

# An Intelligent Group Event Recommendation System in Social networks

Guoqiong Liao<sup>1,2</sup>, Xiaomei Huang<sup>1,4,\*</sup>, Neal N. Xiong<sup>3</sup>, and Changxuan Wan<sup>1,2</sup>

<sup>1</sup>School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330013, China.

<sup>2</sup>Jiangxi Province Key Laboratory of Data and Knowledge Engineering (Jiangxi University of Finance and Economics), Nanchang 330013, China.

<sup>3</sup>Department of Mathematics and Computer Science, Northeastern State University, OK 74464 USA.

<sup>4</sup>School of Mathematics and Information Science, Jiangxi Normal University, Nanchang 330022, China.

\*Corresponding Author: Xiaomei Huang. Email: huangxm501@126.com.

**Abstract**—The importance of contexts has been widely recognized in recommender systems for individuals. However, most existing group recommendation models in Event-Based Social Networks (EBSNs) focus on how to aggregate group members' preferences to form group preferences. In these models, the influence of contexts on groups is considered but simply defined in a manual way, which cannot model the complex and deep interactions between contexts and groups. In this paper, we propose an Attention-based Context-aware Group Event Recommendation model (ACGER) in EBSNs. ACGER models the deep, non-linear influence of contexts on users, groups, and events through multi-layer neural networks. Especially, a novel attention mechanism is designed to enable the influence weights of contexts on users/groups change dynamically with the events concerned. Considering that groups may have completely different behavior patterns from group members, we propose that the preference of a group need to be obtained from indirect and direct perspectives (called indirect preference and direct preference respectively). In order to obtain the indirect preference, we propose a method of aggregating preferences based on attention mechanism. Compared with existing predefined strategies, this method can flexibly adapt the strategy according to the events concerned by the group. In order to obtain the direct preference, we employ neural networks to directly learn it from group-event interactions. Furthermore, to make full use of rich user-event interactions in EBSNs, we integrate the context-aware individual recommendation task into ACGER, which enhances the accuracy of learning of user embeddings and event embeddings. Extensive experiments on two real datasets from Meetup show that our model ACGER significantly outperforms the state-of-the-art models.

**Index Terms**—Event-based social networks, group recommendation, context, attention, neural network.

## I. INTRODUCTION

EVENT-BASED Social Networks (EBSNs) applications, such as Meetup.com, Douban.com, and Plancast.com, have become increasing popular in recent years. EBSNs provide online platforms for users to create, distribute, organize and register all kinds of social events, which promotes the success of offline interactions among users. The events here could be academic meetings, business exhibitions, dining out, and movies night, etc. In order to alleviate the problem of information overload brought by massive events, many models on recommending events for individuals are proposed [1]–[4].

Since people often participate in offline events in groups in real life, recommending events for groups of people has become research focus in recent years [5]–[8]. Traditional

group recommendation methods focus on how to aggregate member preferences to form the group preference, and pay less attention to the influence of contexts (such as time, location, and social relationship) on group preferences. In fact, contexts may have important impacts on group behaviors. For example, when a group decides on which restaurant is suitable for dining out, besides the type of food, the contextual factors such as the restaurant's location, the parking lot's capacity, and the members' free time will all have impacts on the group's decision.

The impacts of contexts have been widely studied in event recommender systems for individuals [1], [4], [9], [10]. These works study the influence mechanism of contexts on individuals by defining linear or non-linear functions manually. However, manually defined functions are not sufficient to capture the deep, highly non-linear interactions among entities such as users, events, and contexts. Moreover, these works focus on the impacts of contexts on individuals rather than groups, so they cannot be directly applied to event recommendation for groups.

Recently Du et al. [11] studied the group event recommendation problem considering influences of contexts including event content, time, location, and social relationship, and proposed a group recommendation method based on learning-to-rank technology. This method incorporates the influences of various contexts, but the influence functions are defined manually. And the method does not differentiate the influence weights of various contexts. There is a recent work modeling the contextual influences in the field of context-aware recommendation for individuals [12]. It proposes a neural attention mechanism to model the influence weights of contexts on users and events. However, its measurement of influence weights of contexts on users only considers the interaction between contexts and users, neglecting the impacts of events, which is not quite consistent with the fact. For example, the influence weight of the context of season on users will change with the type of events. When a user is faced with winter skiing events, the influence weight of the season is great, but when the user is faced with movie events, the influence weight of the season becomes smaller.

How to acquire group preferences accurately is the core problem of group recommendation. Existing group recommendation algorithms focus on obtaining group preferences by aggregating members' preferences, and various group aggregating

gation strategies have been proposed, such as strategies based on social theory (like approval voting, least misery, average, etc. [8], [13]) and strategies considering users' special needs or users' expertise [14]–[16]. However, these works exist following two limitations: (1) predefined group aggregation strategies lack flexibility. When the type of events concerned changes, the group may adopt a different decision-making strategy, and the predefined strategy cannot adapt to it. For example, a tour group often goes hiking in various scenic spots, and the group usually adopts the average strategy to decide the next spot (i.e., each member has equal weight to the final group decision). When the group is faced with some different type of tour project (such as surfing or skiing), members with experiences in such project may have greater weights to the final decision. Then the previous average strategy is no longer applicable. Therefore, we need to study a more flexible and adaptive group aggregation strategy. (2) a group preference is not completely determined by the historical preferences of group members. For example, in real life, an individual user could choose jogging or walking as his/her leisure sport. When he/she is in a group, the group may still choose jogging or walking as its leisure sport, but may also choose playing football, basketball or other sports which need cooperation with each other. We analyzed the events attended by 50 groups of New York City in 2016 on Meetup website. Fig. 1 shows the characteristics of these group events. It can be seen that a large proportion of group events in most groups are similar to the historical events of members, but there are also proportions of group events that are not similar to the historical events of any member. This suggests that group behaviors may be completely different from individual behaviors. For the convenience of discussion, we call the group preference obtained from the historical preferences of group members as **group indirect preference**, and call the group preference completely different from the historical preferences of members as **group direct preference**. Therefore, a reasonable group recommendation model should model both indirect preference and direct preference of a group.

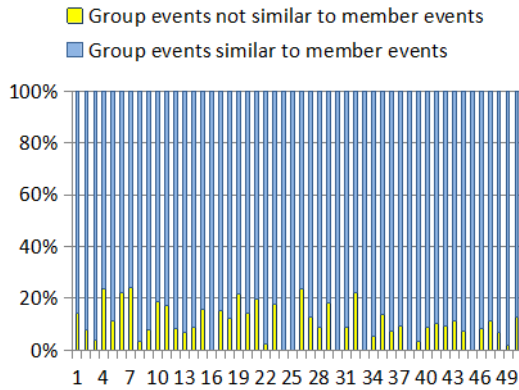


Fig. 1: Characteristics of group events attended by 50 groups in New York city in 2016 on Meetup website.

To address above problems, we propose an Attention-based Context-aware Group Event Recommendation model (ACGER) for EBSNs. The details are as follows.

In order to characterize the complex influence of contexts on users, groups and events, we propose a deep model to learn the representations of users, groups, and events under the influence of contexts. This is motivated by the successful development of deep learning in recent years, which has shown a strong representation ability in image, text, and voice data processing [17]–[19]. The multi-layer architecture including non-linear functions designed in our model can capture the complex and non-linear impacts of contexts on groups, users, and events.

In order to model the situation that the influence weights of contexts may change with the type of events, inspired by the neural attention mechanism [20], [21], we design a neural attention network which learns the influence weights of contexts on users/groups from interactions among users/groups, contexts, and events instead of interactions just between users/groups and contexts.

In order to capture the group preference more accurately, we propose that the calculation of a group preference should include two aspects: the group indirect preference and the group direct preference. The former could be obtained by aggregating members' preferences by employing an attention mechanism, which can learn group aggregation strategy adaptively from data, while the latter could be learned from the historical interactions between groups and events with neural networks.

In summary, the contributions of this paper are listed as follows:

1) We propose an Attention-based Context-aware Group Event Recommendation model (ACGER) in EBSNs. The model can effectively capture the complex and non-linear influence of contexts on users, groups, and events. As far as we know, this is the first work which addresses the context-aware group event recommendation from the perspective of neural representation learning.

2) We design a novel neural attention mechanism, which not only models the interaction between users/groups and contexts, but also incorporates the impacts of events, so that the dynamic change of contextual weights with different events can be captured in time.

3) We propose that the calculation of a group preference should not only consider the indirect preference obtained from group members, but also consider the direct preference which is completely different from the preference of each member. To aggregate member preferences to obtain the indirect preference, an adaptive group aggregation strategy based on a neural attention mechanism is proposed. And the group direct preference is learned from group-event interaction data by neural networks.

4) Extensive experiments on two real datasets from Meetup show that the proposed model ACGER can achieve better recommendation performance compared with the state-of-the-art models.

The rest of this paper is organized as follows. Section II reviews the related works. Section III formulates our problem and presents the framework of our proposed model ACGER. In Section IV, we elaborate the ACGER scheme which includes three main modules. The experiments based on two real-world datasets are conducted and the performance analysis is given

in Section V. The last section VI concludes this paper and points out our future work.

## II. RELATED WORK

In this section, we review some works related to our problem in the literature, including conventional context-aware recommendation methods, context-aware event recommendation for individuals, and context-aware event recommendation for groups.

### A. Conventional Context-Aware Recommendation Methods

Context in the recommender system domain refers to any information that can be used to characterize the situation of an entity. An entity is a person, a place, or an object that is considered relevant to the interaction between a user and a recommender system [22]. Context-aware recommendation algorithms can be classified into three main algorithmic paradigms according to the phase when contextual information is incorporated: contextual pre-filtering, contextual post-filtering, and contextual modeling [6].

In contextual pre-filtering paradigm, contextual information is used for data selection or data construction. Then, ratings can be predicted using any traditional Two-Dimensional (2D) User $\times$ Item recommender system on the selected data. One early work is [23]. It proposes a reduction-based approach, which reduces the problem of multidimensional (MD) contextual recommendations to the standard 2D recommendation space. In this work, the authors also tried to combine several contextual pre-filters into one model at the same time, which provides significant performance improvements over the one pre-filter approaches. [24] proposes User Splitting technique, which splits the user profile into several sub-profiles, and each sub-profile represents the user in a particular context. [25] proposes Item Splitting technique, which splits each item into several fictitious items based on the contexts. [26] splits both users and items in the data set to boost context-aware recommendations.

