings of interacting groups $R_G(u)$ in the graph $\mathcal{G}_{UG}$, given by

$$\mathbf{p}_u^{(g)} = GASE(\mathcal{G}_{UG}, u) = \sum_{l \in R_G(u)} \alpha_l^{(g)} \mathbf{r}_l, \tag{3}$$

where $\mathbf{r}_l$ is the inherent embedding of group l, and $\alpha_l^{(g)}$ is the attention weight of group l in contributing to user u's group-view user embedding when characterizing her preference from her user-group interaction history $R_G(u)$. Similarly, $\alpha_l^{(g)}$ is obtained by an attentive neural network, as in Equation (2) with the inputs of inherent user embedding $\mathbf{p}_u$ and inherent group embedding $\mathbf{r}_l$ for each group $l \in R_G(u)$.

User representation. Obviously, the *inherent*, *item-view* and *group-view* user embeddings (i.e., $\mathbf{p}_u$, $\mathbf{p}_u^{(v)}$ and $\mathbf{p}_u^{(g)}$, respectively) imply the comprehensive preference of a user from different perspectives. It is advisable to merge the three types of user embeddings as the user representation for group recommendation. A simple method is to concatenate the three embeddings together as the user representation. However, this naive method considers the three types of embeddings equally important for each user and does not investigate the influences among themselves. Hence, our GAME performs an MLP over their concatenation to model their relative importance and influences to one another. Accordingly, the user representation $\hat{\mathbf{p}}_u$ for user u is finally obtained by

$$\hat{\mathbf{p}}_u = MLP_{user}([\mathbf{p}_u, \mathbf{p}_u^{(v)}, \mathbf{p}_u^{(g)}]). \tag{4}$$

4.2.2 Item representation modeling. Like user representation modeling, GAME also represents an item from three views containing its inherent features, interacting users, and interacting groups. Figure 4 shows the components for learning item representations: the left GASE models the user-view item embedding from users based on the user-item interaction graph $\mathcal{G}_{UV}$, the right GASE learns the group-view item embedding from groups based on the group-item interaction graph $\mathcal{G}_{GV}$, and an MLP fuses the user-view and group-view item embeddings with the inherent item embedding to get the final item representation. The details are described as follows.

User-view item embedding. An item's characteristics can be reflected by the users who have interacted with it. For instance, the genre of a movie is highly possible to be "cartoon" if the movie is attractive to children or "action" if a plenty of young men are attracted. Hence, it is also benefited to model the item embedding

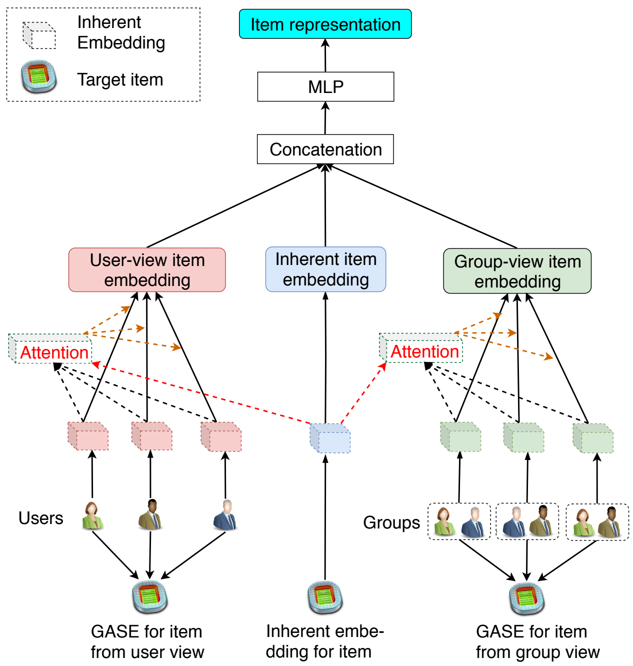


Figure 4: GAME for items. The item representation is modeled from three views, i.e., learning an embedding from each view. They are the *inherent item embedding* to describe the item's inherent features, *user-view item embedding* to capture users' influences based on the user-item interaction graph $\mathcal{G}_{UV}$, and *group-view item embedding* to reflect groups' influences based on the group-item interaction graph $\mathcal{G}_{GV}$. An MLP is finally applied to merge the embeddings from the three different views as the final item representation.

