Group-to-group Recommendation with Neural Graph Matching

Chunchen Wang
Beijing University of Posts and Telecommunications
Wei Wang
Tencent
Cheng Yang
Beijing University of Posts and Telecommunications
Chuan Shi
shichuan@bupt.edu.cn
Beijing University of Posts and Telecommunications
Ruobing Xie
Tencent
Yuanfu Lu
Tencent
Haili Yang
Tencent
Xu Zhang
Tencent

Research Article

Keywords: Recommendation, Graph Neural Network, Data Mining.

Posted Date: July 18th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3107838/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at World Wide Web on March 5th, 2024. See the published version at https://doi.org/10.1007/s11280-024-01250-x.

Group-to-group Recommendation with Neural Graph Matching

Chunchen Wang[1,2], Wei Wang[2], Cheng Yang[1], Chuan Shi[1*], Ruobing Xie[2], Yuanfu Lu[2], Haili Yang[2], Xu Zhang[2]

[1*]School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, 100876, China.
[2]WeChat Search Application Department, Tencent, Beijing, 100000, China.

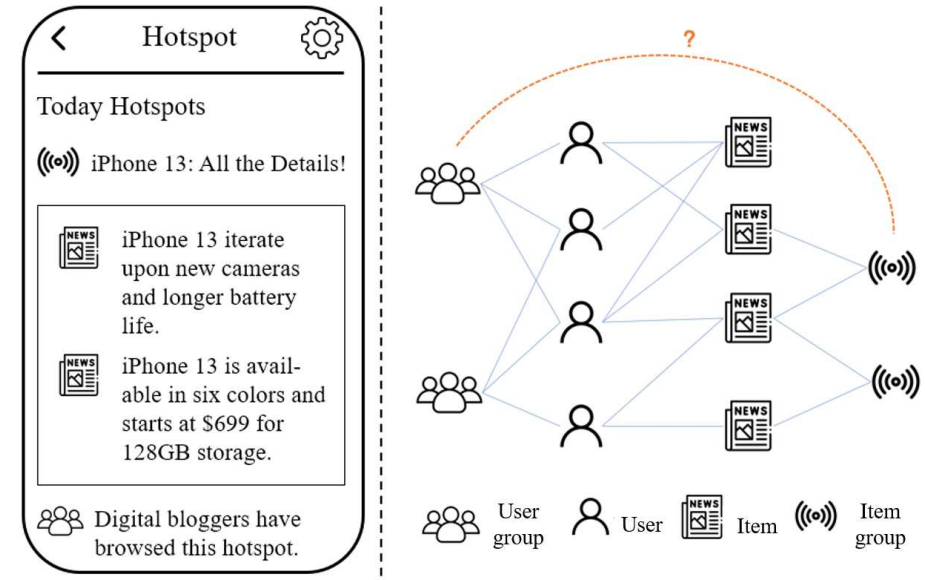
*Corresponding author(s). E-mail(s): shichuan@bupt.edu.cn;
Contributing authors: wangchunchen@bupt.edu.cn; unoywang@tencent.com; yangcheng@bupt.edu.cn; xrbsnowing@163.com; lucasyflu@tencent.com; hailiyang@tencent.com; xuonezhang@tencent.com;

Abstract

Nowadays, with the development of recommender systems, an emerging recommendation scenario called group-to-group recommendation has played a vital role in information acquisition for users. The new recommendation scenario seeks to recommend a group of related items to users with similar interests. To some extent, it alleviates the problem of point-to-point recommendations getting trapped in an information cocoon due to an over-reliance on user behaviors. For the new recommendation scenario, the existing recommendation methods cannot model the complex interactions between user groups and item groups, thus affecting the accuracy of the group-to-group recommendation. In this paper, we propose a group-to-group recommendation method, which abstracts user groups and item groups into graphs and calculates the similarity between two graphs based on graph matching, dubbed as GMRec. Specifically, we construct the graph of user groups and item groups and then calculate the graph similarity scores between user groups and item groups from two perspectives of feature matching and structure matching. Experimental results show that our model achieves higher accuracy than state-of-the-art models on three industrial datasets with different group sizes, with a maximum improvement of 8.2%.

Keywords: Recommendation, Graph Neural Network, Data Mining.

1 Introduction



a) Example of group-to-group recommendation

b) group-to-group recommendation

Fig. 1: A typical example of group-to-group recommendation in a real-world social platform.

With the booming mobile Internet and online social networks in recent years, people can get news faster and more conveniently, sharing it with their friends quickly. However, conventional point-to-point recommendation approaches [1, 2] heavily rely on users' behaviors and often result in an information cocoon, where recommended items are primarily based on users' browsing history. This poses a challenge in recommending new items to users. To overcome these limitations, new recommendation scenarios have emerged, such as group recommendation [3–5] and bundle recommendation [6–8]. These approaches aim to recommend items to a group of users or recommend a group of items to the user. They improve the efficiency of suggesting new items to users, often suggesting items that users haven't viewed previously.

A new recommendation scenario, group-to-group recommendation, is a novel approach that differs from existing personalized recommendations, which typically recommend items to individual users (i.e., point-to-point recommendation). This emerging recommendation scenario focuses on recommending multiple groups of items to a group of users who share similar interests and preferences. The characteristics of

a group depend on its members, who possess both shared characteristics and individual characteristics. Moreover, the recommendations are not limited to specific items but encompass items that share the same topic.