In contextual post-filtering paradigm, contextual information is initially ignored, and any traditional 2D recommender system could be used on the entire data to predict the ratings. Then, the recommendation result is adjusted by using the contextual information. [27] introduces two contextual post-filtering methods: Weight and Filter. The Weight method adjust the recommendation list by reordering the recommended items according to their probability of relevance in the specific context, and the Filter method filters out recommended items that have low probability of relevance in the specific context. One important benefit of both contextual pre-filtering approaches and contextual post-filtering approaches is that all the previous research on 2D recommender systems could be directly applied. However, all these approaches require manual supervision and fine-tuning in the recommendation process.

In contextual modeling paradigm, contextual information is incorporated directly in the recommendation model as an explicit predictor of a user's rating for an item. Some studies work on contextualize Matrix Factorization (FM) approach. [28] presents Context-Aware Matrix Factorization (CAMF),

which extends MF by considering the influence of contexts on items. [29] extends a matrix factorization method SLIM (Sparse Linear Method) to a Contextual SLIM (CSLIM) incorporating contextual conditions for the top- $N$  recommendation task. However, these approaches cannot handle the ternary relational nature of data. Tensor Factorization (TF) is an extension of MF techniques to incorporate different contexts as multifaceted user-item interactions in the recommendation process. One classical method is Multiverse Recommendation [30], which relies on Tucker decomposition and allows to work with any categorical context. To address the implicit feedback, [31] proposed a ranking-based Tensor Factorization (TF) model by directly maximizing Mean Average Precision. Another significant work is proposed by [32]. It applies Factorization Machines (MF) to model the interactions between each pair of entities in terms of their latent factors, such as user-user, user-item, user-context interactions.

As far as recommendation methods in EBSNs are concerned, the existing works mostly adopt the paradigm of contextual modeling, which directly integrates context information into models to characterize the contextual influences.

### B. Context-Aware Event Recommendation for Individuals

Many models have been proposed in the field of context-aware event recommendation for individuals. Qiao et al. [4] proposed a potential factor model to model online and offline social relations, geographical features of events, and implicit feedback of users in event social networks, so as to recommend offline events for users. Macedo et al. [9] extracted social relations, content, time, and geographical features respectively, and then used the learning to rank technology to combine these contextual information to generate event recommendation. Zhang et al. [1] formulated the cold-start event recommendation problem, using Bayesian Poisson factorization as the basic unit to model different contextual factors, and further combined those units to form a unified model through a collective matrix factorization model. Xu et al. [10] proposed a semantic-enhanced and context-aware hybrid collaborative filtering method, which combines semantic content analysis and contextual event influence for user neighborhood selection. Cao et al. [33] combined multiple features about topology, temporal, spatial, and semantic to model user preferences, which alleviates the problem of data sparseness in EBSNs. With the successful application of deep learning and representation learning in the fields of image, speech and natural language processing in recent years [17]–[19], some researchers have also applied these techniques in context-aware event recommendation. Wang et al. [34] considered the temporal and spatial effects of events, and mapped the event, location, and time into low-dimensional space based on event sequential data by representation learning method. Wang et al. [35] utilized convolutional neural network with word embedding to extract the high-level features of contextual information of a user's interested events and built up a user latent model for each user, then they incorporated the user latent models into a probabilistic matrix decomposition model to obtain more accurate recommendation performance.

However, all these methods do not consider either the influence of contexts on group preferences nor the characteristics of group recommendation task (e.g., how to aggregate different preferences of members into a consistent group preference). Therefore, they cannot be directly applied to EBSN group recommendation.

### C. Context-Aware Event Recommendation for Groups

Group recommendation in EBSNs has attracted more and more attention in recent years. Yuan et al. [36] proposed a probability model COM to simulate the generation process of group activities and perform group recommendations. Purushotham et al. [37] proposed a collaborative filtering algorithm based on the Bayesian model to recommend events for groups considering the potential topics of groups. Ji et al. [38] proposed a topic-based probability model for group recommendation, in which the group preference not only considers the interests of members, but also considers the interests of subgroups. Du et al. [39] proposed a probabilistic generative model to jointly learn groups' content preferences and venue preferences. They discovered a strong correlation between organizers and textual contents. Above methods focus on modeling the generative process of group preferences by utilizing the interaction among group members, lacking a deep investigation on the influence of contexts on group behaviors, resulting in a suboptimal performance for group recommendation. Recently, a method named GEF comprehensively considering the influence of various contexts on group recommendation has been proposed [11]. It first models the influences of contexts including time, place, event content, and social relations on the user's preferences, then merges the preferences of users in a group to form the group preference. Finally, by using a learning to rank algorithm to learn the ranking function for each group, it produces the event recommendation lists for groups. This method relies on manually defined function to characterize the influences of contexts on users and events, which is insufficient to model the complex and highly non-linear influences of contexts. In addition, the influence weights of different contexts are not differentiated in this work. In fact, there are a few works recently, which could be used to model the influence of different contexts on users and events. Among them, [40] proposes Neural Factorization Machines (NFM) which enhances FM by modeling nonlinear feature interactions through neural networks. [21] proposes Attentional Factorization Machines (AFM) that improves FM by differentiating the importance of different feature interactions via a neural attention network. Similar to FM, these two models can be applied to the task of context-aware recommendations by specifying the input data. However, NFM fail to differentiate the different importance of context influences. AFM can automatically differentiate the importance of feature interactions, but it models the feature interactions in a linear way. [12] proposes a novel neural model named AIN to adaptively capture the interactions between contexts and users/items. And a neural attention mechanism is employed to model the influence weights of contexts on users/items. However, the neural attention mechanism of AIN neglects that the influence weight of a context on a user may

change when the type of the item concerned changes. In this paper, we employ the deep neural network and representation learning techniques to model the complex and non-linear interactions between contexts and entities including users, events, and groups, and propose a novel neural attention mechanism to weigh the influence of different contexts more accurately.

How to aggregate different members' preferences into a consistent group preference, i.e., the group aggregation strategy, has always been the focus of group recommendation research. In the early works, Masthoff [13] proposed 10 aggregation strategies based on social choice theory, such as approval voting, Borda counting, least misery, average, etc. Ardissono et al. [14] assigned greater weight to people with special needs (such as children or disabled people). Berkovsky et al. [15] judged a user's activeness based on the number of items he/she has rated, and assigned greater weight to more active users. SEO et al. [8] considered the deviation of group members' opinions combining with the average and voting counting strategies. [16] measured the influence weight of a member by the number of times that his/her preference being consistent with the group's preference. However, the aggregation strategies in these methods are all predefined, which is data independent and lacks flexibility. When the decision-making strategy of a group change, the predefined aggregation strategy cannot adapt to it. To overcome above limitation, we propose an adaptive group aggregation strategy based on the neural attention mechanism, which can learn a member's weight from the data. In addition, the acquisition of a group preference in our method not only considers preferences of members, but also considers the preference that is quite different from each member's, which further improves the performance of group recommendation.

## III. PROBLEM FORMULATION AND MODEL FRAMEWORK

### A. Notations and Problem Formulation

We use bold capital letters (e.g.,  $\mathbf{X}$ ) and bold lowercase letters (e.g.,  $\mathbf{x}$ ) to represent matrices and vectors, respectively. We employ non-bold letters (e.g.  $x$ ) to denote scalars, and squiggle letters (e.g.  $\mathcal{X}$ ) to denote sets.  $|\cdot|$  denotes the cardinality of a set. If not clarified, all vectors are in column forms.

Given context variable  $\mathcal{C}$  consisted of  $k$  contextual factors, i.e.,  $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$ , we suppose each contextual factor  $\mathcal{C}_i$  has  $|\mathcal{C}_i|$  values, denoted as  $\mathcal{C}_i = \{c_1^{(i)}, c_2^{(i)}, \dots, c_{|\mathcal{C}_i|}^{(i)}\}$ , where  $c_j^{(i)}$  ( $j \in \{1, \dots, |\mathcal{C}_i|\}$ ) is the  $j$ th value of the  $i$ th contextual factor  $\mathcal{C}_i$ ,  $\mathcal{L}$  is the set of tuple of values of contextual factors, denoted as  $\mathcal{L} = \{(c^{(1)}, c^{(2)}, \dots, c^{(k)}) | c^{(j)} \in \mathcal{C}_j, j \in \{1, \dots, k\}\}$ . Suppose we have a set of  $n$  users  $\mathcal{U} = \{u_1, u_2, \dots, u_n\}$ , a set of  $m$  events  $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$ , and a set of  $s$  groups  $\mathcal{G} = \{g_1, g_2, \dots, g_s\}$ , where each group is consisted of a certain number of users, and we obtain the group-event interaction matrix  $Y = [y_{ij}]_{s \times m}$  and the user-event interaction matrix  $R = [r_{ij}]_{n \times m}$ . Given a target group  $g \in \mathcal{G}$ , a tuple of values of contextual factors  $c = (c^{(1)}, c^{(2)}, \dots, c^{(k)}) \in \mathcal{L}$ , our task is defined as

recommending a list of events that group  $g$  may be most interested in, i.e., top- $N$  event recommendation for group  $g$ .

### B. Model Framework

The ACGER model proposed in this paper is composed of three main components:

(1) Context-aware **embedding learning** module: given a group-event interaction record, the one-hot feature vectors of the related entities including group, event, group members, and contextual factors are taken as the initial inputs of the model, and they are mapped into low-dimensional and dense vectors through an embedding layer. Then, a **Multi-Layer Perceptron (MLP)** is used to capture the effects of interactions between contexts and users/events/groups, and then we get the enhanced embeddings of users/events/groups under the comprehensive influences of contexts through the neural attention mechanism. And the enhanced group embedding encodes the group's direct preference.

(2) Group preference acquisition module: we aggregate the group members' enhanced embeddings through a neural **attention** network to get the **group's indirect preference**. The indirect preference embedding and the direct preference embedding are combined to get the final group preference embedding.

(3) Score prediction module: the **Factorization Machines (FM)** model is used as the group score prediction layer, and a **pairwise ranking loss** is used for the model optimization.