from the user view based on the user-item interaction graph $\mathcal{G}_{UV}$. Symmetrically, GAME devises a GASE to deduce the user-view item embedding $\mathbf{q}_v^{(u)}$ that is the weighted sum over the inherent embeddings of interacting users $R_U(v)$ in the graph $\mathcal{G}_{UV}$, given by

$$\mathbf{q}_v^{(u)} = GASE(\mathcal{G}_{UV}, v) = \sum_{i \in R_U(v)} \beta_i^{(u)} \mathbf{p}_i, \tag{5}$$

where $\beta_i^{(u)}$ is the attention weight of user u in contributing to the user-view item embedding when learning the item representation from its user-item interaction history $R_U(v)$. $\beta_i^{(u)}$ is computed by an attentive neural network, as in Equation (2) with the inputs of inherent item embedding $\mathbf{q}_v$ and inherent user embedding $\mathbf{p}_i$ for each user $i \in R_U(v)$.

Group-view item embedding. The group-item interaction data are very sparse and an occasional group often has only one interaction on a certain item. Fortunately, an item may interact with multiple occasional groups, so the group-item interaction data are still useful for learning item representations. In particular, the item embedding is modeled from the group view by another GASE based on the group-item interaction graph $\mathcal{G}_{GV}$. Accordingly, the group-view item embedding $\mathbf{q}_v^{(g)}$ is aggregated from the inherent embeddings of interacting groups $R_G(v)$ in the graph $\mathcal{G}_{GV}$, given by

$$\mathbf{q}_v^{(g)} = GASE(\mathcal{G}_{GV}, v) = \sum_{l \in R_G(v)} \beta_l^{(g)} \mathbf{r}_l, \tag{6}$$

where $\beta_l^{(g)}$ is the attention weight of group l in contributing to the group-view item embedding when learning the item representation from the group-item interaction history $R_G(v)$ of item v. $\beta_l^{(g)}$ is inferred from an attentive neural network, as in Equation (2) with the inputs of inherent item embedding $\mathbf{q}_v$ and inherent group embedding $\mathbf{r}_l$ for each group $l \in R_G(v)$.

Item representation. The *inherent*, *user-view* and *group-view* item embeddings (i.e., $\mathbf{q}_v$, $\mathbf{q}_v^{(u)}$ and $\mathbf{q}_v^{(g)}$, respectively) reflect the different characteristics of items. It is better to fuse the three types of item embeddings to represent an item comprehensively for group recommendation. It is similar to user representations investigating the relative importance and influences among different embeddings, GAME leverages another MLP on the three types of item embeddings. Correspondingly, the item representation $\hat{\mathbf{q}}_v$ for item v is given by

$$\hat{\mathbf{q}}_v = MLP_{item}([\mathbf{q}_v, \mathbf{q}_v^{(u)}, \mathbf{q}_v^{(g)}]). \tag{7}$$

4.2.3 *Group representation modeling.*

As mentioned before, an occasional group is usually formed temporally; at most cases, an occasional group has one interaction on a certain item. It is not significant to learn the embedding for an occasional group from the item view due to the lack of historical interactions on items for the occasional group. Our experiments also have verified that the item view for occasional groups cannot improve the recommendation performance and we do not show these experimental results in Section 5. Therefore, GAME concentrates on capturing the embedding or representation of a group from the user view.

In particular, GAME models the group representation in terms of its members' *user representations* instead of *inherent user embeddings*, because the user representation is derived from multi-view user embeddings in Equation (4) and constitutes the comprehensive preference of a member. Moreover, each member has the different influence on the preference of a group. For instance, the children greatly affect a family as a group to watch what kind of movies and a sport group is usually leaded by the professional coaches to visit what kind of venues. Accordingly, GAME considers the influence weight of each member on a group when learning the group representation. Further, the influence weight of a member on a given group is not static and should vary for different target items, since people usually have dynamic attentions on different things. For example, a member does not have the same experience on every target item and she may be very familiar with some items but totally new to other items.