Fig.1 illustrates a toy example of group-to-group recommendation in a real-world scenario. This recommendation scenario involves suggesting a series of news to a group of users who share similar occupations or interests. As depicted in Fig1(a), both the user and other digital bloggers receive recommendations of two messages from the same item group. They belong to the same user group as they either share an interest in digital products or are themselves digital bloggers. In Fig. 1(b), we integrate the user-item interactions, user group-user affiliations, and item group-item affiliations into a unified graph representation. Unlike conventional recommendations that focus on suggesting specific items to individual users, the group-to-group recommendation scenario recommends groups of items to all users within the user group. Furthermore, unlike traditional recommendation scenarios, the group-to-group recommendation scenario provides information about the group and the underlying reasons for the recommendation. For instance, it may highlight that other technology bloggers are also following the recommended items, enhancing the interpretability of the recommendations and making them more appealing and user-friendly.

In the group-to-group recommendation scenario, there is no direct connection relationship between user groups and item groups. However, connections exist through users and items within each group. Due to the nature of breaking news, newly appearing news often lacks sufficient related reports and user engagement, such as clicks and follows. As a result, the group-to-group recommendation process faces two key challenges. (1) **How to model group-to-group recommendation relationships?** Existing recommendation methods cannot directly handle a group of users and a group of items simultaneously. Moreover, existing group recommendation and bundle recommendation methods overlook the intricate interactions between user groups and item groups. Hence, the aforementioned limitations of existing recommendation methods prevent them from directly recommending groups of items to groups of users. (2) **How to alleviate the sparsity between users and items in item groups?** In our group-to-group recommendation scenario, which is akin to recommending a group of trending news to a group of users, initially, only a few users have a direct association with the items in the item group. Therefore, it is crucial to enrich the user's representation with additional information beyond their direct involvement with any specific items.

In this paper, we propose GMRec, a novel model that utilizes graph matching and data enhancement techniques to predict the similarity between user groups and item groups. To address the first challenge, we construct a user group graph that incorporates user nodes and their associated tag features. Similarly, we construct another graph for the item group. Subsequently, we calculate the similarity score to determine the recommendation. To tackle the second challenge, we employ data enhancement techniques to alleviate the sparsity issue between users and items within item groups. In detail, we aim to enhance the representation of user groups and item groups by incorporating additional information across three aspects. The first aspect involves enriching the representation of users by incorporating items that are not directly

related to the item groups. This is achieved by calculating feature similarity and the number of common users. The second aspect focuses on utilizing tag-level information to enhance the representation of both user groups and item groups. The third aspect involves leveraging structural information to complement the feature information and enhance the representation of both user groups and item groups. In conclusion, we summarize our contributions as follows:

1) To the best of our knowledge, this work is the first attempt to make group-to-group recommendations for user groups and item groups based on graph matching.
2) We propose a new GMRec method with graph matching to model the group-to-group recommendation overcome that there are no direct connections between user group and item group and learn the representation of the both from global graph information, node-level & tag-level feature matching, and structure matching.
3) The experiments results on three industrial datasets show that GMRec is significantly superior to existing techniques. In addition, the model analysis reveals the role of each GMRec module.

2 RELATED WORKS

2.1 Collaborative Filtering

Factorization machine (FM) [1] is one of the most commonly used collaborative filtering algorithms, which considers the attribute interaction between users and items. FM models the interaction between each pair of attributes and summarizes all the modeling results to make the final prediction. AFM [9] and AutoFIS [10] are extended studies on FM. The former uses attention mechanism to learn different weights to measure different interactions and the latter proposes a two-stage algorithm which can automatically select important low-order and high-order feature interactions in factorization models. NFM [11] and DeepFM [12] add a multilayer perceptron (MLP) on top of attributes or attribute interactions. Compared with AFM and AutoFIS, MLP aims to implicitly capture the structural information in a non-linear way.

2.2 Graph Matching

Graph matching or Graph similarity search is a long-term research topic in computer science, and it has been extensively studied in the field of database and datamining [13, 14]. The graph similarity is usually defined by isomorphism of graph [15, 16], similarity of the structure, *i.e.*graph edit distance [17, 18] and graph kernel methods, such as random walks inside graphs [19, 20]. As the development of GNNs, GIN [21] concluded that GNNs are as powerful as weisfeiler-Lehman [22] algorithm in discriminant graphs. SimGNN [23]. uses the graph-level embeddings learned by GNNs and a node-level embeddings by a pairwise node comparison method to calculate similarity between two graphs. GMN [24] propose the newGraph Matching Networks that computes similarity through cross-graph attention-based matching. And GMCF [25] is the first to represent

each user and each item in a graph form and leverage the framework of neural graph matching for preference matching.

3 PRELIMINARIES

In this section, we will introduce the problem definition, the representations of graph, and basic idea of graph neural network.

3.1 Problem Definition

In this section, we will give a notation table shown in Table 1 and introduce the problem definition, the representations of graph, and basic idea of graph neural network.

Definition 1. ***Group-to-group recommendation scenario***. *The group-to-group recommendation scenario is defined as a undirected graph* $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ *with an object type mapping function* $\psi : \mathcal{V} \rightarrow \mathcal{O}$ *and a edge type mapping function* $\phi : E \rightarrow \mathbf{R}$. *And there are four object types, user group*(c)*, user*(u)*, item*(i)*, item group*(h) *in* $\mathcal{O}$ *and four relation types,* $< c, u >, < u, i >, < i, h >, < c, h >$ *in* $\mathcal{R}$.

