Knowledge-Based Systems 213 (2021) 106659

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

DGC: Dynamic group behavior modeling that utilizes context information for group recommendation

Hyun Ji Jeong, Kwang Hee Lee, Myoung Ho Kim [*]

Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

A R T I C L E   I N F O

Article history:
Received 15 May 2020
Received in revised form 7 November 2020
Accepted 3 December 2020
Available online 18 December 2020

A B S T R A C T

In the real world, group recommendation that recommends items to a set of users (i.e., a group) is a challenging problem since it is very difficult to ensure the satisfaction of all group members with different preferences. Many existing group recommendation systems commonly use aggregation methods, which are insufficient to model group behavior where preference of a user as an individual is often changed when he/she is a member of a group. Some recent methods attempt to reflect this dynamic group behavior. However, they still have limitations in capturing complex relationships between items and group members. Moreover, previous approaches do not fully utilize useful context information available, and only use rating data. Reflecting context information together with ratings helps resolve a well-known data sparsity problem in group recommendation. In this paper, we propose a novel group recommendation framework called Dynamic Group behavior modeling that utilizes Context information for group recommendation(DGC). In DGC, we newly develop dynamic group behavior modeling that enables summarization of complex patterns in group decision making processes. To apply context information, firstly, we extract relevant context information from a heterogeneous information network (HIN) that contains rich information between various entities. Then, context information is properly applied to a group recommendation model by using semi-supervised learning that is composed of a supervised loss for label prediction and an unsupervised loss for context prediction. Experimental results show that our method provides significant performance improvement over other existing methods.

1. Introduction

Group recommendation recommends items to a group, (i.e., a set of users) where each member of the group has his/her own preference (i.e., rating). Rating data of items in group recommendation denotes scores of items that a group provides based on its preferences, which is represented by group–item interaction. There are many applications for group recommendation in our real life such as recommending restaurants to families, recommending movies for watching with friends, etc. Group recommendation is a challenging problem because it is difficult to satisfy all members of a group, each of whom may have different preferences [1]. Previous researches mainly focused on how to aggregate individual preferences of group members [2,3]. Early methods use predefined functions for aggregation such as average, least misery, and maximum satisfaction, etc. Recently, some neural network based group recommendation methods have been proposed, which employ attentive aggregation in which an attention is automatically learned [4]. These aggregation based methods that focus on aggregation of group users' preferences, however, have not properly been discussed.

We often observe that the preferences of a user as a group member may not be the same as preference when the use acts individually. For example, people who like to watch action movies may prefer movies of different genres when they are with their family. Further, each group member has a distinct influence on group decisions. The behavior of group g is determined by the fusion of decisions of members in g. The characteristics of each group member such as their preference and influence on other group members is often dynamic and it depends on what members are in g. Thus, we can notice that the behavior of a group is not an aggregation of static preference scores and influence scores of all members in the group in a simple way; however, it is consolidated result of dynamic characteristics of group members, which is uniquely determined by what members are in the group. We call this phenomenon dynamic group behavior.

Aggregation-based approaches have a problem to accommodate this concept of dynamic group behavior. Let us consider a

[*] Correspondence to: School of Computing, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea.
E-mail addresses: hjjung@dbserver.kaist.ac.kr (H.J. Jeong), kwanghee@dbserver.kaist.ac.kr (K.H. Lee), mhkim@dbserver.kaist.ac.kr (M.H. Kim).

https://doi.org/10.1016/j.knosys.2020.106659
0950-7051/© 2020 Elsevier B.V. All rights reserved.

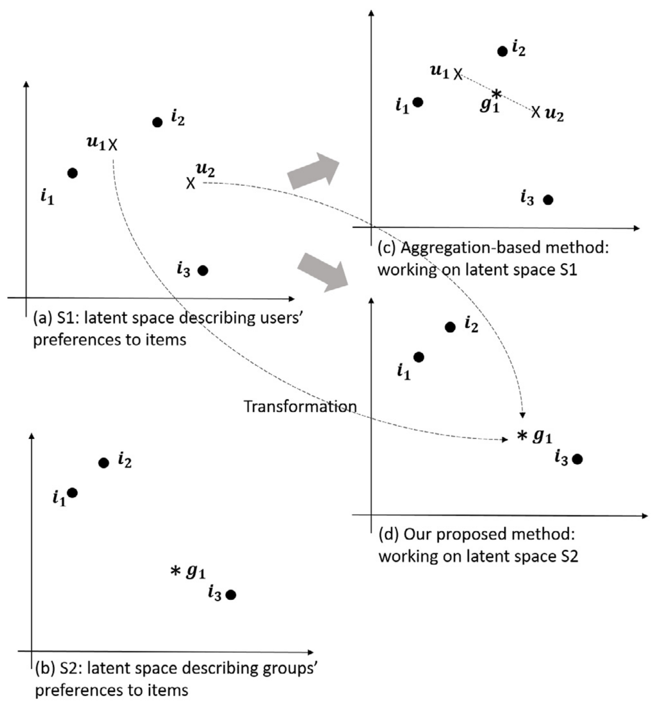

(a) S1: latent space describing users' preferences to items
(b) S2: latent space describing groups' preferences to items
(c) Aggregation-based method: working on latent space S1
(d) Our proposed method: working on latent space S2
Transformation
u_1
u_2
i_1
i_2
i_3
g_1

Fig. 1. Toy examples of dynamic group behavior.

set of users (u1, u2) and a set of items (i1, i2, i3), and let group g1 consist of two users u1 and u2. Suppose that latent space S1 describes users' preferences to items. (We use two dimensional latent space for convenience, however, the latent space may be usually higher than 2). Fig. 1(a) shows latent vectors u_1 and u_2 of two users, and latent vectors i_1, i_2, and i_3 of three items. Then, for a given user u1, we may recommend an item based on the proximity of the items' latent vectors to user u1 in latent space S1. Here, both users u1 and u2 have higher preferences for items i1 and i2 than i3. Now, consider latent space S2 that describes the preferences of the groups to items. Fig. 1(b) show latent vectors i_1, i_2, and i_3 for three items i1, i2, and i3. A latent vector g_1 describes group g1 in latent space S2. We will use a user–item space and a group–item space to denote S1 and S2 for convenience, respectively.

In Fig. 1(b), g1 has a higher preference for i3 than i1 and i2 even though both members of g1, i.e., u1 and u2 do not. Further, we also infer u2 may have a higher effect than u1 on the decision of g1 because u2 prefers i3 more than u1. Such a situation can commonly occur in practice because of what we call "dynamic group behavior". That is, dynamic group behavior occurs when an individual may behave different as being a member of a group. Under this scenario, suppose we want to recommend a Top1 item to g1. If we use an existing aggregation-based approach (e.g., a weighted average aggregation) as shown in Fig. 1(c), the latent vector g_1 of g1 in the user–item space becomes

$$g_1 = w_1u_1 + w_2u_2 \tag{1}$$

where w_i indicates a weight of user i's latent vector u_i. g_1 is positioned only on the dotted line $w_1u_1 + w_2u_2$. That is, the range of g_1 is restricted by the static preferences of users. In Fig. 1(c), the Top1 preference of g1 is i2, although g1 actually prefers i3 the most, as shown in Fig. 1(b). Some recent works attempted to resolve this problem. However, these methods only allow for the translation of the dotted line. They do not support rotation and scaling, which result in a limitation to reflect dynamic group behavior properly.