Formally, given group g and target item v, the group representation $\hat{\mathbf{r}}_g^v$ is the weighted sum over the user representations $\hat{\mathbf{p}}_u$ of all members $u \in g$:

$$\hat{\mathbf{r}}_g^v = \sum_{u \in g} \gamma_u^v \hat{\mathbf{p}}_u, \tag{8}$$

where γ_u^v is the item-specific influence weight of user u in contributing to the group representation of group g on target item v.

It is estimated by an attentive neural network with the Softmax function:

$$\gamma_u^v = Softmax[\hat{\mathbf{h}}^\top \sigma(\hat{\mathbf{W}}([\hat{\mathbf{p}}_u, \hat{\mathbf{q}}_v]) + \hat{\mathbf{b}})] \text{ for } u \in g, \tag{9}$$

where $\hat{\mathbf{h}}$, $\hat{\mathbf{W}}$, and $\hat{\mathbf{b}}$ are learnable parameters. It is worth emphasizing that: (1) The item representation $\hat{\mathbf{q}}_v$ from multiple views in Equation (7) is utilized to compute the item-specific influence weight γ_u^v. (2) The group representation $\hat{\mathbf{r}}_g^v$ inherits the multi-view property from both user representation $\hat{\mathbf{p}}_u$ and item representation $\hat{\mathbf{q}}_v$. (3) The target item v is not required to be one in the interaction history of the group g, so the item-specific influence weights are not affected by the sparse historical interactions of occasional groups on items.

4.3 Neural interaction learning

4.3.1 Interaction layer. From the work [17], the MLP could achieve better performance than conventional matrix factorization [25] in interaction learning. GAME employs an interaction layer with the inputs of the obtained representations of groups (or users) and items to learn the interaction between group g (or user u) and item v. Formally, given the dot-product $\mathbf{c}_0 = [\hat{\mathbf{r}}_g^v \odot \hat{\mathbf{q}}_v]$ or $\mathbf{c}_0 = [\hat{\mathbf{p}}_u \odot \hat{\mathbf{q}}_v]$, the hidden interaction vector $\mathbf{c}$ is computed by

$$\mathbf{c} = MLP_{interaction}(\mathbf{c}_0), \tag{10}$$

which aims to model the non-linear and non-trivial group-item or user-item relationships.

4.3.2 Prediction layer. Finally, GAME utilizes a FCL with the Sigmoid activation function to predict the score of group g (or user u) on item v:

$$\hat{s}'_{g,v} = Sigmoid(\mathbf{w}^\top \mathbf{c} + b) \text{ for group or} \tag{11}$$

$$\hat{s}_{u,v} = Sigmoid(\mathbf{w}^\top \mathbf{c} + b) \text{ for user,} \tag{12}$$

where $\mathbf{w}$ and b are learnable parameters.

4.4 Training optimization

Since we recommend top-K items for groups based on implicit feedback, we adopt pairwise learning for optimizing parameters in GAME. The pairwise learning considers *observed interactions as positive instances* and *unobserved interactions as negative instances*, and assumes that the predicted scores for positive instances are higher than that for negative instances [5, 42]. Due to the sparsity of group-item interaction data, we apply a two-stage training strategy. The first stage learns the users' preferences by minimizing the pairwise loss function in Equation (13) on the user-item interaction instances:

$$\mathcal{L}_u = \sum_{(u,v,v')\in O} (\hat{s}_{u,v} - \hat{s}_{u,v'} - 1)^2, \tag{13}$$

where O is the set of user training instances, each instance (u, v, v') indicates that user u has interacted with item v but not interacted with item v', and $\hat{s}_{u,v}$ and $\hat{s}_{u,v'}$ are the predicted scores for v and v' from Equation (12), respectively. This loss optimizes the score difference close to one. Similarly, the second stage minimizes the pairwise loss function in Equation (14) on the group-item interaction instances:

$$\mathcal{L}_g = \sum_{(g,v,v')\in O'} (\hat{s}'_{g,v} - \hat{s}'_{g,v'} - 1)^2, \tag{14}$$

where O' is the set of group training instances, each instance (g, v, v') indicates that group g has interacted with item v but not interacted with item v', and $\hat{s}'_{g,v}$ and $\hat{s}'_{g,v'}$ are the predicted scores for v and v' from Equation (11), respectively. Both Equations (13) and (14) are optimized by the Adam optimizer [23].