Example1. As shown in Fig. 1, a user groups connects with a set of users, a user has many interactions with items and an item maybe belong to one or more item groups.

After get the definition of the group-to-group recommendation scenario, then we can define the problem of group-to-group recommendation.

Definition 2. ***Group-to-group Recommendation***. *Group-to -group recommendation aims to recommend a group of related items to a group of users with similar interests. In a Graph G abstracted from group-to-group recommendation scenario with node types* $\mathcal{O}$ *and relation types* $\mathcal{R}$*, let* $c, h \subset \mathcal{V}$ *denote the set of user groups and item groups, where* $c = (u_1, u_2, ..., u_m)$*,* $h = (i_1, i_2, ..., i_n)$*. Given a set of node pairs between a user group and an item group, i.e.,* $\mathcal{P} = \{p_{c,h} = 1 : c, u, i, h \in \mathcal{V}, < c, u > \in \mathcal{E}, < u, i > \in \mathcal{E}, < i, h > \in \mathcal{E}\}$*, we aim to predict whether to recommend a group of items to a group of users.*

3.2 Graph Neural Networks

Graph neural networks are generally applied to homogeneous graphs. We consider graph convolutional operations on a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}\}$, where $\mathcal{V}$ is the node set of graph $\mathcal{G}$, $\mathcal{E}$ is the edge set and $\mathcal{X} \in \mathbb{R}^{|\mathcal{V}| \times d_v}$ are node features. Although there are many kinds of GNN models, they basically follow similar operations, *i.e.*, node features aggregation and transform the message along the network topology for a certain layers. Formally, the $k-th$ layer representation of node v can be represented as:

$$\begin{aligned} \mathbf{Z} &= \mathbf{PROPAGATE}(\mathbf{X}; \mathcal{G}; K) \\ &= \left\langle \mathbf{Trans}\left(\mathbf{Agg}\{\mathcal{G}; \mathbf{H}^{k-1}\}\right)\right\rangle, \end{aligned} \tag{1}$$

with $\mathbf{Z^{(0)}} = \mathcal{X}$ and $\mathbf{Z}$ is the output representation after the $K-th$ layer graph neural network. $\mathbf{Agg}\{\mathcal{G}; \mathbf{Z}^{k-1}\}$ means aggregating the $(k-1)-th$ layer result $\mathbf{Z}^{k-1}$ along

Table 1: Descriptions of key notations.

Notations	Descriptions
$\mathcal{G}$	the graph abstracted from group-to-group recommendation scenario
$\mathcal{V}, \mathcal{E}$	the sets of nodes and edges
$\mathcal{O}, \mathcal{R}$	the sets of object types and edge types
c, u, i, h	node types of user group, user, item, item group
$< c, u >$	edge types of the edge between user group and item group
$\mathcal{G}^C$	the graph of user group
$\mathcal{V}^C, \mathcal{E}^C$	the sets of nodes and edges in graph of user group
$\mathcal{G}^H$	the graph of item group
$\mathcal{V}^H, \mathcal{E}^H$	the sets of nodes and edges in graph of item group
$\mathcal{I}^u$	the set of items that selected to be aggregated to get the user embedding
$\theta = (\psi, \omega)$	the parameters of GMRec
λ	L2-regularization parameter
f_{AGG}	aggregate function
$\mathcal{L}$	loss function
x_i	item feature of node i
z_u	user embedding
z_c, z_h	user group embedding and item group embedding
z_{uh}, m_{uh}	the cross interaction embedding and matching result between users and the item group
z_{ic}, m_{ic}	the cross interaction embedding and matching result between items and the user group
m_{ch}	the matching result between the user group tags and the item group
m_{hc}	the matching result between the item group tags and the user group
m_d	the structure-level matching score by the average degrees of the item group connected to the current user group and the degree of the user group
m'_d	the structure-level matching score by the average degrees of the user groups connected to the predicted item group and the degree of the item group
y'	the output of the GMRec, the probability of whether to recommend the item group to the user group

the graph $\mathcal{G}$ for the $k-th$ aggregate operation, and $\mathbf{Trans}(\cdot)$ is layer-wise feature transformation operation including weight matrix and non-linear activation function.

3.3 Graph Matching

If we want to calculate a score between two graphs $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{E}_1\}$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ to measure the similarity between them, we need get the graph representation of them. Each Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. Maybe each node $v_i \in \mathcal{V}$ has a feature vector x_i and each edge $< v_i, v_j > \in \mathcal{E}$ has a feature vector x_{ij} or not. Before we get the representation of the graph, we can learn the node embedding through GNN models. After that we can use sum, max- or mean-pooling to get the

graph representation and calculate graph similarity score through graph embedding scalar product.

4 METHODOLOGY

In this section, we present a novel method GMRec to address the group-to-group recommendation based on graph matching. Next we will introduce the overview of the GMRec and further elaborate the model components.