Context information is another important factor in group recommendation. In a recommendation task for a single user, context information is defined as relationships that connect users or items with other objects, except for the information to be predicted (i.e., a rating of a user) [5]. By context information in group recommendation, we mean relationships of groups and items with other objects, except for the group–item information to be predicted (i.e., a rating of a group). For example, Fig. 2 shows various interactions in the Gowalla dataset where a set of users is a group and venues are items. In the figure, relationships between venues and spots/categories become context information of items (i.e., venues). Note that entities with the same context are more likely to be similar, compared to those that do not share the same context. Hence, context information describes intrinsic features of entities. Although it is known that context information helps to alleviate the data sparsity problem of recommender systems, most previous methods [2,3] for group recommendation have only used rating data. Thus, it is necessary to reflect context information in a group recommendation model appropriately.

In this paper, we propose dynamic group behavior modeling that utilizes context information for group recommendation (DGC). The group behaviors are modeled by using a neural network. As mentioned before, the preference of a group is a complicated mixture of the members' preferences. To consider dynamic group behavior, we propose a dynamic group collaborative filtering (DGCF). DGCF is composed of two parts: dynamic group behavior modeling and group collaborative filtering. Dynamic group behavior modeling models dynamic group behavior by using transformation of latent vectors describing the user–item interaction to latent vectors describing group–item interaction, as shown in Fig. 1(d). A set of latent vectors u_1 and u_2 in the user–item space are transformed to a those in the group–item space. Item latent vectors for i_1, i_2, and i_3 in the user–item space are also transformed to individual latent vectors i_1, i_2, and i_3 in the group–item space, respectively. Through the transformation, we can find i_3 as the Top1 item of g_1 as shown in Fig. 1(d). The transformation allows rotation and scaling for a group vector. In addition, the transformation matrix is automatically trained, which makes it possible to summarize complex patterns of group decision such as importance for item properties or effects of users. Multi-source group collaborative filtering predicts ratings based on group latent vectors and the three item latent vectors. Item latent vectors are generated from user–item interaction, group–item interaction, and item context information. These item latent vectors extracted from various data sources allow to more accurately capture features of items.

Next, to consider context information, we design a loss function based on semi-supervised learning that combines an unsupervised loss for unlabeled data with a supervised loss for label prediction. The semi-supervised learning has been used to improve performance of label prediction using unlabeled data. In our method, a loss function is defined as the combination of a supervised loss for recommendation and an unsupervised loss for context prediction. For predicting context, we design a Item Context Prediction (ICP) that predicts item context vectors generated by the graph embedding method Esim [6]. Graph embedding helps extract feature vectors without depending on a data model of input data.

- We propose a novel method called Dynamic Group Collaborative Filtering (DGCF) that models dynamic group behavior by using transformation from user–item space to group–item space. Further, DGCF performs collaborative filtering based on group latent vectors and item latent vectors extracted from various sources.

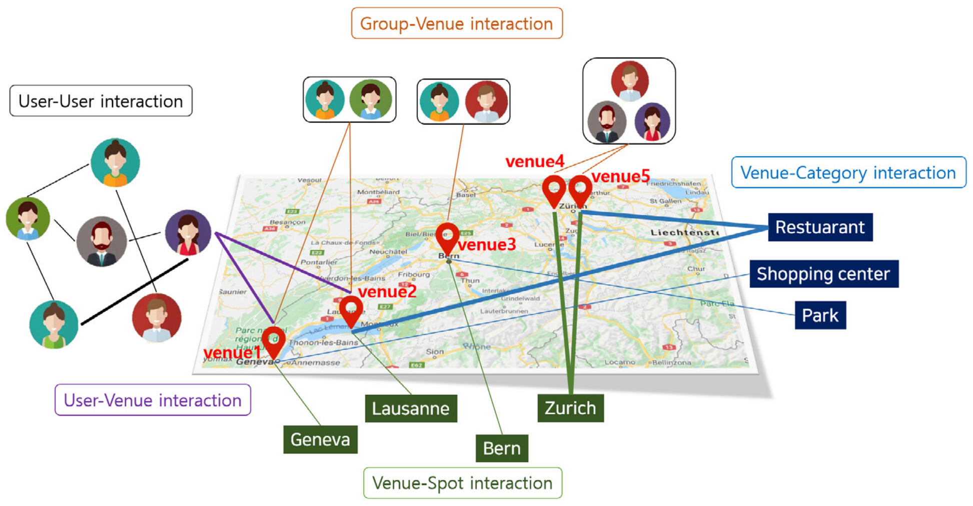


Fig. 2. Toy examples of context information in Gowalla dataset.

- To alleviate the data sparsity of group–item interaction, we propose Item Context Prediction (ICP) that predicts item context generated by graph embedding. The unsupervised loss of ICP is regarded as a regularization term for the recommendation.
- Experiments show that DGC achieves significant performance improvement over other existing methods on three real-world datasets.

2. Related works

2.1. Group recommendation

Group recommendation recommends items to a set of users. Earlier studies of group recommendation focus on defining a function for aggregating individual preferences [3]. Aggregation-based methods can be divided into two categories; preference-based aggregation and score-based aggregation. Preference-based aggregation methods [7,8] calculate prediction scores for group members and then aggregate these prediction scores. Representative functions for preference-based aggregation are CF-AVG and CF-LM, etc. CF-AVG averages prediction scores of group members and CF-LM maximizes the minimum preference scores of each group member. Score-based aggregation methods [9,10] create a virtual person that represents a group and then predict rating scores for the virtual person. Since aggregation methods simply combine personal preferences by using a predefined function, it is not sufficient to reflect the characteristics of a group. Model-based group recommendation has been actively studied recently.

Model-based methods design a model for relationships between a group and an item. Many model-based methods are based on a topic model. The PIT [11] is an author-topic model based group recommendation, which reflects the effects of users on group recommendation. COM [1] designs a model based on the intuition that selections of a group are mixtures of users' preferences, which are affected by topics of a group and the personal preferences of users about an item. HBGG [12] is a model that recommends locations to a group. They mentioned that group mobility regions affect the selections of locations to be visited by a group, so they proposed a method by considering a mobility region of group events. HGGC [13] is a hybrid group recommendation model considering group cohesion. Further, it is based on a topic model. Group cohesion indicates a proportion of group members who actively participate in group activities. They achieve performance improvement by reflecting group cohesion into their model. To alleviate a data sparsity problem, they reflect additional information. Recently, some neural networks based methods have been proposed. AGR [4] is a representative method based on neural network, which performs attentive group recommendation. It reflects the vanilla attention for group members on the preferences of a group. The AGR recommends items to a group and a user simultaneously. Although these previous methods attempt to model group behaviors, they are not sufficient to capture dynamic relationships between groups and items. In our work, we propose a neural-based DGC to model dynamic user preferences in group behavior appropriately and to reflect context information properly.

2.2. Neural network based recommendation

Neural networks have been widely used for various applications such as image recognition, voice recognition, word embedding, etc. Because of the effectiveness of neural networks in many domains, some researches try to apply the neural network to a collaborative filtering problem. A pioneer method [14] is a Restricted Boltzmann Machines (RBM) based on the collaborative filtering of explicit rating data. Recently, an autoencoder-based collaborative filtering technique is widely used, which learns hidden structures of rating data by reconstructing input data. To avoid the failure of generalization for unseen data, a denoising autoencoder (DAE) [15,16] is proposed. Dong et al. [17] incorporates auxiliary information to DAE for reducing a data sparsity problem. Zheng et al. [18] proposed autoregressive collaborative filtering. These previous works are meaningful for neural network based collaborative filtering. However, most of them targets the explicit rating data. To perform a collaborative filtering for implicit rating data, some methods are proposed [19–23]. Xiangnan He et al. [24] introduced a general framework for neural collaborative filtering. They propose a method that allows capturing non-linear features of rating data.

To improve the performance of the recommendation, some methods [25,26] try to reflect unlabeled data such as a context graph by using semi-supervised learning. This approach jointly optimizes both losses for supervised learning and for unsupervised learning. In [26], a general loss function for semi-supervised learning in neural network is defined as

$$\sum_{i \in l} L(f(x_i), y_i) + \lambda \sum_{i,j} L(g(x_i), g(x_j), W_{ij}) \tag{2}$$

Table 1
Notations.