Noted that in order to employ the abundant user-event interaction data in EBSNs to improve the accuracy of embedding learning, i.e., the learning of the user embeddings and the event embeddings, we integrate the context-aware recommendation for individuals task into ACGER. Specifically, given a user-event interaction record, the one-hot feature vectors of user, event and contextual factors are fed into the model. Through an embedding layer, a multi-layer perceptron and a neural attention network in turn, we get the enhanced embeddings of the user/event under the comprehensive influences of the contextual factors. The FM model is still used to predict the score of a user to the target event. The overall framework of ACGER is shown in Fig. 2.

## IV. OUR PROPOSED ACGER SCHEME

### A. Context-aware Embedding Learning Module

Given a set of contextual factors, the goal of this section is to obtain the feature representations of members, events and groups under the contextual influences. The module can be divided into three components and the detailed description is as follows:

1) Obtain the low-dimensional representations. Given contextual factors  $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$  and the tuple of their current value  $c = (c^{(1)}, c^{(2)}, \dots, c^{(k)})$ , the low-dimensional embedding representation of member  $u$  is calculated as follows:

$$\mathbf{u} = \mathbf{P}^T \tilde{\mathbf{u}}, \quad (1)$$

where  $\mathbf{P} \in \mathbb{R}^{|\mathcal{U}| \times d}$  denotes the user embedding matrix,  $d$  denotes the dimension of the embedding vector,  $T$  denotes the matrix transpose operation. And  $\tilde{\mathbf{u}} \in \{0, 1\}^{|\mathcal{U}|}$  is a one-hot

vector, where the location of element 1 representing which row in the user matrix  $\mathbf{P}$  the user  $u$  corresponds to.

Similarly, we obtain the event  $e$ 's embedding  $\mathbf{e}$  from the event embedding matrix  $\mathbf{Q} \in \mathbb{R}^{|\mathcal{E}| \times d}$ , and get the group  $g$ 's embedding  $\mathbf{g}$  from the group embedding matrix  $\mathbf{Z} \in \mathbb{R}^{|\mathcal{G}| \times d}$ . As for the value  $c^{(i)}$  of the contextual factor  $\mathcal{C}_i$ , its embedding  $\mathbf{c}^{(i)}$  could be obtained from the  $i$ th context embedding matrix  $\mathbf{K}_i \in \mathbb{R}^{|\mathcal{C}_i| \times d}$ .

2) Obtain the embedding representation under the influence of each contextual factor. In order to capture the complex and non-linear impact of each contextual factor on members/events/groups, we use a MLP to map the input data to a deep, non-linear hidden space.

Specifically, we first concatenate the user embedding  $\mathbf{u}$  with the context embedding  $\mathbf{c}^{(i)}$ , then we pass the concatenation through a stack of fully connected layers and finally get  $u$ 's influenced representation in the context of  $\mathcal{C}_i$ . The formulation is as follows:

$$\begin{aligned} \mathbf{f}_{i,0}^\psi &= \text{ReLU}(\mathbf{W}_{i,0}^\psi [\mathbf{u}, \mathbf{c}^{(i)}] + \mathbf{b}_{i,0}^\psi), \\ \mathbf{f}_{i,1}^\psi &= \text{ReLU}(\mathbf{W}_{i,1}^\psi \mathbf{f}_{i,0}^\psi + \mathbf{b}_{i,1}^\psi), \\ &\dots\dots\dots \\ \mathbf{u}^{C_i} &= \text{ReLU}(\mathbf{W}_{i,L}^\psi \mathbf{f}_{i,L-1}^\psi + \mathbf{b}_{i,L}^\psi), \end{aligned} \quad (2)$$

where  $[\cdot, \cdot]$  denotes the concatenation of two vectors,  $\text{ReLU}(\cdot)$  is the Rectifier activation function,  $\mathbf{W}_{i,0}^\psi \in \mathbb{R}^{d \times 2d}$  and  $\mathbf{W}_{i,j}^\psi \in \mathbb{R}^{d \times d} (j \in \{1, \dots, L\})$  are the parameter matrices,  $\mathbf{b}_{i,j}^\psi \in \mathbb{R}^d (j \in \{0, \dots, L\})$  denotes the bias vector,  $\mathbf{f}_{i,j}^\psi \in \mathbb{R}^d (j \in \{0, \dots, L-1\})$  denotes the output vector of the  $j$ th hidden layers. The superscript  $\psi$  indicates the marked model parameters are related to Eq. (2).

As for event  $e$  and group  $g$ , we can similarly get their contextual influenced representations  $\mathbf{e}^{C_i}$  and  $\mathbf{g}^{C_i}$  by passing the concatenations  $[\mathbf{e}, \mathbf{c}^{(i)}]$  and  $[\mathbf{g}, \mathbf{c}^{(i)}]$  through their respective MLP.

3) Obtain the unique embedding representation influenced by all contextual factors. Different context has different influence weight on members/events/groups. How to measure the weight accurately is a key problem. Inspired by the neural attention mechanism [20], [21] which can learn the importance of different components in the model from data, we consider using attention to learn the weights of various contextual factors.

In order to obtain the user's unique embedding representation under the influence of all contextual factors, we aggregate the user's embedding representations influenced by each context. To this end, it is necessary to calculate the influence weight of each context on the user. Different from the existing method [12], the influence weight of each context in our model not only considers the interaction between the user and the context, but also considers the event that the user is currently concerned about. Our idea is that we measure how much the user  $u$ 's contextual representation  $\mathbf{u}^{C_i}$  matches the event  $e$ 's contextual representation  $\mathbf{e}^{C_i}$ . The more they match, the more the user prefers the context  $\mathcal{C}_i$  and accordingly  $\mathcal{C}_i$  would be given more weight. Specifically, in order to calculate the attention score  $\eta_{u,e}^{C_i}$  between context  $\mathcal{C}_i$  and user  $u$  when



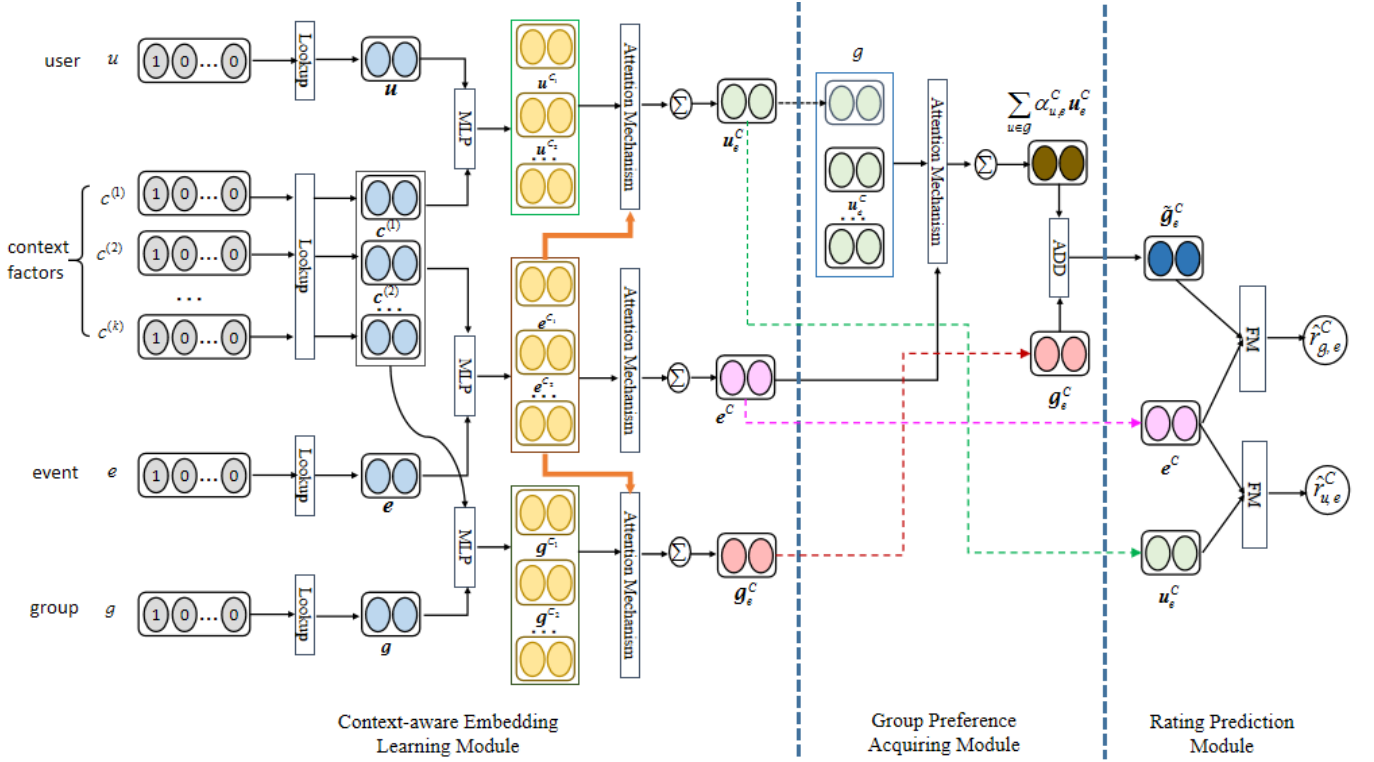


Fig. 2: The framework of ACGER.

$u$  is faced with event  $e$ , we design a neural attention network as follows:

$$\eta_{u,e}^{C_i} = \mathbf{h}^T \text{ReLU}(\mathbf{W}_1^\xi \mathbf{u}^{C_i} + \mathbf{W}_2^\xi \mathbf{e}^{C_i} + \mathbf{b}^\xi), \quad (3)$$

where  $\mathbf{W}_1^\xi \in \mathbb{R}^{d \times d}$  and  $\mathbf{W}_2^\xi \in \mathbb{R}^{d \times d}$  are weight matrices of the attention network,  $\mathbf{b}^\xi \in \mathbb{R}^d$  is the bias vector,  $\mathbf{h} \in \mathbb{R}^d$  is a weight vector which projects the output of the ReLU activation function to a score value. The superscript  $\psi$  indicates the marked model parameters are related to Eq. (3).