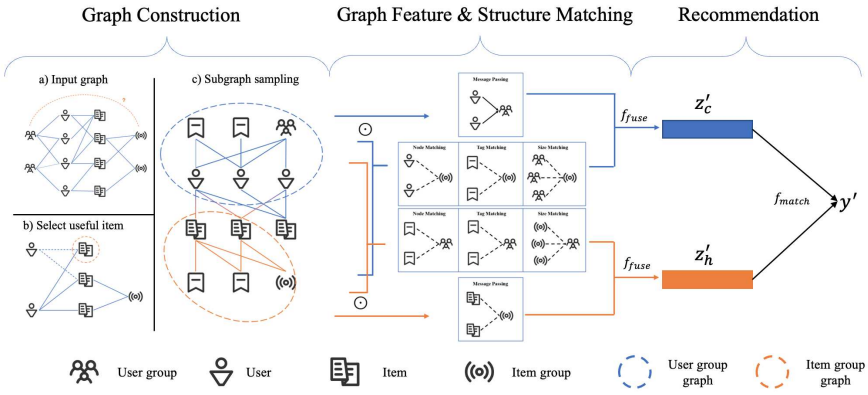


Fig. 2: An Overview of the GMRec model.

4.1 GMRec Overview

In order to solve the difficulty of predicting multiple links between user groups and item groups and improve the capability of recommendation system, we propose a new group-to-group recommendation model based on graph matching, named GMRec. Fig. 2 shows the framework of GMRec with a toy example. In this example, the final target is to predict the probability y' between a user group c and an item group h.

The proposed model GMRec mainly consists of three components: (1) Graph Construction module is responsible for constructing the original input graph abstracted from the group-to-group recommendation scenario into user groups and item groups. (2) graph feature & structure matching module generates the representation of user group and item group by matching the information of graph feature and graph structure. To consider the matching result between the two graphs from different levels, we match the two graphs through features from the global level, node level, and tag level. Then fusing the embedding with the structural information to obtain the representation of two graphs. (3) Finally, a recommendation module inputs the fused embeddings of user group and item group, and outputs the probability y' of the operation that

whether to recommend the item group h to the user group c. Here, we assume an example of input data sample, the input graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$.

4.2 Graph Construction

Obviously, the user graph consists a set of users and the item graph consists a set of items. We formally represent the graph pertinent to the user group as $\mathcal{G}^C = \mathcal{V}^C, \mathcal{E}^C$, where $\mathcal{V}^C$ encompasses not only the users but also their associated tag attributes, and $\mathcal{E}^C$ encapsulates the interconnections among the nodes within the user group graph, $\mathcal{G}^C$. Moreover, the selection of user nodes that constitute the graph is predicated upon their historical interaction with items within affiliated groups. This meticulously curated selection is instrumental in ensuring that the nodes genuinely epitomize the attributes and proclivities of users who have engaged in browsing activities involving items of correlated domains.

If consideration is restricted solely to items within specific groups, the representation of users tends to be skewed during the process of learning user embeddings. To ameliorate this limitation, we employ a dual-faceted approach to ascertain which items should be aggregated to represent the users more holistically. Firstly, an emphasis is placed on feature congruence, where items that exhibit a high degree of similarity in attributes with those within specific groups are selected. This is achieved by computing the cosine similarity among the features. For instance, in Figure 2c), the cosine similarity scores can be calculated between the first item and the remaining items to determine whether the item should be considered for aggregation via Graph Neural Network (GNN).

Concurrently, the selection of an appropriate item is performed through analysis at the user-behavior level. Specifically, the degree of relatedness between two items is evaluated based on the number of common neighbors they possess. For illustration, in Fig. 2c), the first and second items have two common users, which may indicate a heightened similarity between them.

Ultimately, the selected items are utilized to compute the user's representation, and pertinent tags are extracted from these items to further enrich the representation.

While in terms of graph of item group $\mathcal{G}^H = \{\mathcal{V}^H, \mathcal{E}^H\}$, node set $\mathcal{V}^H$ just contains items and the tags which are extracted from item features, $\mathcal{E}^H$ contains the connection relations of the group, items and tag features.

4.3 Graph Feature & Structure Matching.

In this subsection, we will introduce how to get graph representation through message passing and multi-aspect matching. We introduce a feature-matching technique that is predicated on Graph Neural Networks (GNNs) and operates at both the node-wise and tag-wise levels. This technique assimilates global information through message passing, appraises individual node preferences via cross-matching, and aligns tag-wise features. In parallel, the structure-matching approach contemplates the manner in which the dimensions of user and item groups could exert an influence on their inherent characteristics.

4.3.1 Global Representation through Message Passing

As Fig. 2 shows, the user group or item group we constructed is a tree structure, which we can use root node embedding as the whole structure representation. At first, the users' embedding need to be aggregated by the related item features, we can get the user's embedding z_u by:

$$z_u = f_{AGG}(x_i : i \in \mathcal{I}^u), \tag{2}$$

where f_{AGG} means mean aggregator, $\mathcal{I}^u$ means the set of items that selected to be aggregated to get the user embedding.

After get the user embedding z_u, we can get user group embedding z_c by

$$z_c = \textbf{PROPAGATE}(z_u; \mathcal{G}^C; K = 1), u \in \mathcal{V}^C, \tag{3}$$

where z_u is the embedding of users who belongs to the user group, $\mathcal{G}^C$ guide the direction of message passing, $K = 1$ means we only use one layer GNN to gain user group representation by aggregating user embeddings.