Table 2: Statistics of the two datasets

Dataset	Meetup-NYC	Meetup-CA
Number of groups (L)	13,785	18,313
Number of users (M)	46,895	59,988
Number of items (N)	3,071	5,036
Avg. No. of users of a group	16.90	13.69
Avg. No. of interacting items of a group	1	1
Avg. No. of participating groups of a user	3.40	3.28
Avg. No. of interacting items of a user	4.97	4.18
Avg. No. of interacting users of an item	15.30	11.91
Avg. No. of interacting groups of an item	4.49	3.64

5 PERFORMANCE EVALUATION

We first describe the experimental settings in Section 5.1 and then present the experimental results and analysis in Section 5.2.

5.1 Experimental settings

5.1.1 Datasets. We conduct experiments on two real datasets: Meetup-NYC and Meetup-CA [30], crawled from Meetup.com in New York City and California, respectively. In the two datasets, an event from Meetup consists of a group of users and a venue. In line with the existing works [18, 36], we consider an event as a group, a user participating in the event as a member, and a venue as an item. The objective is to recommend a venue to hold an event for a group. Table 2 shows the statistics of the two datasets, in which the groups interact with only one item on average because they are occasional or cold-start, as mentioned before.

5.1.2 Compared models. We evaluate the performance of GAME by comparing it with the state-of-the-art models:

- NCF-AVG, NCF-LM and NCF-MS combine NCF [17] with the predefined aggregation strategies including average [2], least misery [1], and maximum satisfaction [4] that take the average, minimal and maximal score of all members as the group's score, respectively.
- FM-AVG [32] combines factorization machine (FM) with the average strategy by viewing all users with equal importance and taking the average embedding of all members in a group as the group representation.
- MoSAN [36] directly sums all members' preferences as the group's preference. Each member's preference is aggregated from the inherent embeddings of the rest members in the group by a sub-attention network that estimates the weights from the rest members to the member.

Table 3: Overall performance comparison on the Meetup-NYC and Meetup-CA datasets

Dataset	Meetup-NYC						Meetup-CA					
Metric	*HR@K*			*NDCG@K*			*HR@K*			*NDCG@K*		
	$K = 5$	$K = 10$	$K = 20$	$K = 5$	$K = 10$	$K = 20$	$K = 5$	$K = 10$	$K = 20$	$K = 5$	$K = 10$	$K = 20$
NCF-AVG	0.8103	0.8237	0.8310	0.7504	0.7547	0.7566	0.7662	0.7810	0.7922	0.7228	0.7277	0.7305
NCF-LM	0.6322	0.6971	0.7624	0.5498	0.5708	0.5873	0.6472	0.6996	0.7455	0.5728	0.5897	0.6012
NCF-MS	0.7831	0.8027	0.8168	0.7131	0.7194	0.7230	0.7583	0.7676	0.7832	0.7200	0.7231	0.7270
FM-AVG	0.7729	0.7947	0.8147	0.7333	0.7404	0.7454	0.7324	0.7488	0.7684	0.7030	0.7083	0.7133
MoSAN	0.7784	0.7922	0.8096	0.7454	0.7499	0.7542	0.7362	0.7518	0.7725	0.6999	0.7049	0.7100
AGREE	0.8172	0.8252	0.8364	0.7580	0.7606	0.7634	0.7687	0.7870	0.7998	0.7242	0.7301	0.7334
GRADI	0.8027	0.8136	0.8317	0.7633	0.7668	0.7714	0.7701	0.7870	0.8132	0.7393	0.7448	0.7512
GAME	**0.8212**	**0.8332**	**0.8915**	**0.7736**	**0.7774**	**0.7918**	**0.7895**	**0.8394**	**0.9418**	**0.7529**	**0.7686**	**0.7944**

- AGREE [5] aggregates the users' preferences (inherent embeddings) with the group preference (inherent embedding) via a standard attention network and adopts NCF to model the interactions between groups and items.
- GRADI [18] is a group recommendation model using attentive dual influences; it simultaneously explores the influences from the members to a group and the influences from the group to its members.