Symbols	Description
F_g	Embedding vector for group g
$\hat{p}_u$	Embedding vector for user u trained in GMF
$\hat{q}_i$	Embedding vector for item i trained in GMF
x_i	Embedding vector of item i for group–item interaction
z_i	Embedding vector of item i for context information
$\tilde{F}_g$	Transformed matrix of F_g
$\tilde{q}_i$	Transformed vector of $\hat{q}_i$
zr_i	Latent vector of item i for prediction of rating score by using context information
zc_i	Latent vector of item i for context prediction
c_i	Feature of item i for context, which is obtained by graph embedding
ϕ^U	Prediction vector for user–item interaction
ϕ^G	Prediction vector for group–item interaction
ϕ^C	Prediction vector for context information
ϕ^{GCF}	Concatenated prediction vector
$\hat{r}_{gi}$	Rating score

where x_i indicates ith features, y_i denotes x_i's label, and l denotes a set of labeled data. Further, f denotes a neural network based prediction function and g means an embedding function. W_{ij} indicates an affinity matrix. If the graph data are used for context information, W_{ij} is 1 if there is an edge between node i and node j, otherwise 0. In addition λ denotes a constant for a weight factor. In this formula, $L(f(x_i), (y_i))$ is a supervised loss and $L(g(x_i), g(x_j), W(ij))$ denote an unsupervised loss. The unsupervised loss is used to force embedding vectors of x_i and x_j to have similar values when $W_{ij} = 1$. In the final loss function, unsupervised loss is operated as a regularization term, i.e., it provides a proper constraint for training the supervised loss.

3. Problem definition and preliminaries

3.1. Problem definition

Let G, U, and I be a set of groups, a set of users, and a set of items, respectively. Assume that there are m groups, l users, and n items. A group $g \in G$ is a set of users, $g \subseteq U$. Rating data received from a group are classified into two types: explicit feedback and implicit feedback. Explicit feedback indicates a score that a group explicitly gives based on their preference. Implicit feedback is an opinion that a group indirectly express through e.g., purchase, visit, and so on. Our paper aims to predict the ratings of unobserved interactions between groups and items based on implicit feedback data. An implicit feedback matrix is $R \in \mathbb{R}^{m\times n}$ where $R_{gi} = 1$ if an implicit feedback is observed between group g and item i, otherwise $R_{gi} = 0$. A matrix $Y \in \mathbb{R}^{l\times n}$ indicates the preferences of users on items. If user u prefers item i, $Y_{ui} = 1$, otherwise $Y_{ui} = 0$. Given G, U, I, R, and Y, our goal is to predict the missing rating scores $\hat{R}_{ij}$ between groups and items. Notations used in our models are summarized in Table 1.

3.2. Generalized matrix factorization (GMF)

Generalized matrix factorization (GMF) [24] is a generalized matrix factorization based on neural networks. It trains embedding vectors of users and items based on rating data thereby reflecting non-linearities existing in complex structures of interactions between users and items. The graphical representation of GMF is shown in Fig. 3. By using GMF, we generate user and item latent vectors based on Y to capture the features of the user–item interactions.

Inputs of GMF are one-hot encodings v_u^U and v_i^I of user u and item i, where the value is 1 at position u or i, otherwise 0. We

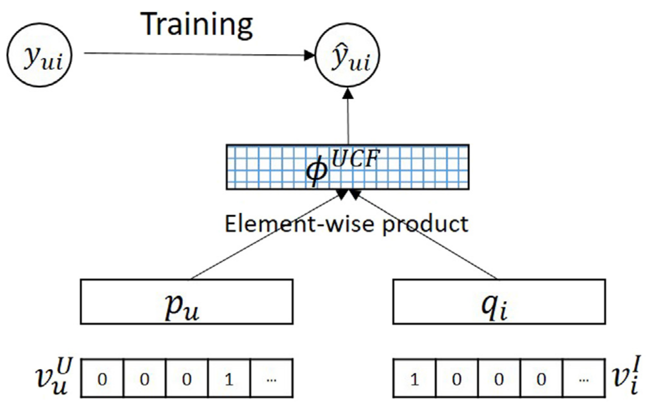


Fig. 3. GMF model.

feed v_u^U and v_i^I to an embedding layer for an informative vector representation (i.e., an embedding vector). Embedding vectors for u and i are defined as $p_u = P^T v_u^U$ and $q_i = Q^T v_i^I$ where $P \in \mathbb{R}^{l\times K}$ and $Q \in \mathbb{R}^{n\times K}$ are matrices denoting latent factors for user and items, respectively. K denotes the dimension of embedding vectors. Then, we employ the element-wise product as a mapping function.

$$\phi_{GMF}(p_u, q_i) = p_u \odot q_i \tag{3}$$

where $\phi^{GMF} \in \mathbb{R}^{K\times 1}$ and $\odot$ denotes the element-wise product.

To treat a binary class of an implicit feedback, an output layer of GMF is defined as

$$\hat{y}_{ui} = sigmoid(W_y^T(p_u \odot q_i)) \tag{4}$$

where W_y^T represents the weight vector. p_u identifies the preferences of user u and q_i implies features of item i.

By training GMF, we can acquire $\hat{p}_u$ and $\hat{q}_i$. The optimization function L_{GMF} of GMF is based on binary cross-entropy between real implicit rating scores and estimated rating scores, which are defined as

$$\begin{aligned} L_{GMF} &= -\sum_{(u,i)\in Y^+} log\hat{y}_{ui} - \sum_{(u,i)\in Y^-} log(1-\hat{y}_{ui}) \\ &= \sum_{(u,i)}\{y_{ui}log\hat{y}_{ui} + (1-y_{ui})log(1-\hat{y}_{ui})\} \end{aligned} \tag{5}$$

where Y^+ indicates a set of observed instances and Y^- denotes negative instances generated by uniform sampling. To optimize L_{GMF}, stochastic gradient descent (SGD) with mini-batch training is performed. Firstly, we randomly sample a batch of observed instances (i.e., positive instances), and we sample a batch of negative instances according to the predefined ratio. Then, L_{GMF} is trained by using SGD.

4. DGC model

4.1. Overview

Dynamic Group behavior modeling utilizing Context information for group recommendation (DGC), as shown in Fig. 4, consists of two parts for (1) dynamic group collaborative filtering (DGCF) and (2) item context prediction (ICP). (1) DGCF has two components; dynamic group behavior modeling and group collaborative filtering. To capture dynamic group behavior, we design dynamic group behavior modeling by using transformation from the user–item space to the group–item space. A set of user latent vectors are transformed to a group latent matrix. Through this process, a hidden structure of dynamic group behavior can be reflected