As the same way we get the embedding of user groups, we can also get item group embedding z_h by:

$$z_h = \textbf{PROPAGATE}(z_i; \mathcal{G}^H; K = 1), i \in \mathcal{V}^H, \tag{4}$$

where z_i is the embedding of items who are related to the item group, $\mathcal{G}^H$ guide the direction of message passing, $K = 1$ means we only use one layer GNN to aggregate item embeddings.

4.3.2 Node-wise Feature Matching

In the graph of user group, we acquire the representation of the entire group. However, it is imperative to recognize that individual users within the group may harbor distinct preferences. Motivated by this notion, and drawing inspiration from group recommendation strategies, we incorporate cross-interaction matching to compensate for any potential loss in the root node representation. Intuitively, for a user u in the user group, we anticipate a high matching score with an item group or individual item, if the user exhibits a pronounced preference for it. In the realm of collaborative filtering, post-training, the embeddings of entities with strong interdependencies tend to converge. Adopting the principles outlined in [25], we employ Bi-interaction [11] for node matching, which preserves the monotonically increasing correlation between interaction modeling results and attribute similarities. Specifically, we can model the cross interaction of user embedding and item group embedding as:

$$z_{uh} = z_u \odot z_h, u \in \mathcal{V}^C, \tag{5}$$

where z_u is the embedding of the user in user group and z_h is the embedding of item group, z_uh is the node matching result of two node and $\odot$ is the element-wise product. After we get one matching result between a user and an item group, we can get element-wise sum to aggregate the cross matching result between all users in

user group and an item group in another graph. The aggregated matching result is $m_{uh} = \sum_{u \in \mathcal{V}^C} z_{uh}$, where $\mathcal{V}^C$ is the node set of user group and m_{uh} is the matching result between the users in user group and the item group.

Similar to the cross matching between the users and item group, the cross matching z_{ic} between the items in the item group and user group can be calculated as:

$$z_{ic} = z_i \odot z_c, i \in \mathcal{V}^H, \tag{6}$$

where z_i is the embedding of items and z_c is the embedding of the user group. And the aggregated matching result of all items in item group and the user group in the other graph is $m_{ic} = \sum_{i \in \mathcal{V}^H} z_{ic}$, where $\mathcal{V}^H$ is the node set of item group and m_{ic} is the matching result between the items in item group and the user group.

4.3.3 Tag-wise Feature Matching

As Fig. 2 shows, the user group and item group are the first level, users and items are the second level and the tag features are the third level. As the node embeddings are all aggregated by item features and the tag features reflect the characteristics of the items. So we can see the tag features connected to the users as the characteristics of the items related to the user's browsing in the past. As same as the cross matching in Sec. 4.3.2, we can calculate the cross matching score between feature tags in user group and group node of item group:

$$m_{ch} = \sum z_{tag} \odot z_h, \forall tag \in \mathcal{V}^C, \tag{7}$$

where z_{tag} is the embedding of the tag which belongs to the user group and the embedding is acquired by pretrained model [26], z_h is the embedding of the user group, m_{ch} is the matching result between the user group tags and the item group.

As same to the matching result between tags in user group and the group of item, we can get the matching result between tags in item group and user group as:

$$m_{hc} = \sum z_{tag} \odot z_c, \forall tag \in \mathcal{V}^H, \tag{8}$$

where z_{tag} is the embedding of the tag which belongs to the item group, z_c is the embedding of the user group, m_{hc} is the matching result between the tags which are belonged to the item group and the user group.

4.3.4 Graph Structure Matching

In addition to considering the matching results of attribute features, we also consider the matching of structural information. Generally speaking, when a person browses some breaking news, he will tend to share the new findings with his friends and probably they are all in a same user group. Therefore, the law presented from the data level can be roughly reflected in the scale of user groups and item groups. Then we can get the degree embedding of user group and item group as d_C, d_H. After that, we will calculate the structure-level matching score by the average degrees of the item

groups connected to the current user group and the degree of the item group currently predicted:

$$m_d = mean(A_{CH} d_H) \odot d_h, \tag{9}$$

where $A_C H$ means the connections between user groups and item groups, d_H is the degree embeddings of all item groups, d_h is the embedding of predicted item group and m_d is the matching score. And we can also calculate the structure-level matching score by the average degrees of the user groups connected to the predicted item group and the degree of the user group as:

$$m'_d = mean(A_{CH} d_C) \odot d_c, \tag{10}$$

where d_C is the degree embeddings of all user groups and $m + d'$ is the matching score.

4.4 Recommendation

We can get the vector representations of the user group and item group through fusing global graph information and multi-aspect matching results. Specifically, the fusing function $f_{fuse} \in \mathbb{R}^{4\times d} \to \mathbb{R}^d$ takes the root node embedding z_c (or z_h), cross matching of leaf nodes z_{uh}(or z_{ic}), tag-wise matching $m_c h$(or $m_h c$) and structure wise matching m_d (or m'_d). Formally, we can get fused representation $z'_c = f_{fuse}(z_c, z_{uh}, m_c h, m_d)$ (or $z'_h = f_{fuse}(z_h, z_{ic}, m_h c, m'_d)$) as the final graph representation.

When it comes fusing function f_{fuse}, *i.e.*, concentration, element-wise addition or multilayer perceptron can be used to fuse the graph representation. Here, we choose vector concentration as the operation of fusing function. When we finally use graph matching to measure how similar the user group and item group are, we can use function f_{match},respectively:

$$y' = f_{match}(z'_c, z'_h) = (z'_c)^T z'_h, \tag{11}$$

where y' is the probability of the recommendation results between user group and item group.