5.1.3 Evaluation metrics. For performance evaluation, each dataset is randomly split into two parts: 80% as the training set and 20% as the test set. Since it is time-consuming to rank all items for a group during evaluation, we randomly select 100 items for a group that are not interacted by the group and rank the test items of the group among the selected 100 items, which is in line with the previous works [5, 9, 17, 24]. Two standard metrics, including Hits Ratio and Normalized Discounted Cumulative Gain at top-K recommendations [5, 17, 47], known as *HR@K* and *NDCG@K*, respectively, are used to measure the recommendation accuracy. For a test instance, namely a group-item pair, *HR@K* measures whether the test item is ranked in the top-K recommendations; *NDCG@K* considers the rank of a hit by assigning a higher value to the hit at a higher rank. We calculate the two metrics for each test instance and report the average.

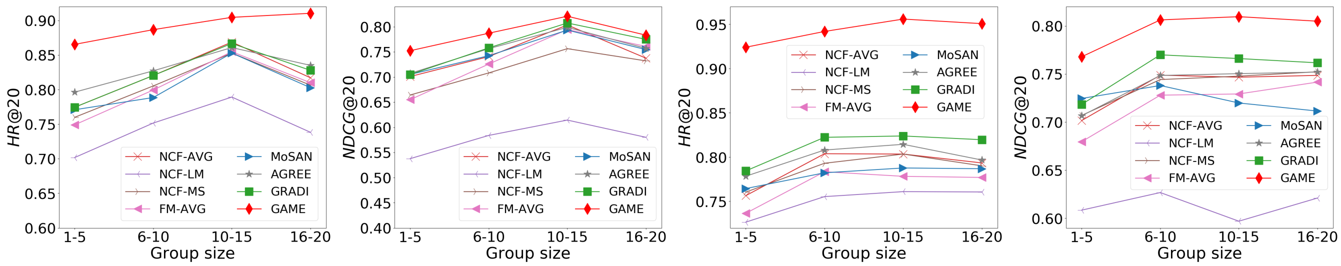
5.1.4 Implementation details. GAME is implemented with the Pytorch and trained by minimizing the pairwise loss in Equations (13) or (14) with four negative instances for each user or group [5, 17]. The embedding layer is initialized by the Glorot strategy [13] and the other layers are randomly initialized by the standard normal distribution. The embedding size is set as $\mathtt{D} = 32$. In attentive neural networks and NCF, we empirically set the first layer size to 32, and employ three layers with the tower structure and ReLU activation function [14]. We apply the Adam optimizer [23] for all models, where the batch size, learning rate, dropout rate, and regularizer are searched in [64, 128, 256, 512], [0.001, 0.005, 0.05, 0.01], [0, 0.1, 0.3, 0.5, 0.8], and $[0, 1e-5, 1e-4, 0.001, 0.01]$, respectively.

5.2 Experimental results

We compare our GAME with the state-of-the-art group recommendation models in Section 5.2.1, investigate the effect of group sizes and performance on cold-start items in Sections 5.2.2 and 5.2.3, respectively, and study the components of GAME in Section 5.2.4.

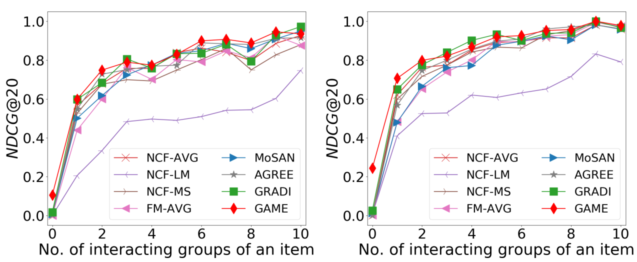
5.2.1 Overall performance comparison. Table 3 shows the best performance of all the models under different settings in terms of *HR@K* and *NDCG@K* on the both datasets. **We have four important observations.** (1) Among NCF-based models, NCF-AVG approaches higher values of *HR* and *NDCG* than NCF-LM and NCF-MS, because the two datasets consist of occasional groups with unstable members and only a few interactions, and the aggregation strategies of NCF-LM and NCF-MS only focus on the minority of members. (2) NCF-AVG, AGREE and GRADI which investigate interactions with NCF perform better than FM-AVG and MoSAN with FM. Our explanation is that NCF models the interactions by combining the linearity of matrix factorization and non-linearity of MLP while FM models the interactions via taking the average embedding of the users in a group and considering all users equally important. (3) Both AGREE and GRADI are generally better than NCF-based models (i.e., NCF-AVG, NCF-MS and NCF-LM), as they consider the distinct importance of the members in a group. Moreover, GRADI is superior to AGREE at most cases, because GRADI explores the dual influences between a group and its members. (4) Most importantly, our GAME always performs the best and its advantage becomes large with the increase of top-K. The reason is that GRADI does not consider the influences from the counterpart views when modeling users and items. This indicates the effectiveness of GAME by modeling the representations of users, items and groups from their counterpart views based on the interaction graph via attentive neural networks. In sum, these experimental results validate the superior performance of GAME for occasional group recommendation.