We normalize the value of  $\eta_{u,e}^{C_i}$  with a softmax function, and obtain the influence weight of context  $C_i$  on user  $u$  when he/she is faced with  $e$ .

$$\beta_{u,e}^{C_i} = \text{softmax}(\eta_{u,e}^{C_i}) = \frac{\exp \eta_{u,e}^{C_i}}{\sum_{i=1}^k \exp \eta_{u,e}^{C_i}}. \quad (4)$$

Finally we get  $u$ 's enhanced embedding influenced by all contextual factors, which is calculated as follows:

$$\mathbf{u}_e^C = \sum_{i=1}^k \beta_{u,e}^{C_i} \mathbf{u}^{C_i}. \quad (5)$$

Similarly, the attention score  $\mu_{g,e}^{C_i}$  and the influence weight  $\gamma_{g,e}^{C_i}$  of context  $C_i$  on group  $g$  when  $g$  is faced with event  $e$  are calculated respectively as follows:

$$\mu_{g,e}^{C_i} = \mathbf{t}^T \text{ReLU}(\mathbf{W}_1^\zeta \mathbf{g}^{C_i} + \mathbf{W}_2^\zeta \mathbf{e}^{C_i} + \mathbf{b}^\zeta) \quad (6)$$

$$\gamma_{g,e}^{C_i} = \text{softmax}(\mu_{g,e}^{C_i}) = \frac{\exp \mu_{g,e}^{C_i}}{\sum_{i=1}^k \exp \mu_{g,e}^{C_i}}, \quad (7)$$

where  $\mathbf{W}_1^\zeta \in \mathbb{R}^{d \times d}$ ,  $\mathbf{W}_2^\zeta \in \mathbb{R}^{d \times d}$ ,  $\mathbf{b}^\zeta \in \mathbb{R}^d$ ,  $\mathbf{t} \in \mathbb{R}^d$  are model parameters. The superscript  $\zeta$  indicates the marked model parameters are related to Eq. (6).

Then, we get group  $g$ 's enhanced embedding influenced by all contextual factors, which is calculated as follows:

$$\mathbf{g}_e^C = \sum_{i=1}^k \gamma_{g,e}^{C_i} \mathbf{g}^{C_i}. \quad (8)$$

where  $\mathbf{g}_e^C$  encodes the group  $g$ 's direct preference.

At last, the attention score  $\pi_e^{C_i}$  and the influence weight  $\rho_e^{C_i}$  of context  $C_i$  on event  $e$  are calculated as follows:

$$\pi_e^{C_i} = \mathbf{s}^T \text{ReLU}(\mathbf{W}_1^\varepsilon \mathbf{e} + \mathbf{W}_2^\varepsilon \mathbf{e}^{C_i} + \mathbf{b}^\varepsilon), \quad (9)$$

$$\rho_e^{C_i} = \text{softmax}(\pi_e^{C_i}) = \frac{\exp \pi_e^{C_i}}{\sum_{i=1}^k \exp \pi_e^{C_i}}, \quad (10)$$

where  $\mathbf{W}_1^\varepsilon \in \mathbb{R}^{d \times d}$ ,  $\mathbf{W}_2^\varepsilon \in \mathbb{R}^{d \times d}$ ,  $\mathbf{b}^\varepsilon \in \mathbb{R}^d$ , and  $\mathbf{s} \in \mathbb{R}^d$  are model parameters. The superscript  $\varepsilon$  indicates the marked model parameters are related to Eq. (9).

The event  $e$ 's enhanced embedding influenced by all contextual factors is calculated as follows:

$$\mathbf{e}^C = \sum_{i=1}^k \rho_e^{C_i} \mathbf{e}^{C_i}. \quad (11)$$

### B. Group Preference Acquiring Module

The goal of this section is to obtain an embedding vector for each group. In this paper, group preference is defined as the combination of indirect preference and direct preference. The

group's direct preference is encoded in the group's enhanced embedding obtained by Eq. (8), and the group's indirect preference is obtained by aggregating the members' embeddings, in which the key problem is how to measure the influence weights of group members. We use the neural attention mechanism to learn them from data. Next, we elaborate on the process of obtaining group indirect preference.

Suppose group  $g$  is making a decision on event  $e$ ,  $\alpha_{u,e}^C$  denotes the influence weight of member  $u$  on the group  $g$ 's decision in the contexts  $\mathcal{C}$ ,  $\mathbf{u}_e^C$  denotes  $u$ 's enhanced embedding influenced by all contextual factors, and embedding  $\mathbf{e}^C$  denotes the property of event  $e$  influenced by all contextual factors. Then  $\alpha_{u,e}^C$  is defined as the output of a neural attention network with embeddings  $\mathbf{u}_e^C$  and  $\mathbf{e}^C$  as the inputs:

$$\delta_{u,e}^C = (\mathbf{p})^T \text{ReLU}(\mathbf{W}_1^T \mathbf{u}_e^C + \mathbf{W}_2^T \mathbf{e}^C + \mathbf{b}^T), \quad (12)$$

$$\alpha_{u,e}^C = \text{softmax}(\delta_{u,e}^C) = \frac{\exp \delta_{u,e}^C}{\sum_{u' \in g_i} \exp \delta_{u',e}^C}, \quad (13)$$

where  $\delta_{u,e}^C$  is the attention score between member  $u$  and event  $e$ ,  $\mathbf{W}_1^T \in \mathbb{R}^{d \times d}$ ,  $\mathbf{W}_2^T \in \mathbb{R}^{d \times d}$  are the parameter matrices of the attention network,  $\mathbf{b}^T \in \mathbb{R}^d$  is the bias vector,  $\mathbf{p}$  is a parameter vector projecting the value of ReLU function into a score. The softmax function normalizes the score value to get the final weight  $\alpha_{u,e}^C$ . The superscript  $\tau$  indicates the marked model parameters are related to Eq. (12).

With the attention mechanism defined above, we can learn adaptively the aggregation strategy, which a group may change for different events, from interactions among contexts, groups and events. Next, we use the weights to aggregate the members' embeddings to form the group's indirect reference. In order to obtain the group's preference embedding, we combine the indirect preference with the direct preference by using an addition operation, which is utilized to combine different signals in the embedding space in work [21]). Specifically, when considering the event  $e$ , group  $g$ 's embedding, denoted as  $\tilde{\mathbf{g}}_e^C$ , is calculated as follows:

$$\tilde{\mathbf{g}}_e^C = \sum_{u \in \mathcal{U}_g} \alpha_{u,e}^C \mathbf{u}_e^C + \mathbf{g}_e^C, \quad (14)$$

where  $\sum_{u \in \mathcal{U}_g} \alpha_{u,e}^C \mathbf{u}_e^C$  and  $\mathbf{g}_e^C$  denotes the indirect preference and the direct preference of group  $g$  respectively,  $\mathcal{U}_g$  is the set of members of group  $g$ .

### C. Rating Prediction Module

In order to predict the group rating, we select FM [41]) model. This is because the interaction data in EBSNs is very sparse, and FM can model the high-order interaction between features more effectively than other methods on the sparse dataset [42]. Specifically, we feed the concatenation of group embedding and event embedding  $\mathbf{x} = [\tilde{\mathbf{g}}_e^C, \mathbf{e}^C]$  into FM, then the predicted rating of  $g$  on target event  $e$  is calculated as follows:

$$\hat{r}_{g,e}^C = w_0 + (\mathbf{w}_1^\phi)^T \mathbf{x} + \frac{1}{2} \sum_{k=1}^p [((\mathbf{v}_k^\phi)^T \mathbf{x})^2 - ((\mathbf{v}_k^\phi)^T \mathbf{x}^2)], \quad (15)$$

where  $w_0 \in \mathbb{R}$  is the global bias,  $\mathbf{w}_1^\phi \in \mathbb{R}^{2d}$  is the parameter vector,  $\mathbf{v}_k^\phi \in \mathbb{R}^{2d}$  is the  $k$ th column vector of parameter matrix  $\mathbf{V} \in \mathbb{R}^{2d \times p}$ , the hyper-parameter  $p \in \mathbb{N}_0^+$  denotes the dimension of factorized parameters. The superscript  $\phi$  indicates the marked model parameters are related to Eq. (15).

We rank the candidate events according to their predicted scores, and finally select the top- $N$  events to form the event list recommended for the group.

In addition to the group-event interaction data, there are also rich user-event interaction data in EBSNs. In order to reinforce the task of group event recommendation, we integrate the task of context-aware event recommendation for individuals into our model. Specifically, given a user-event interaction pair  $(u, e)$  and the current values of contextual factors, the one-hot feature vectors of user  $u$ , event  $e$ , and values of contextual factors are taken as the initial input data, and are passed through the embedding layer, MLP, and the attention network in turn to get the enhanced user embedding  $\mathbf{u}_e^C$  and the enhanced event embedding  $\mathbf{e}^C$ . Then, the concatenation of these two embeddings  $\mathbf{x} = [\mathbf{u}_e^C, \mathbf{e}^C]$  are fed into FM to get user  $u$ 's prediction score on target event  $e$ , denoted as  $\hat{r}_{u,e}^C$ . Since the two recommendation tasks share user embeddings, event embeddings and part of network weight parameters, the learning effect of group recommendation task is reinforced.

### D. Model Optimization

We treat the group event recommendation task as a ranking task, and select the commonly used pairwise learning method BPR (Bayesian Personalized Ranking) [43] to optimize the model parameters.

The pairwise learning method assumes that the observed interaction events should have a higher recommended ranking than the unobserved interaction events. The optimization objective function of the group recommendation task is as follows:

$$\sum_{(g,e,e') \in \mathcal{R}_{train}} -\log \sigma(\hat{r}_{g,e}^C - \hat{r}_{g,e'}^C) + \lambda_\Theta \|\Theta\|^2 \quad (16)$$

where  $\sigma(\cdot)$  denotes the logistic function,  $\Theta$  is the parameters to be learned in the neural network,  $\lambda_\Theta$  is the regularization hyper-parameter,  $\mathcal{R}_{train}$  is the training set in which  $(g, e, e')$  denotes that group  $g$  interacted with event  $e$  and did not interact with event  $e'$ .