4.5 Model Training

During training, we can minimize the loss for each group-to group pairs to train the model:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}\left(F_{GMRec}\left(\mathcal{G}_n^C, \mathcal{G}_n^H; \theta\right), y_n\right) + \lambda\left(\|\theta\|_2\right), \tag{12}$$

where F_{GMRec} is the prediction of our model that outputs the probability y', $\mathcal{L}(\cdot)$ corresponds to a loss function (*e.g.* binary cross-entropy loss), θ is the model parameters, λ is the L2-regularization parameter for reducing overfitting. With the aforementioned updating and inference rules, the training algorithm is shown in Algorithm 1.

Algorithm 1 Model training for GMRec

Require: Undirected graph $\mathcal{G}$; feature matrix $\mathcal{X}$; labels $\mathcal{Y}$;regularization weight λ; randomly initialized parameters θ.

Ensure: Final prediction y'.

Construct the graph of user group $\mathcal{G}^C = \{\mathcal{V}^C, \mathcal{E}^C\}$ and the graph of item group $\mathcal{G}^H = \{\mathcal{V}^H, \mathcal{E}^H\}$;
while not done **do**
  Set the batch of group-to-group pairs for training;
  for all group-to-group pairs $< c, h >$ in the batch **do**
    Calculate global representation z_c, z_h by Eq. (3-4);
    Calculate n-wise feature matching z_{uh}, z_{ic} by Eq. (5-6);
    Calculate t-wise feature matching $z_c h$, $z_h c$ by Eq. (7-8);
    Calculate structure-wise matching m_d, m'_d by Eq. (9-10);
    Calculate fused graph representation z'_c, z'_h;
    Predict the probability between user group and item group y' by Eq. (11);
    Calculate loss $\mathcal{L}$ by Eq. (12);
  end for
  Back-propagate the global parameter θ;
end while

5 EXPERIMENT

In this section, we conduct experiments on three industrial datasets to evaluate our model performance, and then present an interpretability analysis. Finally, we analyse the model components through ablation studies and parameter analysis.

5.1 Experiment Setup

5.1.1 Datasets

We compare our model with baselines through three industrial datasets of different sizes. Table 2 shows the statistic information of three datasets. Because we don't find a suitable public dataset with plaintext features. So we collect three industrial datasets from real business scenarios, which is the biggest social platform serving more than one billion users and users can browse and share articles. To protect user privacy, we anonymize the data and conduct strict desensitization processing. Each dataset contains a certain number of user groups, each user group is connected to a large number of users and each user will browse some items, and some items belonging to item groups. And the tag of users and items are extract from plaintext. Due to the principle of anonymity, we don't describe the source of the dataset in detail. We will indicate the source of the dataset and the company of the dataset after review. For each data sample of pairs of the user group and item group, the ground truth is whether the users in the user group have interactions more than 10 with the items in the item group. For each dataset, we split it into the training set, validation set, and test set.

Table 2: Statistics of the datasets.

Dataset	# User groups	Users	items	# Item groups
Dataset(small)	247	29,577	96,734	796
Dataset(median)	500	59,627	156,943	943
Dataset(large)	976	120,844	316,676	1,054

5.1.2 Baselines

Our proposed method will be compared with the following five types of baselines: (1) recommendation method focusing on feature interaction, including FM [1] and W&D [27]; (2) recommendation methods based on GNNs, including Fi-GNN [10] and NGCF [2]; (3) heterogeneous information network based information for recommendation, including HeteMF [28] and HERec [29]; (4) group recommendation including GAME [3] and GroupIM [4]; (5) Bundle Recommendation including DAM [6] and BGCN [7].

Table 3: Experimental results on three industrial datasets. The best method is bolded, and second best is underlined. The improvements of GMRec over the second best models are shown in the last row of each ratio.

Model	Dataset (small)[1]				Dataset (median)				Dataset (large)			
	N@20	**N@40**	**R@20**	**R@40**	**N@20**	**N@40**	**R@20**	**R@40**	**N@20**	**N@40**	**R@20**	**R@40**
FM	0.132	0.176	0.289	0.412	0.129	0.148	0.272	0.402	0.116	0.167	0.265	0.387
W&D	0.135	0.179	0.293	0.414	0.124	0.152	0.271	0.410	0.121	0.172	0.269	0.391
Fi-GNN	0.145	0.188	0.301	0.427	0.133	0.164	0.282	0.419	0.130	0.178	0.275	0.406
NGCF	0.143	0.186	0.302	0.425	0.134	0.161	0.276	0.405	0.126	0.176	0.273	0.395
HeteMF	0.136	0.182	0.296	0.413	0.131	0.152	0.274	0.407	0.121	0.170	0.269	0.388
HERec	0.139	0.184	0.297	0.412	0.135	0.151	0.276	0.409	0.123	0.171	0.273	0.393
GAME	0.189	0.237	0.342	0.469	0.193	0.251	0.366	0.479	0.176	0.251	0.334	0.423
GroupIM	0.192	0.231	0.345	0.467	0.194	0.253	0.372	0.484	0.178	0.252	0.337	0.431
DAM	0.193	0.236	0.352	0.473	0.191	0.250	0.369	0.478	0.176	0.249	0.329	0.417
BGCN	0.206	0.252	0.402	0.491	0.186	0.243	0.357	0.463	0.184	0.265	0.347	0.436
GMRec	**0.217**	**0.264**	**0.435**	**0.503**	**0.196**	**0.256**	**0.384**	**0.498**	**0.190**	**0.269**	**0.356**	**0.447**

5.1.3 Hyperparameter Settings

We randomly split each dataset into training, validation, and test set for each group-to-group pairs with a ratio of $6:2:2$. The validation set is only used to decide the best parameter settings, and the test set is only used to evaluate the models. By default, we use the following hyper-parameter settings: We randomly initialize parameters with Xavier initializer [30] and adopt Adam [31] to optimize our GMRec model. The pretrained node feature dimension is 128; the node representation dimension is 64. The learning rate is 1×10^{-3}; The λ for the regularization is 1×10^{-5}. The batch size is 64.