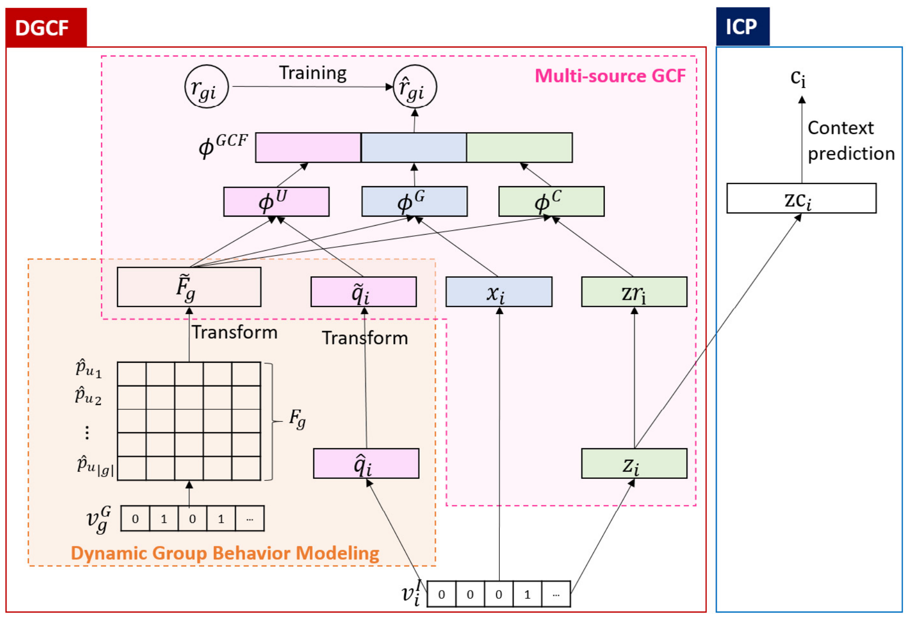


Fig. 4. The DGC framework.

to a transformation matrix. An item latent vector in the user–item space is transformed to that in the group–item space. In the collaborative filtering of a group, we represent group–item interactions as the inner product between a group latent matrix and three item latent vectors. Item latent vectors are generated from group–item interaction, user–item interaction and item's context information. Item latent vectors extracted from various data sources allow to more accurately capture features of group behavior. (2) ICP reflects context information to alleviate a data sparsity problem. Semi-supervised learning that combines a supervised loss and an unsupervised loss is used to boost supervised learning performance by using unlabeled data. The unsupervised loss plays a role as a constraint of the supervised loss. We define a unsupervised loss L_{ICP} for item context prediction, which is used for a regularization penalty for a supervised loss L_{DGCF} for recommendation. The loss function for DGC is a weighted sum of L_{DGCF} and L_{ICP}.

$$L_{DGC} = L_{DGCF} + L_{ICP} \tag{6}$$

4.2. Dynamic Group Collaborative Filtering (DGCF)

DGCF consists of two components; dynamic group behavior modeling and multi-source group collaborative filtering. In this section, we describe these two parts in more detail. Dynamic group behavior modeling is designed to reflect user's preference as being an individual may be different from his/her preference as being a member of group. Further, it can learn different effects of group members on group decision. This dynamic group behavior can be modeled as the transformation of the latent vectors of groups and items from a user–item space to a group–item space. We first define a group embedding matrix in the user–item space as a concatenation of members' embedding vectors. There are many ways to learn the latent vectors of users from the user–item interactions. In our methods, we use user latent vectors acquired by GMF in Section 3.2. If $g = \{u_1, u_2, \ldots, u_{|g|}\}$, where $|g|$ is the size of group g, then $F_g \in \mathbb{R}^{|g|\times K}$ becomes $[\hat{p}_{u_1}, \hat{p}_{u_2}, \ldots, \hat{p}_{u_{|g|}}]$, where K is a dimension of embedding vectors. The intuition for modeling a group embedding matrix as a set of embedding vectors of group members is as follows. A group is a set of users, and therefore, group preferences are inevitably affected by preferences of the group members. Moreover, modeling a group embedding matrix as the concatenation of the members' embedding vectors is useful to reflect the complex relationships between members using transformation. Next, we generate item embedding vectors in the user–item space. The item embedding vector $\hat{q}_i$ is also trained by GMF.

Since F_g and $\hat{q}_i$ are trained in a user–item space, they do not reflect dynamic group behavior that occur depending on their group memberships. To apply dynamic group behavior, we transform F_g and $\hat{q}_i$ in the user–item space to $\tilde{F}_g$ and $\tilde{q}_i$ in the group–item space as follows:

$$\tilde{F}_g = T_g^T F_g,\ \tilde{q}_i = T_I^T \hat{q}_i. \tag{7}$$

Two transformation matrices $T_g \in \mathbb{R}^{|g|\times K}$ for group g and $T_I \in \mathbb{R}^{K\times K}$ for item i are used, which are trained by minimizing errors between real rating values and the output of the model. Through this process, a hidden structure of the dynamic group behavior (effects of users, changed preferences of users about properties of items) can be established in T_g. A column of T_g represents the impacts of users. A row of T_g denotes a degree of changes in the user preferences about each item property when users are group members. T_I allows the embedding vectors of items in the user–item space to be appropriately located into the vectors of the group–item space. For the transformation of the group embedding matrix, the result of the matrix multiplication is interpreted as the concatenation of the outer product of a column vector of T_g (i.e., transformation vector for a member) and a latent vector for a member. It means that the F_g reflects the mixture of complex relationships between members in the group. On the other hand, the weighted average of members' latent vectors, which were used in existing methods, does not reflect features in relationships between members.

Another component, multi-source group collaborative filtering (Multi-source GCF), predicts ratings of groups based on latent factor modeling that represents group–item interactions as the inner product between a group latent matrix and an item latent vectors. In multi-source GCF, we generate three item latent vectors to reflect various features in the user–item space, group–item space, and context information. These item vectors captured from various sources allow us to reflect item features more accurately.

The latent vectors $\tilde{q}_i$ for the user–item space is acquired in dynamic group behavior modeling. A vector x_i that models item features in group–item interaction, which is a latent vector in group–item space , is obtained by feeding v_i^I to a fully connected embedding layer in which v_i^I is 1 if the ith index is 1, otherwise 0.

$$x_i = X^T v_i^I \tag{8}$$

where $X^T \in \mathbb{R}^{n\times K}$ denotes a latent factor matrix for items in the group–item space. To reflect the context features of item i, we define a new item embedding vector z_i as

$$z_i = Z^T v_i^I \tag{9}$$

where $Z^T \in \mathbb{R}^{n\times K}$ is a latent factor matrix.

Then, we feed the embedding vector z_i to a latent vector zr_i to reflect adaptive features for rating prediction.

$$zr_i = W_{zr} z_i + b_{zr} \tag{10}$$

where W_{zr} and b_{zr} are the weight matrix and the bias vector, respectively.

Next, prediction vectors are generated to model relationships between groups and items. The prediction vectors are a dot product between a group latent matrix and each of the three item latent vectors. Given that we have three item latent vectors $\tilde{q}_i$, x_i, and zr_i generated from user–item interactions, group–item interactions and item context information, ϕ_{gi}^U, ϕ_{gi}^G, and ϕ_{gi}^C for prediction are generated as follows:

$$\begin{aligned} \phi_{gi}^U &= \tilde{F}_g \tilde{q}_i \\ \phi_{gi}^G &= \tilde{F}_g x_i \\ \phi_{gi}^C &= \tilde{F}_g zr_i \end{aligned} \tag{11}$$

These vectors imply relationships between groups and items as discovered in different data sources. We, then, concatenate these vectors.

$$\phi_{gi}^{GCF} = \begin{bmatrix} \phi_{gi}^U \\ \phi_{gi}^G \\ \phi_{gi}^C \end{bmatrix} \tag{12}$$

Further, a final prediction score $\hat{r}_{gi}$ for group g and item i is

$$\hat{r}_{gi} = sigmoid(W_r \phi_{gi}^{GCF} + b_r) \tag{13}$$

where W_r and b_r are the weight vector and bias vector, respectively.

We use binary cross entropy as the loss for rating scores, which is defined as

$$L_{DGCF} = -\sum_{(g,i)\in R} r_{gi} log \hat{r}_{gi} - \sum_{(g,i)\in R^-} (1 - r_{gi}) log(1 - \hat{r}_{gi}) \tag{14}$$

where R^+ and R^- denote a set of observed interactions and a set of unobserved interactions in R, respectively. We uniformly sample unobserved instances in R^- during training time.