Similarly, the objective function of individual recommendation task is as follows:

$$\sum_{(u,e,e') \in \mathcal{R}'_{train}} -\log \sigma(\hat{r}_{u,e}^C - \hat{r}_{u,e'}^C) + \lambda_{\Theta'} \|\Theta'\|^2, \quad (17)$$

where  $\Theta'$  is the set of parameters to be learned in the neural network,  $\mathcal{R}'_{train}$  is the training set in which  $(u, e, e')$  denotes that user  $u$  interacted with event  $e$  and did not interact with  $e'$ .

Stochastic gradient descent is used to minimize above objective functions. The optimization algorithm for group recommendation task is summarized in Algorithm 1. And the recommendation algorithm for ACGER model is presented in Algorithm. 2.

---

**Algorithm 1:** Optimization algorithm for group recommendation task in ACGER

---

**Input:**  $\mathcal{R}_{train}$ , learning rate  $lr$ , regularization hyper-parameter  $\lambda_\Theta$ , FM hyper-parameter  $p$ .

**Output:** updated model parameters  $\Theta$ .

```

1 Initialize  $lr$  and model parameters  $\Theta$ ;
2 repeat
3   Draw  $(g, e, e')$  from  $\mathcal{R}_{train}$ ;
4   Compute  $\mathbf{g}, \mathbf{e}, \mathbf{e}', \mathbf{c}^{(i)}$  ( $i \in \{1, \dots, k\}$ ),  $\mathbf{u}$  ( $\forall u \in g$ )
      by equations similar to Eq. (1) //obtain the
      low-dimensional embeddings of entities;
5   Compute  $\mathbf{g}^{C_i}, \mathbf{e}^{C_i}, \mathbf{e}'^{C_i}, \mathbf{u}^{C_i}$ ,  $i \in \{1, \dots, k\}$  by
      equations similar to Eq. (2)//obtain the embeddings
      under the influence of each context factor;
6   Compute  $\mathbf{u}_e^C$ ,  $\forall u \in g$ , by Eq. (3)-(5);
7   Compute  $\mathbf{g}_e^C$  by Eq. (6)-(8);
8   Compute  $\mathbf{e}^C$  by Eq. (9)-(11);
9   Compute  $\mathbf{g}$ 's embedding  $\tilde{\mathbf{g}}_e^C$  by Eq. (12)-(14);
10  Compute  $\hat{\mathbf{r}}_{g,e,e'}^C = \hat{\mathbf{r}}_{g,e}^C - \hat{\mathbf{r}}_{g,e'}^C$  by Eq. (15);
11  for each parameter  $\theta$  in  $\Theta$  do
12     $\theta \leftarrow \theta + lr \cdot \left( \frac{e^{-\hat{\mathbf{r}}_{g,e,e'}^C}}{1+e^{-\hat{\mathbf{r}}_{g,e,e'}^C}} \cdot \frac{\partial \hat{\mathbf{r}}_{g,e,e'}^C}{\partial \theta} + \lambda_\theta \cdot \theta \right)$ ;
13  end
14 until convergence;
15 return  $\Theta$ .
```

---



---

**Algorithm 2:** Recommendation Algorithm for ACGER

---

**Input:**  $\mathcal{U}, \mathcal{E}, \mathcal{C}, \mathcal{L}, \mathcal{G}$ , group-event interaction matrix  $Y$ , user-event interaction matrix  $R$ , target group  $g \in \mathcal{G}$ , candidate event set  $\mathcal{E}_{cand}$ , and given contextual values  $(c^{(1)}, c^{(2)}, \dots, c^{(k)})$ .

**Output:** the recommended event list  $L_g$  for group  $g$ .

```

1 build model ACGER;
2 initialize model parameters;
3 repeat
4   model.training( $\mathcal{R}_{train}$ ) based on Eq. (1) - (5),
      (9)-(11), (15),(17) //utilize user-event interactions;
5   model.training( $\mathcal{R}_{train}$ ) as Algorithm 1 //utilize
      group-event interactions;
6   model.evaluate( $\mathcal{R}_{test}$ ) //evaluate the performance of
      individual recommendation;
7   model.evaluate( $\mathcal{R}_{test}$ ) //evaluate the performance of
      group recommendation;
8 until convergence;
9 for  $e$  in  $\mathcal{E}_{cand}$  do
10   $\hat{\mathbf{r}}_{g,e}^C = \text{model.predict}(g, e, (c^{(1)}, c^{(2)}, \dots, c^{(k)}))$  based on
      Eq. (1) - (15);
11 end
12  $L_g = \text{topN}(\mathcal{E}_{cand})$  //select  $N$  events with the greatest
      predict scores of  $\hat{\mathbf{r}}_{g,e}^C$  given contexts  $\mathcal{C}$ ;
13 return  $L_g$ .
```

---

## V. PERFORMANCE ANALYSIS

In this section, we conducted extensive experiments on real datasets to answer the following research questions:

- (1) How does our proposed model ACGER perform compared with the state-of-the-art group recommendation model?
- (2) How is the effectiveness of our designed attention network for learning the contextual influence weight?
- (3) How is the effectiveness of our designed attention network for learning the group aggregation strategy?
- (4) How do the three components of the model — attentive context-aware embedding learning, group embedding learning, and individual recommendation task contribute to the performance of ACGER?

### A. Experimental Settings

1) *Datasets:* The datasets of this paper come from Meetup.com<sup>1</sup>, a popular EBSN platform. Through this platform, users can create events online, reply on whether to participate in events, join in various online social groups, and participate in events offline. We use the API interface provided by Meetup to obtain the relevant experimental data in 12 months of 2016. We choose to recommend events within the city scope, and choose New York and San Diego city in USA for experiments since they have the largest number of events published in 2016. We generate groups consisted of 2 to 6 users who often participate in events together, and collect events participated by each group and collect events attended by each user. In order to eliminate noise data and ensure the reliability of experimental results, the selected users and groups are required to participate in at least 10 events. In consideration of the universality of groups with the size of 2 users, the selected events are required to have at least two participants. After the data pre-processing, the statistics of the two city datasets are shown in Table I, where #U-E denotes the number of user-event interactions and #G-E denotes the number of group-event interactions

TABLE I: Statistics of the Meetup datasets.

City	#Users	#Events	#Groups	#U-E	#G-E
New York	2,849	10,024	2,727	288,447	101,141
San Diego	2,419	10,685	1,992	287,469	70,239

Four contextual factors were considered in our experiment: organizer, venue, time, and event content. The values of organizer and venue can be converted into one-hot vectors according to their integer ID. And since the value of time is continuous and the value of content is textual, we need to perform data pre-processing on these two factors to get their initial vector representations.

For the event content denoted as  $\mathcal{C}_{cont}$ , we regard each content text as a document, and all event content documents constitute a corpus. We use the natural language processing technology CBOW (Continuous Bag-of-Words) [44] to map

<sup>1</sup><http://www.meetup.com>



each word in the corpus into a low-dimensional word vector. The vector of the content of an event  $e$  is calculated as follows:

$$\mathbf{l}_e^{\text{cont}} = \frac{\sum_{w \in \mathcal{W}_e} \mathbf{z}_w}{|\mathcal{W}_e|}, \quad (18)$$

where  $\mathcal{W}_e$  denotes the set of words in the content text of event  $e$ ,  $\mathbf{z}_w \in \mathbb{R}^d$  denotes the vector of word  $w$ . The content embeddings of all events are combined to form a pre-trained embedding matrix, which is used to initialize the content embedding matrix in ACGER.

The context of time represents the start time of an event. In order to map a continuous timestamp into a discrete time slot, we adopt a weekday-hour pattern, such as “2 (day of the week), 16:00-17:00 (hour of the day)”. Then, we get at most  $7 \times 24$  discrete time slots for the context of time. Next, the time of each event is mapped to a  $7 \times 24$ -dimensional one-hot vector according to its time slot.

2) *Evaluation Metrics*: For each dataset, we rank the group-event/user-event interactions according to the start time of events. Then, we take the first 80% of ranked interactions as the training set, 10% as the validation set, and the last 10% as the test set. Validation set is used for tuning the hyper-parameters. In the test set, the interacted events of each group/user are regarded as the real interested events to evaluate the recommendation performance of algorithms. Since we use the pairwise loss as our objective function, positive samples are selected from the events that the group/user has interacted with, and negative samples are selected from the events that the group/user has not interacted with.

In order to evaluate the performance of top- $N$  recommendation methods, we adopt three widely used evaluation metrics: precision (P@N), recall (R@N), and NDCG (Normalized Discounted Cumulative Gain, NDCG@N) [36]. Among them, NDCG measures the ability of a method to rank the events of truly interest higher in the recommendation list. For each metric, the higher the value, the better the recommendation performance.

3) *Baselines*: To justify the effectiveness of our method, we compared it with the following methods:

(1) GERF [11]: This is a method of context-aware event recommendation for groups. In this method, context influences are defined manually, the group feature vector is obtained by concatenating the feature vectors of group members, and a simple linear model is adopted for predicting the group’s rating for an event. This method does not differentiate the influences of different contextual factors and treats group members as equally important.

(2) UL [8]: This is a traditional context-unaware group recommendation method. The group aggregation strategy in this method is manually predefined, which combines the deviation of group members’ opinions with average and approval voting strategies.

(3) AIN\_ACGER2 [12]: This method uses AIN (Attention Interaction Network, [12]) to model the influence of contextual factors on users/groups and events. The difference between this method and ACGER is that this method doesn’t consider the impact of events when calculating the influence weight of contextual factors on groups/users.

(4) ACGER1\_LinerBpr: This is a variant of ACGER, which firstly obtains the embedding of users, events and groups under the influence of all contextual factors by using the same attention networks as that in ACGER. Then, the members’ embeddings are concatenated to form the group embedding and a linear model is employed to predict the ratings (just the same as GERF). This method is used to compare with GERF. The difference between this method and GERF is that this method uses attention networks to model contextual influences rather than defining influences manually.

4) *Experimental Settings*: We implemented the neural network-based methods such as AIN\_ACGER2, ACGER1\_LinerBpr and our ACGER in PyTorch. Other methods such as GERF and UL are implemented in Python. For the methods based on neural networks, Adam algorithm is used for optimization. The minimum batch size and learning rate are respectively in the range of [128, 256, 512, 1024] and [0.001, 0.005, 0.01, 0.05, 0.1]. For the embedded layer and hidden layers, we use the Gaussian distribution with mean value of 0 and standard deviation of 0.1 to initialize their parameters randomly. In neural attention network, the embedding dimensions of users, groups, events and contexts are all empirically set to 32. For each MLP, we deploy it with two hidden layers with dimensions set to 48 and 40 respectively. The factorization dimension in FM are set to 10. And the weight parameters of three elements in the calculation of predicted group rating in UL method are determined by grid search. We repeat 5 times for each setting to report the average results.