To avoid over-fitting, we apply an early stopping strategy and dropout (dropout rate is 0.5). We conduct experiments on a Linux server with a single GPU (GeForce RTX) and CPU (Intel Xeon E5-2620). We implement the proposed GMRec with PyTorch 1.4.0. The code and datasets will be released after review.

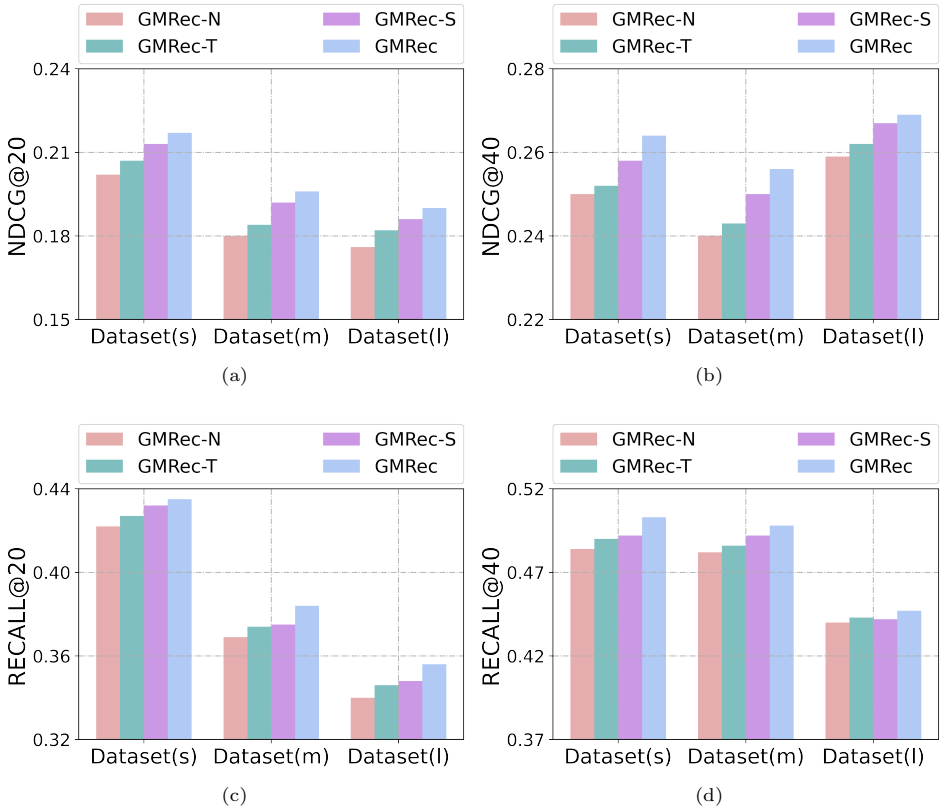


Fig. 3: Ablation study of GMRec components.

5.2 Main Experiments

In this section, we present the main results and compare our GMRec with four types of baselines. Table 3 respectively present the performances of all methods. The best and second-best results in each column are highlighted in bold font and underlined, respectively. We only present the most challenging baselines in Table 3 for brevity, and have the following observations:

- Our proposed GMRec significantly outperforms all baseline methods on the three industrial datasets with four different training ratios. It shows that our model

is effective and superior to the existing methods in solving the group-to-group recommendation problem. And it shows the robustness of the model.

• Compared with the point-to-point recommendation models, *i.e.,* FM, W&D, Fi-GNN, and NGCF, our model can further capture more complicated interactions between node attributes. FM and W&D does not consider the complicated attribute interaction between the user groups and item groups. NGCF only consider the similarity between user group and item group representations and ignore the complicated interactions between users and items. GMRec models attribute interactions in a structure of graph matching, which is more suitable for group-to-group recommendation and gains better performance.

• Compared with the HIN-based information for a recommendation, *i.e.,* HeteMF and HERec, our model capture more useful information through more aspects of matching, such as graph level matching level, node matching level, and structure matching level. Although the methods of HINs can learn more information about the different types of node and relations, maybe this information has little influence on this problem.

• Compared with the group recommendation and bundle recommendation, *i.e.,* GAME, GroupIM, DAM, and BGCN, our model can better capture the complicated attribute interactions between two groups. The method of group recommendation may consider mutual information between representations of groups and group members or multi-view embeddings for the groups, but they don't consider the influence between the group of users and the group of items. And the bundle recommendation may consider more about the related items that should be recommended together. But it ignores the influences between the related users. So our methods perform better than the group recommendation.