4.3. Item Context Prediction (ICP)

In our method, we use the context information of items for modeling group behavior. We define the item context information as interactions of items with other objects, except for group–item interactions that are targets to be predicted in our proposed model. The context information can be represented by heterogeneous information network (HIN) that can accommodate various types of information. To incorporate context information in our model more easily, we encode the context information using a graph embedding method ESim [6] that maps nodes of HIN to the vectors of real numbers such that these vectors preserve network properties, e.g., information about neighbors and paths. ESim produces meta-path guided embedding vectors. It makes vertices co-occurring in many path instances of the same meta-path have similar embedding vectors. A meta-path is a sequence of link types between two node types. For example, consider the Gowalla network in Fig. 2, where "venues" corresponds to "items". There are meta-paths, both starting from and ending in the Venue type, whose length is less than or equal to 3; for example, Venue–Spot–Venue, Venue–Category–Venue, Venue–User–Venue, and Venue–User–User–Venue (thick lines in Fig. 2). Embedding vectors for venues that preserve the interactions of venues with other objects, i.e., spots, categories, and users will be generated. That is, these embedding vectors contain the context information of items and they are used for our group behavior modeling.

The feature $c_i \in \mathbb{R}^{1\times D}$ is denotes the context information of item i where D denotes the dimensions of the feature. To reflect context information in our model, we generate a latent vector zc_i for context prediction by feeding z_i, Then, we predict context feature c_i from zc_i. This makes our latent vector zc_i is fit to c_i, i.e., zc_i involves context features existing in HIN.

$$zc_i = tanh(W_{zc} z_i + b_{zc}) \tag{15}$$

where W_{zc} denotes a weight matrix for fully connected layer from z_i to zc_i and b_{zc} is a bias, respectively. The reason why we use *tanh* as an activation function is to force the range of zc_i to be the same as that of c_i.

As mentioned before, we use a semi-supervised loss function to reflect context information for recommendation. The supervised loss of the semi-supervised loss function is described in Section 4.2. Now, we define a unsupervised loss function by using a hinge loss. The hinge loss is generally used for maximum margin classification such as SVM, which reduces misclassifications and leads to better accuracy.

The loss function for context prediction is defined as follows. Given a latent vector zc_i and a context c_i of item i, an objective function L_{ICP} for context information prediction is

$$L_{ICP} = -\sum_{(zc_i, c_i)} max(0, 1 - zc_i \cdot c_i) \tag{16}$$

L_{ICP} is operated as the regularizer of the total loss function L_{DGC}. We can easily generalize this idea by utilizing context information about users and groups with a little overhead on the group recommendation model.

5. Learning and prediction

We jointly train our loss function L_{DGC} composed two loss functions L_{DGCF} and L_{ICP} using SGD with Adam instead of vanilla stochastic gradient descent (SGD) because the Adam optimizer achieves fast convergences and reduces memory requirements. We first create a batch of labeled data (g, i, r_{gi}) and perform gradient steps to optimize L_{DGCF}. Then, we sample a batch of unlabeled data (i, c_i) and optimize the L_{ICP} using SGD with Adam. These procedures are performed repeatedly until convergence. After learning parameters of our model, ratings $\hat{r}_{gi}$ for all test group g and all item i are predicted using DGC. Then, we rank items based on the estimated rating.

6. Experiments

6.1. Experimental settings

6.1.1. Dataset

We evaluate our method on four real-world datasets; Meetup CA, Meetup NYC, Gowalla and CAMRa2011. Meetup [27] is an

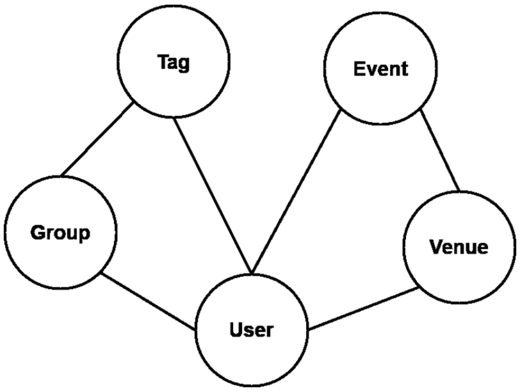


Fig. 5. Schema of Meetup dataset.

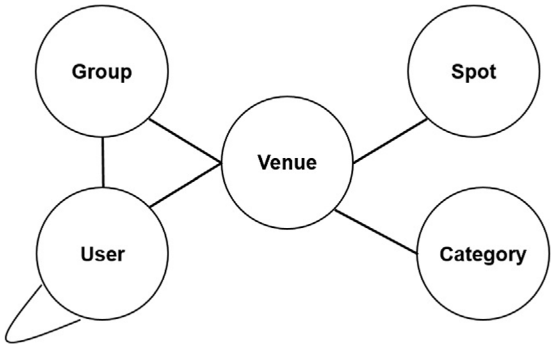


Fig. 6. Schema of Gowalla dataset.

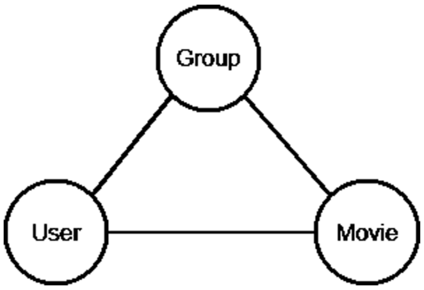


Fig. 7. Schema of CAMRa2011 dataset.

Table 2
Statistics of datasets.

Dataset	Meetup NYC	Meetup CA	Gowalla	CAMRa2011
# Groups	375	606	427	290
# Users	45,510	56,876	320	602
# Items	1,049	1,489	7,673	7,710
# avg. members for a group	382.71	256.76	2.20	2.08
# avg. interactions for a group	9.37	8.61	22.07	502.15

event-based social network that contains various types of information. We use two datasets from Meetup; the state of California (CA) and New York (NYC). Meetup has five types of entities: Group, Tag, User, Event and Venue. A schema of heterogeneous information network for Meetup is shown in Fig. 5. In our method, we perform experiments to recommend tags for groups, i.e., tags are regarded as items. To reflect various features of entities for the Tag type, we use meta-paths, both starting from and ending in the Venue type, whose length is less than or equal to 4 such as Tag–User–Tag, Tag–User–Group–Tag, Tag–User–Event–User–Tag, and Tag–User–Venue–User–Tag. Further, we use the Gowalla (GWL) dataset generated by Liu et al. [28], which is a location-based social network. It does not contain explicit group information so we extract implicit groups based on [29]. Purushotham et al. [29] regards a set of users who visit the same venue within 1 h and follow each other as a group. A schema of GWL is described in Fig. 6, which has five types of entities: Group, User, Venue, Spot, and Category. In GWL, we recommend venues to a group for our experiment. Interesting meta-paths are Venue–Spot–Venue, Venue–Category–Venue, Venue–User–Venue, and Venue–User–User–Venue. A final dataset for experiments is CAMRa2011 [4]. CAMRa2011 contains movie rating records of individual users and households. We treat households as groups and recommend movies to a group (i.e., a movie is regarded as an item). There are three types of entities; Group, User, and Movies. The schema of CAMRa2011 are described in Fig. 7. Target meta-paths to generate embedding vectors for item context are Movie–User–Movie and Movie–User–Group–User–Movie. Statistics for these data are given in Table 2. From this heterogeneous information network, item context information c_i is generated by ESim. For each dataset, we randomly select 20% items for each group as the test dataset while the remaining items are used for the training dataset.

6.1.2. Comparison methods

The following are representative group recommendation methods used for performance comparison with our proposed method.