5.2.2 Effect on group sizes. To investigate the effect of group sizes on the recommendation performance, the group sizes (i.e., number of members in a group) are divided into four classes (1-5, 6-10, 11-15 and 16-20). Note that few groups have more than 20 members and are omitted in this experiment. Due to limited space and similar



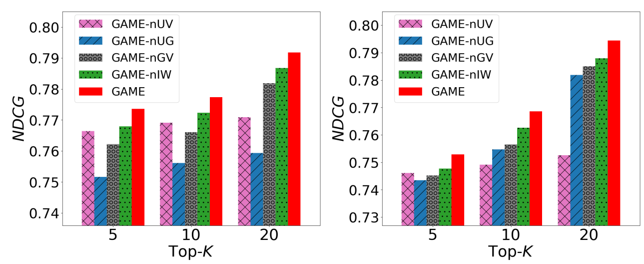
(a) Meetup-NYC **(b) Meetup-NYC** **(c) Meetup-CA** **(d) Meetup-CA**

Figure 5: Effect of group sizes.



(a) Meetup-NYC **(b) Meetup-CA**

Figure 6: Performance on cold-start items.



(a) Meetup-NYC **(b) Meetup-CA**

Figure 7: Performance on the variants of GAME.

results, Figure 5 only depicts the performance of all the models at top 20 recommendations. GAME achieves significantly better performance than other models with respect to different group sizes on the both datasets, which is consistent with the overall performance in Table 3. Especially, the margin between GAME and other models for smaller groups is higher than that for larger groups, which indicates the superiority of GAME in representing groups with a few members as GAME exploits more interactions from multiple views based on the interaction graph.

5.2.3 Performance on cold-start items. The recommendation for occasional groups severely suffers from the cold-start problem including both cold-start groups and cold-start items. In reality, most occasional groups are cold-start and have only one interaction on an item, as shown in Table 2. Actually, all the aforementioned results are for the cold-start groups, i.e., occasional groups; here we focus on the cold-start items with few interactions on groups. Due to limited space, Figure 6 shows the performance of all the models on *NDCG* that measures the rank position and is stricter than *HR*. *NDCG* generally increases as the number of interacting groups of an item gets large, since more interactions will provide more information for learning the item representation. More importantly, GAME is very competitive to the other models and shows the best performance at most cases, especially on the new items without any interaction on groups. The main reason is that GAME models the item representation from multiple views based on the interaction graph and the group representation based on the item-specific influence weights.

5.2.4 Study on the components of GAME. Here we study the components of GAME by evaluating the four variants: (1) GAME-nUV without the embeddings from the user-item interaction graph (i.e., the item-view user embedding and user-view item embedding), (2) GAME-nUG without the embedding from the user-group interaction graph (i.e., the group-view user embedding), (3) GAME-nGV without the embedding from the group-item interaction graph (i.e, the group-view item embedding), and (4) GAME-nIW without the influence weights of members for the group representation (i.e., taking the average of members' user representations).