## B. Overall Performance Comparison (RQ1)

Fig. 3 shows the top- $N$  ( $N=5, 10$ ) recommendation performance of our ACGER and comparative methods on New York and San Diego datasets. We can see that ACGER achieves the best performance on both datasets with respect to three metrics. ACGER obtains improvements over the best baseline AIN\_ACGER2 by 2.8% in P@5, 2.1% in R@5 and 3.4% in NDCG@5 on New York dataset. On San Diego dataset, ACGER improves over AIN\_ACGER2 by 2.5% in P@5, 2.4% in R@5 and 3.2% in NDCG@5. This proves the validity of ACGER. Specifically, we can make the following observations: (1) context-aware methods (GERF, AIN\_ACGER2, ACGER1\_LinerBpr, and ACGER) have better performance than the context-unaware method (UL). This confirms the positive effect of context information on improving recommendation. (2) among the context-aware methods, the performance of neural network-based methods (AIN\_ACGER2, ACGER1\_LinerBpr, and ACGER) are better than that of method GERF which does not employ neural networks. This demonstrates the superiority of neural networks, especially their great ability in modeling the high-order interactions among different entities. (3) our model ACGER outperforms AIN\_ACGER2 on both datasets in three metrics. This is due to the fact that the influence of a contextual factor on a user/group may change when the type of an event concerned changes. Therefore, the performance can be further improved by taking into account the impacts of events when

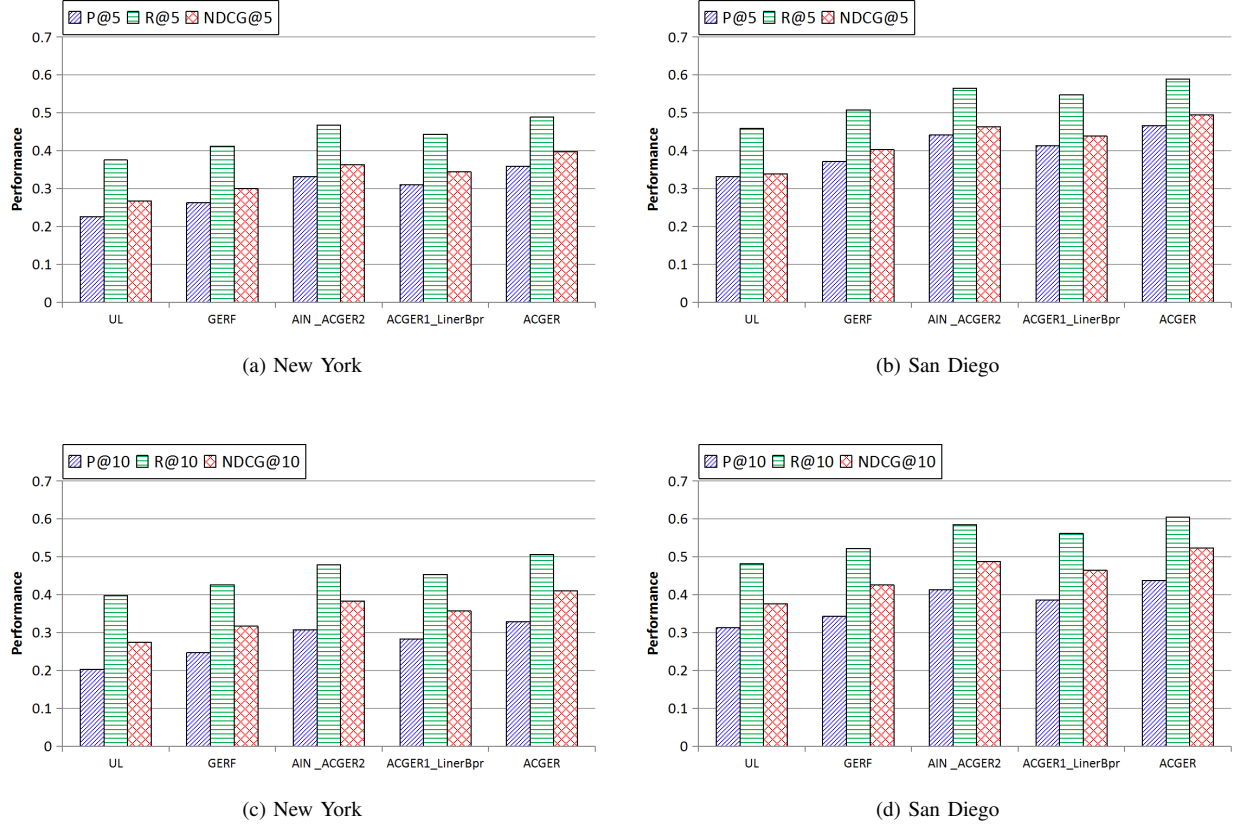


Fig. 3: Top- $N$  recommendation performance comparison between ACGER and baselines.

measuring the weight of contextual factors on users/groups. (4) The performance gap between ACGER1\_LinerBpr and ACGER shows the effectiveness of our model ACGER in modeling the group preference.

### C. Effect of Attention for Context Influence (RQ2)

To demonstrate the effectiveness of attention mechanism of ACGER in distinguishing the influences of different contextual factors, we compare ACGER with its following variants:

1) Avg\_ACGER2: In this method, the influence weights of different contextual factors on the entity (i.e., group, user, and event) are equal.

2) AIN\_ACGER2: In this method, the influence weight of a contextual factor on users/groups is calculated by AIN method, in which the influence on users/groups only considers the interaction between users/groups and context, without capturing the dynamic change of the influence weight of the contextual factor on users/groups when the type of events changes.

3) SingleU\_ACGER2: This method considers the influence of contextual factors on users and groups, neglecting their influence on events.

4) SingleE\_ACGER2: This method considers the influences of contextual factors on events, neglecting their influence on users and groups.

Fig. 4 shows the top- $N$  ( $N=5, 10$ ) recommendation performance comparison results of ACGER and its variants

on two datasets. We have the following observations: (1) The performances of Avg\_ACGER2 on both datasets are relatively low, slightly better than that of SingleE\_ACGER2, which performs the worst on both datasets. This is because that Avg\_ACGER2 does not distinguish the influence weights of different contextual factors. (2) the performance of AIN\_ACGER2 is not as good as that of ACGER, which indicates that modeling the dynamic change of influence weights of contextual factors on users/groups with the type of events helps to improve recommendation performance. (3) ACGER consistently outperforms SingleU\_ACGER2 and SingleE\_ACGER2 on both datasets with respect to three metrics. This may be due to the fact that contextual factors characterize the situation where users/groups interact with events, and thus have influences on users/groups and events at the same time. Therefore, considering the effects of contexts on both users/groups and events leads to better performance. (4) SingleU\_ACGER2 slightly outperforms SingleE\_ACGER2 on both datasets, which indicates that considering the contextual influences on users/groups is more effective than considering the contextual influences on events to improve recommendation performance. It shows that the interests of users/groups are more sensitive to the contextual factors than the properties of events.

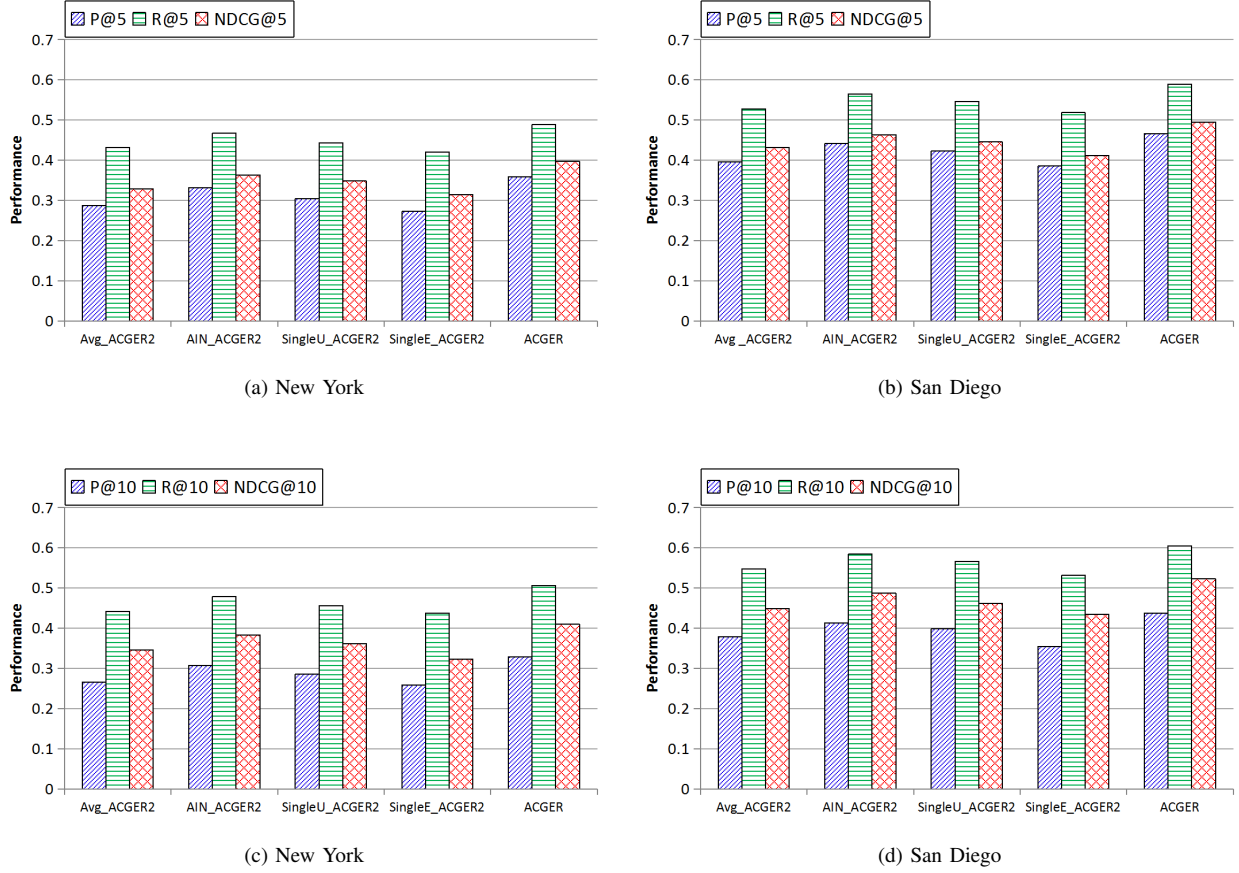


Fig. 4: Top- $N$  recommendation performance of ACGER and its variants in measuring the influence of contextual factors.