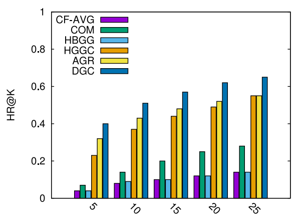
- CF-AVG [30] is one of the baseline methods, which ranks items by averaging the personal preferences of group members.
- COM [1] is a topic model based group recommendation method. The model is designed under these assumptions: (i) each group selects items based on their topics. (ii) The topics are decided as group members' preferences. (iii) Preferences of group members are dynamic.
- HBGG [12] is a group recommendation model that is designed by using a probabilistic topic model. This model improves the performance by combining geographical information with a group recommendation model.
- AGR [4] is an attentive group recommendation based on a neural network. It uses vanilla attention to measure the importance of group members on the preferences of a group. AGR recommends items to a group and a user, simultaneously. In our experiments, we compare group recommendation performance of AGR with that of our method.
- HGGC [13] is a topic model based group recommendation. It enhances performance by considering group cohesion.

6.1.3. Evaluation metrics

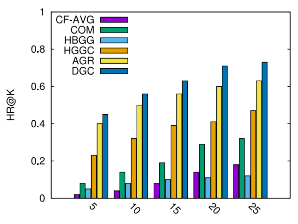
We use two representative metrics to analyze the performances of our method; Hit Ratio at Top N (HR@N) and Normalized Discounted Cumulative Gain at top N (NDCG@N). HR@K is the fraction of hits over the number of groups. The hits for a group indicates that any test item of the group appears in the recommendation list. NDCG@K is calculated based on correctly predicted ranks in the top N list. It assigns higher scores to higher ranked items.

6.2. Implementation detail

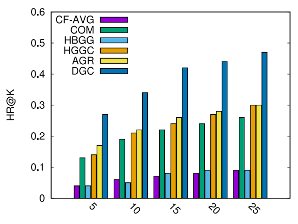
We implement DGC using a PyTorch framework accelerated by Nvidia GTX 1080 Ti GPU. For initialization, we apply a Xavier initialization method. We optimize our method based on the Adam



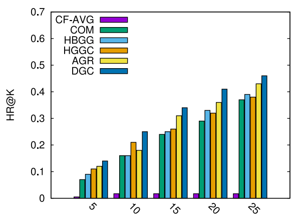
(a) HR - Meetup CA



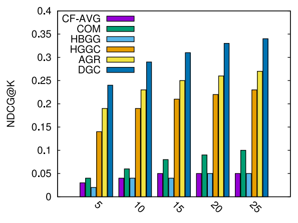
(b) HR - Meetup NYC



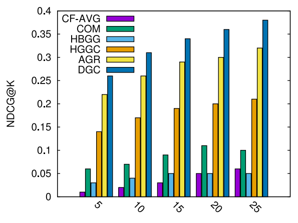
(c) HR - Gowalla



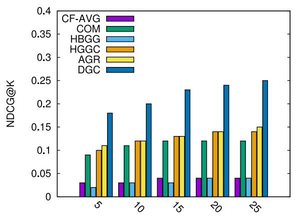
(d) HR - CAMRa2011



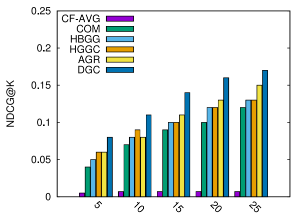
(e) NDCG - Meetup CA



(f) NDCG - Meetup NYC



(g) NDCG - Gowalla



(h) NDCG - CAMRa2011

Fig. 8. Performance comparison with various K.

optimizer. For experiments, we set the number of factor as 32. An embedding vector for the item's context information is generated with 50 dimensions by default. For each group, we rank items by using all unrated items as negative items. The models are trained for 50 epochs. We explore a mini-batch size and a learning rate for [128, 256, 512, 1024] and [0.001, 0.005, 0.01, 0.05, 0.1]. As a result, 256 is the best mini-batch size. 0.001 is the best learning rate for Meetup CA and NYC. For Gowalla and CAMRa2011, 0.01 is selected for the best learning rate. The evaluation results are the average results repeating the experiments five times.

6.3. *Performance comparison*

Fig. 8 shows the results of previous methods and our proposed methods DGC on four real-world datasets, Meetup CA, Meetup NYC, Gowlla, and CAMRa2011 in that order. For performance comparison, we set the ratio of negative samples to 4. We perform performance comparison for the Top K items of the ranked list with K set to [5, 10, 15, 20, 25].

We observe that the performances of the aggregation-based method CF-AVG is considerably poor. Note that this aggregation-based method simply aggregates users' preferences, which does not consider group behavior. Compared to CF-AVG, the topic model-based approaches such as COM, HBGG, and HGGC have better performances in general because these methods attempt to design a model considering complex group behavior. COM reflects the dynamic preferences of users and HBGG attempts to reflect the additional information. We can observe that COM achieves better performances than HBGG, which results from appropriately modeling group topics as mixtures of dynamic users' preferences. HGGC reflects the group cohesion, which contributes to performance improvements. The AGR and DGC outperform CF-AVG, COM, HBGG, and HGGC. From these experiments, we can know that neural network based methods achieve better performance compared to aggregation based and topic model based methods.

The experimental results show that DGC achieves remarkable performance improvements for all evaluation metrics on all datasets. The HR@10 of DGC increase by 12%, 19%, 54%, and 31% for Meetup NYC, Meetup CA, Gowalla and CAMRa2011, respectively, compared with that for AGR. Through extensive performance experiments, we show that our proposed model, is properly designed. Next, we analyze the effect of each component in more detail.

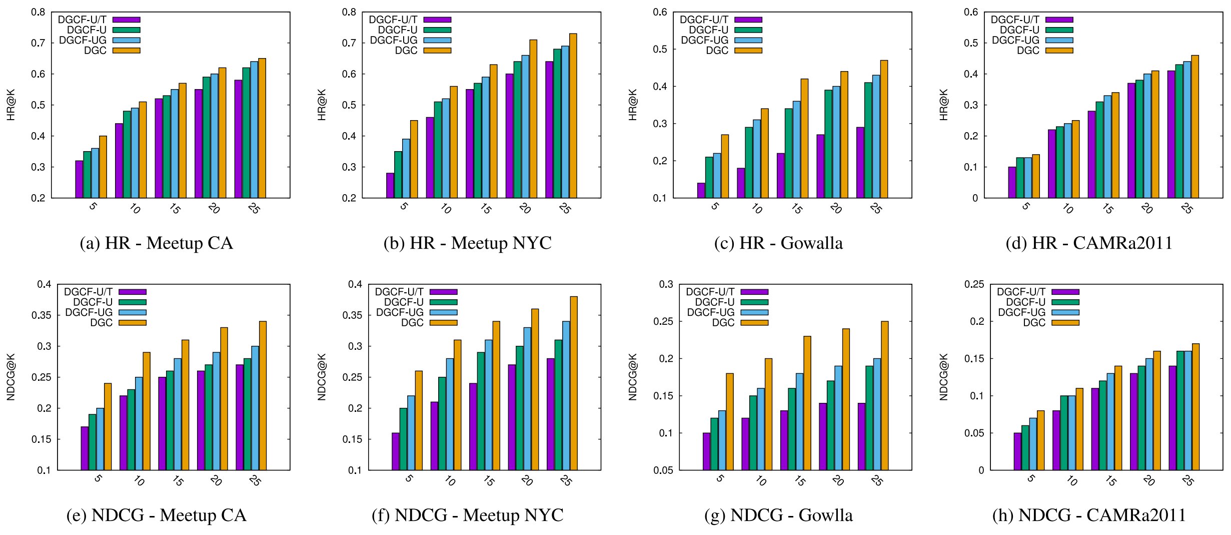
6.4. *Ablation study*

In this section, to further understand the importance of each components, we compare DGC with its three variants.