Figure 7 depicts the recommendation performance of the four variants in comparison to GAME on *NDCG*. In general, we have three important observations. (1) GAME always obtains better results than GAME-nUV, GAME-nUG, GAME-nGV and GAME-nIW on the two datasets. Therefore, it makes sense to derive the multi-view embeddings from the whole interaction graph and the group representation based on the item-specific influence weights of members for group recommendation. (2) GAME-nIW performs slightly worse than GAME but better than GAME-nUV, GAME-nUG and GAME-nGV, which implies that the multi-view embeddings from the interaction graph are more important than the item-specific influence weights of members. Thus, it is more significant to extensively exploit the interaction data from multiple views than to devise a new method for aggregating member representations as the group representation. (3) GAME-nUG reports worse recommendation accuracy than GAME-nUV and GAME-nGV at all cases on the Meetup-NYC dataset and most cases on the Meetup-CA dataset. This reflects that the group-view user embeddings from the user-group interaction graph are more useful than the item-view user embeddings and user-view item embeddings from the user-item interaction graph, and the group-view item embeddings from the group-item interaction graph. The reason is that the inherent embeddings are directly learned from both user-item and group-item interaction data based on the two-stage training strategy in Section 4.4, so they can capture some interactions of users or groups on items, and compensate the missing item-view user embeddings, user-view item embeddings, and group-view item embeddings.

6 CONCLUSION AND FUTURE WORK

In this paper, we have proposed the new model GAME for occasional group recommendation by learning graphical and attentive multi-view embeddings of users, items and groups. Specifically, GAME models the representations of users and items through taking full advantage of the heterogeneous information in the interaction graph consisting of users, items and groups, i.e., learning the three types of embeddings for users and items from one independent view and two counterpart views. Further, it learns the item-specific influence weights of members via the attention mechanism to aggregate the members' representations as the group representation. Thus, GAME effectively alleviates the sparse issue of the group-item interaction data and is able to make group recommendation better based on neural collaborative filtering. The experimental results on the two real datasets show the superiority of GAME compared to the other state-of-the-art models, especially for both cold-start groups and cold-start items. However, GAME requires more computation costs than other models, because it needs to calculate the attention in multiple views. Currently, this work only considers the observed graphs for group recommendation. In reality, the hidden graphs (e.g., social influences between users or groups) also affect the recommendation performance, which is one direction of our future works.