#### D. Effect of Attention for Group Aggregation (RQ3)

To demonstrate the effectiveness of the attention-based group aggregation strategy, we replace the strategy in ACGER with other predefined strategies, and obtain the following variants:

1) ACGER1\_Avg [15]: The group aggregation strategy in this method adopts the average strategy.

2) ACGER1\_BC [13]: The group aggregation strategy in this method adopts the “Borda Count (BC)” strategy, which scores ratings based on the ranking results. Specifically, for each group member, first rank the events according to their ratings, then score each event according to its ranking position (for example, the lowest ranking scores 0 and the highest ranking  $n - 1$  in the  $n$  events). Finally, scores of all members for each event are summed, and events are ranked according to the summed scores to get the recommendation list for the group.

3) ACGER1\_Exp [16]: The group aggregation strategy in this method adopts weighted sum method, where the weight of a member is decided by the number of events he/she attends. Generally speaking, the more events a user has participated in, the more expert he/she may be.

4) ACGER1\_MP [45]: The group aggregation strategy in this method adopts “Most Pleasure (MP)” strategy, which takes

the highest value in members’ ratings as the group’s rating for a candidate event.

The experimental results are shown in Fig. 5. As you can see, there is no predefined strategy that always wins. For example, when  $N = 5$  on New York dataset, ACGER1\_MP outperforms ACGER1\_BC (as shown in Fig. 5 (a)), but underperforms when  $N = 10$  (as shown in Fig. 5 (b)). The Similar situation occurs on San Diego dataset. When  $N = 5$ , ACGER1\_Exp outperforms ACGER1\_BC (as shown in Fig. 5 (c)), but underperforms when  $N = 10$  (as shown in Fig. 5 (d)). ACGER shows great flexibility and superiority because it can learn the group aggregation strategy from data.

#### E. Contribution Analysis of Components (RQ4)

To evaluate the contribution of the main components of ACGER to the group recommendation performance, we conducted some ablation experiments. We compare the ACGER with its following variants:

1) Avg\_ACGER2, ACGER1\_Avg: Avg\_ACGER2 is the ACGER taking the contextual factors as equally important, and ACGER1\_Avg is the ACGER adopting the average strategy as the group aggregation strategy. The aim of these two variants is to study the contribution of attention networks.

2) ACGER\_U, ACGER\_G: ACGER\_U is the ACGER with group indirect preference only, and ACGER\_G is the ACGER

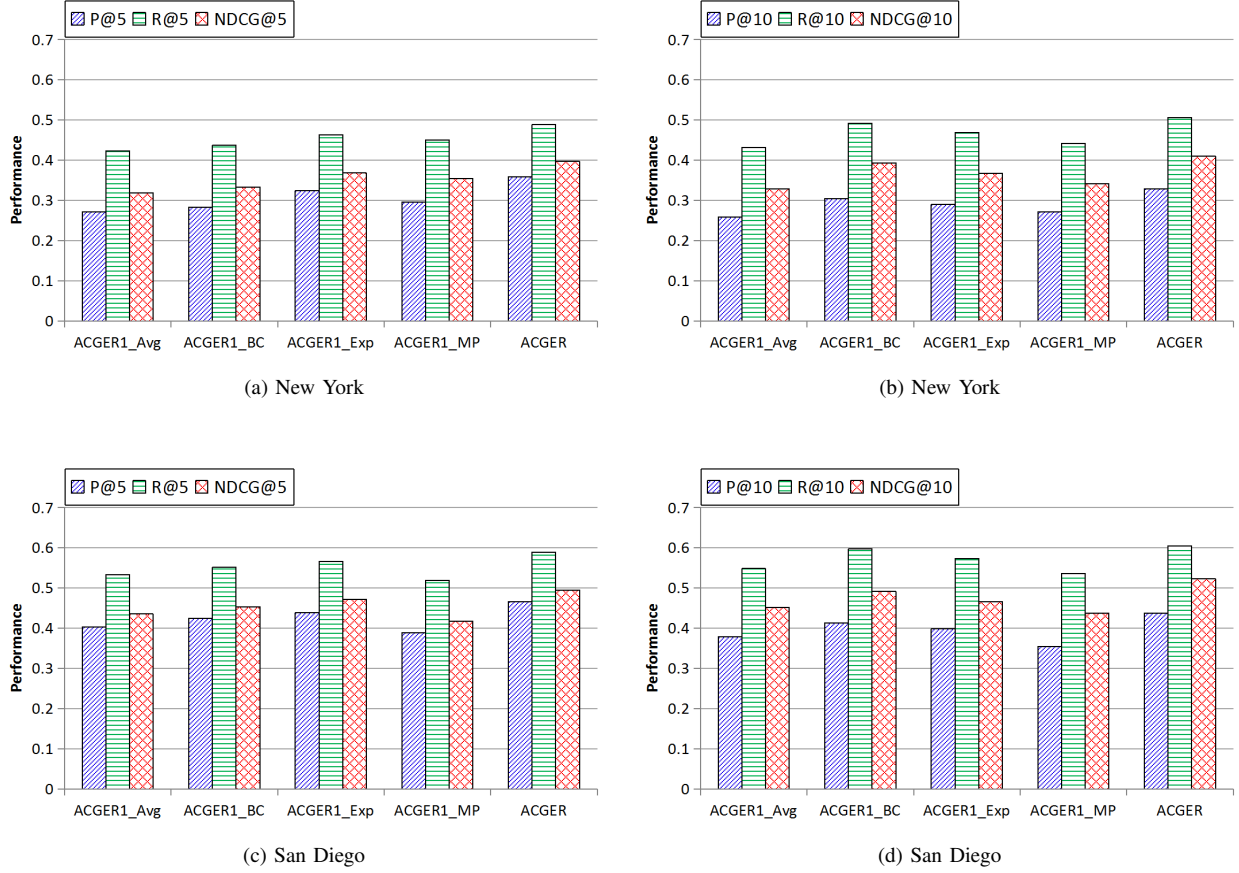


Fig. 5: Top- $N$  recommendation performance of ACGER and its variants in comparing group aggregation strategy.

with group direct preference only. Our purpose is to study the contribution of two different type of group preferences to the recommendation performance.

3) ACGER\_Grp: ACGER\_Grp is the ACGER without adding the individual recommendation task. The purpose of this method is to study the contribution of individual recommendation task to the group recommendation performance.

4) ACGER2: This method denotes the ACGER without considering the influences of contextual factors. Our purpose is to study the contribution of modeling contextual influences to the recommendation performance.

The experimental results are shown in Fig. 6. We have observations as follows: (1) Compared with ACGER, the performances of Avg\_ACGER2 and ACGER1\_Avg decrease on both datasets with respect to three metrics, which demonstrates that our attention-based weight calculation methods for contextual factors and group members are effective. (2) ACGER\_U and ACGER\_G underperform ACGER on two datasets. This indicates that on two datasets, group embedding is affected by both group indirect preference and group direct preference. The performance of ACGER\_U is superior to that of ACGER\_G, which reveals that the group indirect preference has a larger impact in learning group preference on two datasets. (3) the performance of ACGER\_Grp is inferior to that of ACGER on both datasets. For example, compared with ACGER, the

performance of ACGER\_Grp decreases by 10.03% in  $P@5$ , 7.58% in  $R@5$  and 10.08% in  $NDCG@5$  when  $N = 5$  on New York dataset, and the similar phenomenon could be observed on San Diego dataset. This indicates that the individual recommendation task could effectively reinforce the group recommendation task. (4) The performance of ACGER2 is significantly inferior to that of ACGER. This indicates that context information has a great impact on the performance of our model, because it can greatly improve the accuracy of group preference learning.

#### F. Industrial Applications

In the market environment, participating in product exhibitions is one of the effective marketing methods for enterprises. Through the exhibition, the enterprise can show its products, enterprise strength, and brand image to the industry peers and live audiences, and it can quickly grasp the status and trend of domestic and international industries and get the information on the new products. By participating the exhibition, the enterprise could find potential customers and new cooperative partners at a lower cost than general marketing channels. Therefore, enterprises have the inherent needs for participating in product exhibitions. An EBSN application provides a platform for organizers to release exhibition information and for enterprise users to sign up for the exhibitions. However, with