- DGCF-U/T: DGCF-U/T predicts the labels only by using $\phi_{gi}^{U} = F_g \hat{q}_i$ without transformation of F_g and $\hat{q}_i$. That is, $\phi_{gi}^{GCF} = \phi_{gi}^{U}$.
- DGCF-U: DGCF-U performs dynamic group behavior modeling, while it does not use ϕ_{gi}^{G} and ϕ_{gi}^{C} when generating ϕ_{gi}^{GCF}. That is, we calculate $\phi_{gi}^{U} = \tilde{F}_g \tilde{q}_i$ after transforming F_g and $\hat{q}_i$, and use this ϕ_{gi}^{U} for ϕ_{gi}^{GFC} when predicting labels.
- DGCF-UG: DGCF-UG performs dynamic group behavior modeling, while it does not apply context information. Thus, the prediction vectors are as follows; $\phi_{gi}^{U} = \tilde{F}_g \tilde{q}_i$ and $\phi_{gi}^{G} = \tilde{F}_g x_i$ and we generate $\phi_{gi}^{GCF} = \begin{bmatrix} \phi_{gi}^{U} \\ \phi_{gi}^{G} \end{bmatrix}$.

Fig. 9 shows the performances of DGCF-U/T, DGCF-U, DGCF-UG and DGC on four datasets: Meetup CA, Meetup NYC, GWL, and CAMRa2011. In the figure, DGCF-U outperforms DGCF-U/T in all cases on all datasets, which indicates that the dynamic group behavior modeling is effective on group recommendation. The performance of DGCF-UG is better than DGCF-U, which implies that both features extracted from user-item interactions and group-item interactions contribute to the performance improvement of group recommendation. We also observe that DGC achieves better performance than DGCF-UG on all datasets. It reveals that utilizing item context information is beneficial to group recommendation. Specifically, in Gowalla having various context information of items, the performance is greatly improved. It implies that context information is important for performance improvement.

Table 3 that describes the performance comparison of the representative existing method AGR with each component of our method in four datasets; Meetup CA, Meetup NYC, GWL and CAMRa2011. AGR is known to show the best performance on group recommendation as far as we are aware of. Our method without dynamic behavior modeling does not show good performance compared with AGR. However, we can see that DGCF-U, DGCF-UG and DGC which apply dynamic behavior modeling perform better than AGR in almost all cases.



(a) HR - Meetup CA (b) HR - Meetup NYC (c) HR - Gowalla (d) HR - CAMRa2011

(e) NDCG - Meetup CA (f) NDCG - Meetup NYC (g) NDCG - Gowlla (h) NDCG - CAMRa2011

Fig. 9. Ablation Study.

Table 3
Performance comparison (HR@K) of AGR with each component of our method.

K	AGR	DGCF-U/T	DGCF-U	DGCF-UG	DGC
5	0.32	0.32	0.35	0.36	0.40
10	0.43	0.44	0.48	0.49	0.51
15	0.48	0.52	0.53	0.55	0.57
20	0.52	0.55	0.59	0.60	0.62
25	0.55	0.58	0.62	0.64	0.65

(a) Meetup CA dataset

K	AGR	DGCF-U/T	DGCF-U	DGCF-UG	DGC
5	0.40	0.28	0.35	0.39	0.45
10	0.50	0.46	0.51	0.52	0.56
15	0.56	0.55	0.57	0.59	0.63
20	0.60	0.60	0.64	0.66	0.71
25	0.63	0.64	0.68	0.69	0.73

(b) Meetup NYC dataset

K	AGR	DGCF-U/T	DGCF-U	DGCF-UG	DGC
5	0.17	0.14	0.21	0.22	0.27
10	0.22	0.18	0.29	0.31	0.34
15	0.26	0.22	0.34	0.36	0.42
20	0.28	0.27	0.39	0.40	0.44
25	0.30	0.29	0.41	0.43	0.47

(c) GWL dataset

K	AGR	DGCF-U/T	DGCF-U	DGCF-UG	DGC
5	0.12	0.10	0.13	0.13	0.14
10	0.18	0.22	0.23	0.24	0.25
15	0.31	0.28	0.31	0.33	0.34
20	0.36	0.37	0.38	0.40	0.41
25	0.43	0.41	0.43	0.44	0.46

(d) CAMRa2011 dataset

6.5. Importance of transformation

The primary motivation of our work is to learn a transformation matrix from the concatenation of the latent factors of users in an user-item space to a group latent factor in a group-item space. For further illustrating effectiveness of the transformation, we perform two case studies. First, we randomly select two pairs of a group and an item, (G102, I456) and (G69, I561), from the test sets of Meetup CA and Meetup NYC, respectively. G102 consists of 7 users (U40096, U46793, U33560, U31861, U46923, U46827, U40126) and G69 consists of 13 users (U44641, U24789, U19931, U1732, U24883, U32835, U44633, U24885, U24854, U24822, U24839, U44642, U44643). Then, we

(a) Heatmap of transformation matrix T_{102} for $G102$ of Meetup CA

(b) Heatmap of transformation matrix T_{69} for $G69$ of Meetup NYC

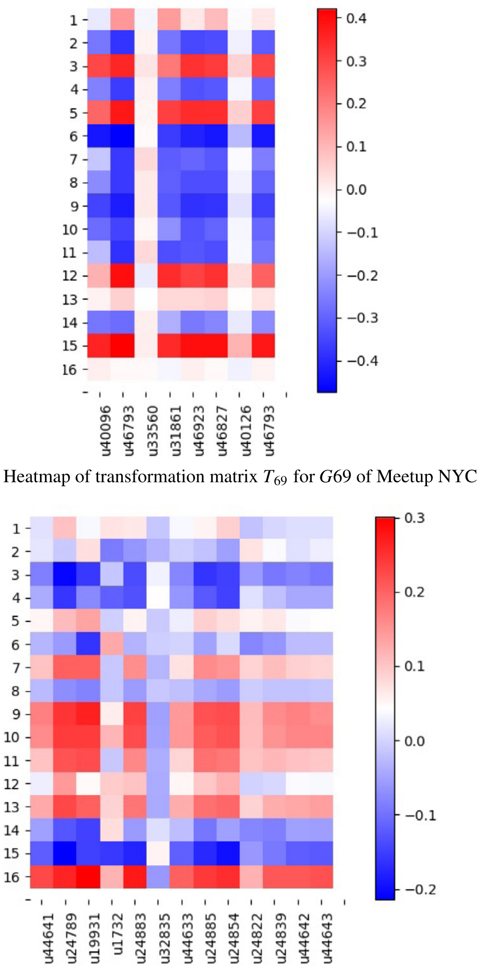


Fig. 10. Heatmap of transformation matrix.

train the DGCF-UG model and extract a trained transformation matrix. Fig. 10 shows the heatmaps of two trained transformation matrices for (G102, I456) and (G69, I561). In the color scale, the red color indicates the value is positive and the blue color denotes

Table 4
Effect of the number of negative samples in Meetup NYC.

	HR@5	HR@10	HR@15	HR@20	HR@25
1	0.3969	0.5254	0.6052	0.6715	0.7047
2	0.3710	0.5295	0.6223	0.6767	0.7130
3	0.3855	0.5492	0.6311	0.6870	0.7192
4	0.4508	0.5596	0.6321	0.7098	0.7306
5	0.3938	0.5368	0.6321	0.6912	0.7295

the value is negative. In addition, a darker color indicates that an absolute value is larger and a lighter color indicates that an absolute value is smaller. Each row of the heatmap denotes a group member, and each column is a latent dimension. In this example, we use 16 latent dimensions for an effective description.