REFERENCES

[1] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu. Group recommendation: Semantics and efficiency. In *VLDB*, 2009.
[2] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *ACM RecSys*, 2010.
[3] S. Berkovsky and J. Freyne. Group-based recipe recommendations: Analysis of data aggregation strategies. In *ACM RecSys*, 2010.
[4] L. Boratto and S. Carta. State-of-the-art in group recommendation and new approaches for automatic identification of groups. In *Information Retrieval and Mining in Distributed Environments*, volume 324, pages 1–20. Springer Berlin Heidelberg, 2010.
[5] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong. Attentive group recommendation. In *ACM SIGIR*, 2018.
[6] A. J. Chaney, M. Gartrell, J. M. Hofman, J. Guiver, N. Koenigstein, P. Kohli, and U. Paquet. A large-scale exploration of group viewing patterns. In *ACM TVX*, 2014.
[7] W. Cheng, Y. Shen, Y. Zhu, and L. Huang. DELF: A dual-embedding based deep latent factor model for recommendation. In *IJCAI*, 2018.
[8] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi. Visual tracking using attention-modulated disintegration and integration. In *IEEE CVPR*, 2016.
[9] A. M. Elkahky, Y. Song, and X. He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*, 2015.
[10] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin. Graph neural networks for social recommendation. In *WWW*, 2019.
[11] W. Fan, Y. Ma, D. Yin, J. Wang, J. Tang, and Q. Li. Deep social collaborative filtering. In *ACM RecSys*, 2019.
[12] S. Feng, H. Zhang, L. Wang, L. Liu, and Y. Xu. Detecting the latent associations hidden in multi-source information for better group recommendation. *Knowledge-Based Systems*, 171(1):56–68, 2019.
[13] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.
[14] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *AISTATS*, 2011.
[15] C. Guo, B. Li, and X. Tian. Flickr group recommendation using rich social media information. *Neurocomputing*, 204(5):8–16, 2016.
[16] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. DeepFM: A factorization-machine based neural network for CTR prediction. In *IJCAI*, 2017.
[17] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua. Neural collaborative filtering. In *WWW*, 2017.
[18] Z. He, C.-Y. Chow, and J.-D. Zhang. GRADI: Towards group recommendation using attentive dual top-down and bottom-up influences. In *IEEE BigData*, 2019.
[19] Z. He, C.-Y. Chow, and J.-D. Zhang. STANN: A spatio-temporal attentive neural network for traffic prediction. *IEEE Access*, 7:4795–4806, 2019.
[20] D. Herzog and W. Wörndl. A user study on groups interacting with tourist trip recommender systems in public spaces. In *ACM UMAP*, 2019.
[21] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
[22] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and W. Cao. Deep modeling of group preferences for group-based recommendation. In *AAAI*, 2014.
[23] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
[24] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *ACM SIGKDD*, 2008.
[25] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 424(8):30–37, 2009.
[26] X. Liu, Y. Tian, M. Ye, and W. Lee. Exploring personal impact for group recommendation. In *ACM CIKM*, 2012.
[27] J. F. McCarthy and D. A. Theodore. MusicFX: An arbiter of group preferences for computer supported collaborative workouts. In *ACM CSCW*, 1998.
[28] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. PolyLens: A recommender system for groups of users. In *ECSCW*, 2001.
[29] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *KDDCup*, 2007.
[30] T.-A. N. Pham, X. Li, G. Cong, and Z. Zhang. A general recommendation model for heterogeneous networks. *IEEE TKDE*, 28(12):3140–3153, 2016.
[31] E. Quintarelli, E. Rabosio, and L. Tanca. Recommending new items to ephemeral groups using contextual user influence. In *ACM RecSys*, 2016.
[32] S. Rendle. Factorization machines. In *IEEE ICDM*, 2010.
[33] A. Said, S. Berkovsky, and E. W. De Luca. Group recommendation in context. In *CAMRa*, 2011.
[34] J. B. Schafer, J. A. tan, and J. Riedl. E-commerce recommendation applications. *DMKD*, 5(1):115–153, 2001.
[35] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.
[36] L. V. Tran, T.-A. N. Pham, Y. Tay, Y. Liu, G. Cong, and X. Li. Interact and decide: Medley of sub-attention networks for effective group recommendation. In *ACM SIGIR*, 2019.
[37] C. Van Der Malsburg. Frank Rosenblatt: Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. In *Proceedings of the First Trieste Meeting on Brain Theory*, 1986.
[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.
[39] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *IEEE CVPR*, 2017.
[40] W. Wang, G. Zhang, and J. Lu. Member contribution-based group recommender system. *Decision Support Systems*, 87:80–93, 2016.
[41] W. Wang, G. Zhang, and J. Lu. Hierarchy visualization for group recommender systems. *IEEE TSMC*, 49(6):1152–1163, 2017.
[42] X. Wang, X. He, L. Nie, and T.-S. Chua. Item silk road: Recommending items from information domains to social users. In *ACM SIGIR*, 2017.
[43] X. Wang, Y. Liu, J. Lu, F. Xiong, and G. Zhang. TruGRC: Trust-aware group recommendation with virtual coordinators. *FGCS*, 94:224 – 236, 2019.
[44] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang. A neural influence diffusion model for social recommendation. In *ACM SIGIR*, 2019.
[45] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong. Deep item-based collaborative filtering for top-n recommendation. *ACM TOIS*, 37(3):33:1–33:25, 2019.
[46] M. Ye, X. Liu, and W.-C. Lee. Exploring social influence for recommendation: A generative model approach. In *ACM SIGIR*, 2012.
[47] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou. Social influence-based group representation learning for group recommendation. In *IEEE ICDE*, 2019.
[48] H. Yin, L. Zou, Q. V. H. Nguyen, Z. Huang, and X. Zhou. Joint event-partner recommendation in event-based social networks. In *IEEE ICDE*, 2018.
[49] Z. Yu, X. Zhou, Y. Hao, and J. Gu. TV program recommendation for multiple viewers based on user profile merging. *UMUAI*, 16(1):63–82, 2006.
[50] Q. Yuan, G. Cong, and C. Lin. COM: A generative model for group recommendation. In *ACM SIGKDD*, 2014.
[51] J.-D. Zhang and C.-Y. Chow. GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *ACM SIGIR*, 2015.
[52] L. Zheng, C. Li, C. Lu, J. Zhang, and P. S. Yu. Deep distribution network: Addressing the data sparsity issue for top-n recommendation. In *ACM SIGIR*, 2019.