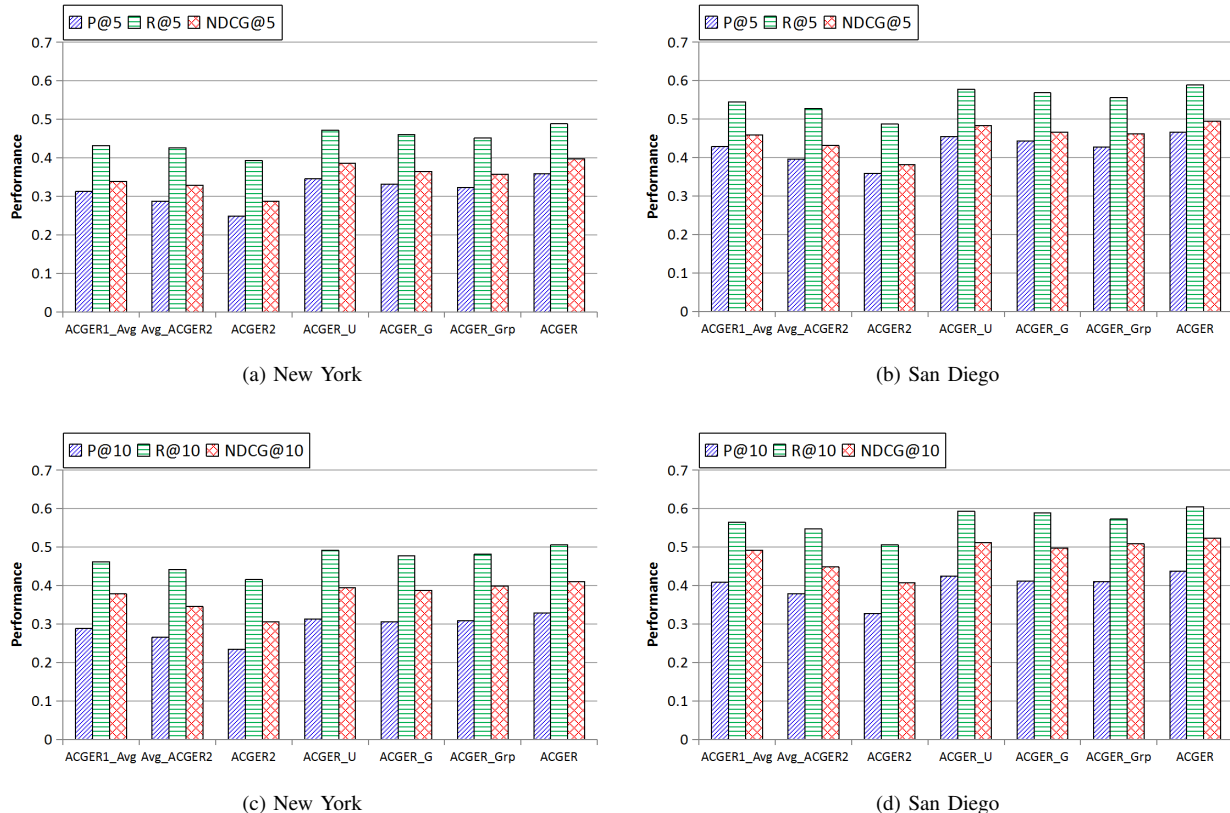


Fig. 6: Contribution analysis of different components of ACGER.

the popularity of the platform, more and more information is published on the platform, and it becomes increasingly difficult for enterprise users, especially enterprise groups who often attend exhibitions together, to find their interested exhibitions. Therefore, it is urgent for the platform to provide exhibition recommendation service (i.e., event recommendation service) to help them find their interested exhibitions efficiently.

In addition to exhibitions, other types of events, such as technical seminars and professional summit forums, are also events of interest to enterprises. An EBSN event recommender system can also meet such needs.

## VI. CONCLUSION AND FUTURE WORK

In this paper, we study an Attention-based Context-aware Group Event Recommendation model (ACGER) for EBSNs. ACGER employs the neural networks with attention mechanism to model the complex and highly non-linear impacts of contexts on users, groups, and events. In order to model the situation where the influence weight of a contextual factor on users/groups may change with the type of events concerned, ACGER leverages a novel attention network, which not only considers the interaction between users/groups and context, but also considers the impacts of events. To overcome the limitation of lacking flexibility of the predefined group aggregation strategy in existing group recommendation methods, ACGER uses the neural attention mechanism to learn an adaptive group aggregation strategy from the data. Such mechanism

enables a group automatically adjust its decision strategy according to the currently concerned events. Considering that a group often have different behavior patterns from its members, ACGER not only considers the indirect preferences aggregated from members' preferences, but also considers the direct preferences specific to group itself, which makes the group preferences captured more accurate. In addition, in order to make full use of user-event interaction data, we integrate the individual recommendation task into ACGER to reinforce the group recommendation task. Extensive experiments on two real datasets show that ACGER achieves higher recommendation performance than state-of-the-art methods.

In the future, as event sequence could be regarded as a special contextual factor, how to characterize the influence of event sequence on group recommendation could be further studied. We also try to explore the influence of trust relationship on user preference by using trust network embedding technology. In addition, an EBSN is a special heterogeneous network combining online and offline networks. Inspired by the recent development of graph neural network, how to employ this technique to model EBSN network structure for better recommendation is another interesting work.

## REFERENCES

- [1] W. Zhang and J. Wang, "A collective bayesian poisson factorization model for cold-start local event recommendation," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1455–1464.



- [2] H.-C. Wang, H.-T. Jhou, and Y.-S. Tsai, "Adapting topic map and social influence to the personalized hybrid recommender system," *Information Sciences*, vol. 60, no. 11–14, pp. 1684–1706, 2018.
- [3] G. Liao and H. Zhiwei, "A global recommendation strategy considering constraints in event-based social networks," *Journal of Nanjing University, Natural Science*, vol. 54, no. 1, pp. 11–12, 2018.
- [4] Z. Qiao, P. Zhang, C. Zhou, Y. Cao, L. Guo, and Y. Zhang, "Event recommendation in event-based social networks," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014, pp. 3130–3131.
- [5] L. Boratto and S. Carta, "Art: group recommendation approaches for automatically detected groups," *International Journal of Machine Learning & Cybernetics*, vol. 6, no. 6, pp. 953–980, 2015.
- [6] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, 2nd ed. Boston, MA, USA: Springer, 2015.
- [7] W. Wang, G. Zhang, and J. Lu, "Member contribution-based group recommender system," *Decision Support Systems*, vol. 87, pp. 80–93, 2016.
- [8] Y.-D. Seo, Y.-G. Kim, E. Lee, K.-S. Seol, and D.-K. Baik, "An enhanced aggregation method considering deviations for a group recommendation," *Expert Syst. Appl.*, vol. 93, pp. 299–312, 2018.
- [9] A. Q. Macedo, L. B. Marinho, and R. L. T. Santos, "Context-aware event recommendation in event-based social networks," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 123–130.
- [10] M. Xu and S. Liu, "Semantic-enhanced and context-aware hybrid collaborative filtering for event recommendation in event-based social networks," *IEEE Access*, vol. 7, pp. 17 493–17 502, 2019.
- [11] Y. Du, X. Meng, Y. Zhang, and P. Lv, "Gerf: A group event recommendation framework based on learning-to-rank," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 4, pp. 674–687, 2020.
- [12] L. Mei, P. Ren, Z. Chen, L. Nie, J. Ma, and J. Nie, "An attentive interaction network for context-aware recommendations," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 157–166.
- [13] J. Masthoff, "Group modeling: Selecting a sequence of television items to suit a group of viewers," *User Modeling and User-Adapted Interaction*, vol. 14, no. 1, pp. 37–85, 2004.
- [14] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso, "Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices," *Applied Artificial Intelligence*, vol. 17, no. 8, pp. 687–714, 2003.
- [15] S. Berkovsky and J. Freyne, "Group-based recipe recommendations: Analysis of data aggregation strategies," in *Proceedings of the 4th ACM Conference on Recommender Systems*, 2010, pp. 111–118.
- [16] E. Quintarelli, E. Rabosio, and L. Tanca, "Recommending new items to ephemeral groups using contextual user influence," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 285–292.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
- [18] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.
- [19] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translates," in *3rd International Conference on Learning Representations*, 2015.
- [21] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 3119–3125.
- [22] A. K. Dey, "Understanding and using context," *Personal Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, Jan. 2001.
- [23] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems*, vol. 23, no. 1, 2005.
- [24] L. Baltrunas and X. Amatriain, "Towards time-dependant recommendation based on implicit feedback," in *Proceedings of the 3rd ACM Conference on Recommender Systems In Workshop on context-aware recommender systems*, 2009, pp. 423–424.
- [25] L. Baltrunas and F. Ricci, "Experimental evaluation of context-dependent collaborative filtering using item splitting," *User Modeling & User Adapted Interaction*, vol. 24, no. 1–2, pp. 7–34, 2014.
- [26] Y. Zheng, R. Burke, and B. Mobasher, "Splitting approaches for context-aware recommendation: An empirical study," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, 2014, pp. 274–279.
- [27] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, and A. Pedone, "Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems," in *Proceedings of the 3rd ACM Conference on Recommender Systems*, 2009, pp. 265–268.
- [28] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proceedings of the 5th ACM Conference on Recommender Systems*, 2011, pp. 301–304.
- [29] Y. Zheng, B. Mobasher, and R. Burke, "Cslim: Contextual slim recommendation algorithms," in *Proceedings of the 8th ACM Conference on Recommender Systems*, 2014, pp. 301–304.
- [30] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the 4th ACM Conference on Recommender Systems*, 2010, pp. 79–86.
- [31] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver, "Tfmap: Optimizing map for top-n context-aware recommendation," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012, pp. 155–164.
- [32] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011, pp. 635–644.
- [33] J. Cao, Z. Zhu, L. Shi, B. Liu, and Z. Ma, "Multi-feature based event recommendation in event-based social network," *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 618–633, 2018.
- [34] Y. Wang and J. Tang, "Event2vec: Learning event representations using spatial-temporal information for recommendation," in *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2019, pp. 314–326.
- [35] Z. Wang, Y. Zhang, H. Chen, Z. Li, and F. Xia, "Deep user modeling for content-based event recommendation in event-based social networks," in *Proceedings of 2018 IEEE Conference on Computer Communications*, 2018, pp. 1304–1312.
- [36] Q. Yuan, G. Cong, and C.-Y. Lin, "Com: a generative model for group recommendation," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 163–172.
- [37] S. Purushotham and C.-C. J. Kuo, "Personalized group recommender systems for location- and event-based social networks," *ACM Transactions on Spatial Algorithms and Systems*, vol. 2, no. 4, pp. 16:1–16:29, 2016.
- [38] K. Ji, Z. Chen, R. Sun, K. Ma, Z. Yuan, and G. Xu, "Gist: A generative model with individual and subgroup-based topics for group recommendation," *Expert Systems with Applications*, vol. 94, pp. 81–93, 2018.
- [39] Y. Du, X. Meng, and Y. Zhang, "Cvtm: a content-venue-aware topic model for group event recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 7, pp. 1290–1303, 2019.
- [40] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 355–364.
- [41] S. Rendle, "Factorization machines," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, 2010, pp. 995–1000.
- [42] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for ctr prediction," in *Proc. 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 1725–1731.
- [43] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.
- [44] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, vol. 9, pp. 249–256, 2010.
- [45] L. Boratto, S. Carta, and G. Fenu, "Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modeling and clustering," *Future Generation Computer Systems*, vol. 64, pp. 165–174, 2016.