Fig. 10(a) describes a transformation matrix T_{102} of (G102, I456). We observe several other groups to which U46793, U40096, U31861, U46823, or U46827 belong so as to select the item I456 in training data. This observation means that U46793, U40096, U31861, U46823, and U46827 have a high level of influence on the decision of G102. Absolute values are relatively high in column vectors for U46793, U40096, U31861, U46823 and U46827. Fig. 10(b) illustrates a transformation matrix for a pair (G69, I561) in the Meetup NYC dataset. We observe that groups, which U24789, U19931, U24883, U24885, U24839, and U24854 belong to, select item I561 in the training dataset. In Fig. 10(b), the column vectors of these users have higher absolute values. From these two examples, we can know that higher weights are assigned to influential group members in the transformation matrix.

6.6. Effect of the ratio of Top K

We illustrate the effects of varying K on Top K, which denotes the number of interest items. Evaluation is performed on Meetup CA and Meetup NYC where the value of K is among [5, 10, 15, 20, 25]. The results are described in Fig. 8. Since HR@K and NDCG@K show similar patterns, we focus our analysis on HR@K. Performances for all comparison methods and our proposed method show a continuous improvement as the K increases.

6.7. Effect of the number of negative samples

Table 4 lists the performance of DGC in Meetup NYC when increasing the number of negative samples (1 to 5) drawn for each positive sample. In Table 4, we can observe that our method achieves the best performance when the number of negative samples is 4. Larger than 4, the performance slightly degrades. These results confirm that we can ensure efficiency in training with a small number of negative samples.

7. Conclusion

We proposed dynamic group behavior modeling that utilizes context information for group recommendation, called DGC. The DGC consists of two components: DGCF and ICP. In DGCF, dynamic group behavior such as dynamic preferences and different effects of group members can be modeled by dynamic group behavior modeling that transforms latent vectors for users and items in user–item space to those in group–item space. Through this transformation, complex patterns of group decision are reflected into the trained transformation matrix. Further, DGCF includes multi-source collaborative filtering that predicts ratings for groups using item features extracted from multiple sources. Context information of items can affect the quality of group recommendation, and therefore, our DGC model reflects context information through ICP. It contributes to alleviate a data sparsity problem. To validate our methods, extensive experiments were performed, which showed that our method provides significant performance improvement over other previous methods.

CRediT authorship contribution statement

Hyun Ji Jeong: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Kwang Hee Lee:** Conceptualization, Methodology, Software. **Myoung Ho Kim:** Conceptualization, Validation, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1A2C1004032), the MSIT (Ministry of Science and ICT), Republic of Korea under the ITRC (Information Technology Research Center) support program (IITP-2020-0-01795) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation), and the Bio & Medical Technology Development Program of the National Research Foundation (NRF), Republic of Korea funded by the Korean government (MSIT) (No. 2013M3A9C4078137).

References

[1] Q. Yuan, G. Cong, C.-Y. Lin, COM: A generative model for group recommendation, in: KDD '14, 2014, pp. 163–172.
[2] S. Berkovsky, J. Freyne, Group-based recipe recommendations analysis of data aggregation strategies, in: Proceedings of the Fourth ACM Conference on Recommender Systems, ACM, 2010, pp. 111–118.
[3] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank aggregation methods for the web, in: Proceedings of the 10th International Conference on World Wide Web, in: WWW '01, ACM, New York, NY, USA, ISBN: 1-58113-348-0, 2001, pp. 613–622, http://dx.doi.org/10.1145/371920.372165, http://doi.acm.org/10.1145/371920.372165.
[4] D. Cao, X. He, L. Miao, Y. An, C. Yang, R. Hong, Attentive group recommendation, in: The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval, in: SIGIR '18, ACM, New York, NY, USA, ISBN: 978-1-4503-5657-2, 2018, pp. 645–654, http://dx.doi.org/10.1145/3209978.3209998, http://doi.acm.org/10.1145/3209978.3209998.
[5] B. Hu, C. Shi, W.X. Zhao, P.S. Yu, Leveraging meta-path based context for top-n recommendation with a neural co-attention model, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1531–1540.
[6] J. Shang, M. Qu, J. Liu, L.M. Kaplan, J. Han, J. Peng, Meta-path guided embedding for similarity search in large-scale heterogeneous information networks, 2016, arXiv preprint arXiv:1610.09769.
[7] J.F. McCarthy, T.D. Anagnost, MusicFX: an arbiter of group preferences for computer supported collaborative workouts, in: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work, ACM, 1998, pp. 363–372.
[8] Z. Yu, X. Zhou, Y. Hao, J. Gu, TV Program recommendation for multiple viewers based on user profile merging, User Model. User-Adapt. Interact. 16 (1) (2006) 63–82.
[9] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank aggregation methods for the web, in: Proceedings of the 10th International Conference on World Wide Web, ACM, 2001, pp. 613–622.
[10] J. Masthoff, Group modeling: Selecting a sequence of television items to suit a group of viewers, in: Personalized Digital Television, Springer, 2004, pp. 93–141.
[11] X. Liu, Y. Tian, M. Ye, W.-C. Lee, Exploring personal impact for group recommendation, in: CIKM '12, 2012, pp. 674–683.

[12] Z. Lu, H. Li, N. Mamoulis, D. Cheung, HBGG: a hierarchical Bayesian geographical model for group recommendation, in: SDM '17, ISBN 978-1-61197-497-3, 2017, pp. 372–380.
[13] H.J. Jeong, M.H. Kim, HGGC: A hybrid group recommendation model considering group cohesion, Expert Syst. Appl. 136 (2019) 73–82.
[14] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann machines for collaborative filtering, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 791–798.
[15] S. Li, J. Kawale, Y. Fu, Deep collaborative filtering via marginalized denoising auto-encoder, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 811–820.
[16] F. Strub, J. Mary, Collaborative filtering with stacked denoising autoencoders and sparse inputs, in: NIPS Workshop on Machine Learning for ECommerce, 2015.
[17] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, F. Zhang, A hybrid collaborative filtering model with deep structure for recommender systems, in: AAAI, 2017, pp. 1309–1315.
[18] Y. Zheng, B. Tang, W. Ding, H. Zhou, A neural autoregressive approach to collaborative filtering, 2016, arXiv preprint arXiv:1605.09477.
[19] A.M. Elkahky, Y. Song, X. He, A multi-view deep learning approach for cross domain user modeling in recommendation systems, in: Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 278–288.
[20] A. Van den Oord, S. Dieleman, B. Schrauwen, Deep content-based music recommendation, in: Advances in Neural Information Processing Systems, 2013, pp. 2643–2651.
[21] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1235–1244.
[22] X. Wang, Y. Wang, Improving content-based and hybrid music recommendation using deep learning, in: Proceedings of the 22nd ACM International Conference on Multimedia, ACM, 2014, pp. 627–636.
[23] F. Zhang, N.J. Yuan, D. Lian, X. Xie, W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 353–362.
[24] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.
[25] C. Yang, L. Bai, C. Zhang, Q. Yuan, J. Han, Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 1245–1254.
[26] J. Weston, F. Ratle, H. Mobahi, R. Collobert, Deep learning via semi-supervised embedding, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 639–655.
[27] T.N. Pham, X. Li, G. Cong, Z. Zhang, A general graph-based model for recommendation in event-based social networks, in: 2015 ICDE, 2015, pp. 567–578.
[28] X. Liu, Y. Liu, K. Aberer, C. Miao, Personalized point-of-interest recommendation by mining users' preference transition, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, ACM, 2013, pp. 733–738.
[29] S. Purushotham, C.-C.J. Kuo, J. Shahabdeen, L. Nachman, Collaborative group-activity recommendation in location-based social networks, 2014.
[30] S. Amer-Yahia, S.B. Roy, A. Chawlat, G. Das, C. Yu, Group recommendation: Semantics and efficiency, Proc. VLDB Endow. 2 (1) (2009) 754–765.