5.3 Analysis of Model Components

In this subsection, we conduct ablation studies to evaluate the performance improvement brought by node-wise matching, tag-wise matching, and structure-wise matching. We design three ablated models to show the performance of three modules. 1) The first ablated model **GMRec-N** removes the node-wise attribute feature matching module. 2) The second ablated model **GMRec-T** removes the tag-wise attribute feature matching module. 3) The third ablated model **GMRec-S** removes the structure-wise matching result, and only containing attribute feature module. Fig. 3 presents the results of our model and three ablated models. We can find that the node-wise feature matching module plays the most important role in our model. And tag-wise matching and structure-wise matching also enrich the information in the final representation.

5.4 Analysis of Model Hyperparameters

In this subsection, we will investigate how the primary hyper -parameters influence the performance of GMRec. In specific, we present the analysis of (1) the embedding dimension of node representation; (2) the number of additional items selected to enrich the user representation; (3) the batch size; (4) the effect of the regularization weight. The results are shown in Fig. 4 5 and we have the following observations:

• As shown in Fig. 4, our model performance will be improved and get steady when the dimension of node representations increases to 64 and the batch size increases to 64, which indicates the robustness of GMRec.

• In our experiments, we try to explore the impacts of the number of additional items which do not belong to item groups. The results are provided in Fig. 5(a). As we can find that our model performance can be improved with the increase in the number of items, and the performance tends to be stable when the number of aggregated neighbors reaches 20.

• As shown in Fig. 5(b), we summarize the impact of the L2-regularization parameter λ with different settings. We can find that the recommendation result will improve at first and reach the best performance at $\lambda = 1e - 5$. And then it will decrease. It can be answered by the reason that too small or too large a value of λ will have a bad effect on the model.

6 CONCLUSION

In this paper, we propose a novel model to solve the group-to-group recommendation problem. Based on the idea of graph matching, we consider the matching results between user groups and item groups from the feature and structure level. Such multi-faceted matching results can reflect the preference characteristics of a user group for an item group from various aspects, to improve the accuracy of recommendation. We conducted experiments on three realistic datasets and obtained the experimental results. Ablation experiments further illustrate the role of different components in our model. For future work, an interesting direction is to further consider the influence and weight of different users within the user group. For example, the weight of a very influential or active user is different. On the other hand, a learnable module can be used to select more appropriate items to enrich the user's presentation.

7 Declarations

- Ethical Approval We collect three industrial datasets from real business scenarios, which is the biggest social platform serving more than one billion users, and users can browse and share articles. To safeguard user privacy, we have implemented data anonymization techniques and rigorous desensitization measures.
- Competing interests not applicable
- Funding This work is supported in part by the National Natural Science Foundation of China (No. 62192784, U1936104, U20B2045, 62172052, 62002029).
- Authors' contributions Chunchen Wang and Cheng Yang wrote the main manuscript text. Wei Wang and Ruobing Xie prepared figures and fixed text. All authors reviewed the manuscript.
- Availability of data and materials We will indicate the source of the three industrial datasets and the company of the datasets after review.

References

[1] Rendle, S.: Factorization machines. In: Proceedings of ICDE, pp. 995–1000 (2010).

IEEE

[2] Wang, X., He, X., Wang, M., Feng, F., Chua, T.-S.: Neural graph collaborative filtering. In: Proceedings of SIGIR, pp. 165–174 (2019)

[3] He, Z., Chow, C.-Y., Zhang, J.-D.: Game: Learning graphical and attentive multi-view embeddings for occasional group recommendation. In: Proceedings of SIGIR, pp. 649–658 (2020)

[4] Sankar, A., Wu, Y., Wu, Y., Zhang, W., Yang, H., Sundaram, H.: Groupim: A mutual information maximization framework for neural group recommendation. In: Proceedings of SIGIR, pp. 1279–1288 (2020)

[5] Guo, L., Yin, H., Chen, T., Zhang, X., Zheng, K.: Hierarchical hyperedge embedding-based representation learning for group recommendation. ACM Transactions on Information Systems (TOIS) **40**(1), 1–27 (2021)

[6] Chen, L., Liu, Y., He, X., Gao, L., Zheng, Z.: Matching user with item set: Collaborative bundle recommendation with deep attention network. In: Proceedings of IJCAI, pp. 2095–2101 (2019)

[7] Chang, J., Gao, C., He, X., Jin, D., Li, Y.: Bundle recommendation with graph convolutional networks. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1673–1676 (2020)

[8] Zhao, S., Wei, W., Zou, D., Mao, X.: Multi-view intent disentangle graph networks for bundle recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 4379–4387 (2022)

[9] Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., Chua, T.-S.: Attentional factorization machines: Learning the weight of feature interactions via attention networks. Proceedings of IJCAI (2017)

[10] Li, Z., Cui, Z., Wu, S., Zhang, X., Wang, L.: Fi-gnn: Modeling feature interactions via graph neural networks for ctr prediction. In: Proceedings of CIKM, pp. 539–548 (2019)

[11] He, X., Chua, T.-S.: Neural factorization machines for sparse predictive analytics. In: Proceedings of SIGIR, pp. 355–364 (2017)

[12] Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: a factorization-machine based neural network for ctr prediction. Proceedings of IJCAI (2017)

[13] Yan, X., Yu, P.S., Han, J.: Substructure similarity search in graph databases. In: Proceedings of SIGMOD, pp. 766–777 (2005)

[14] Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for

business process model similarity search. In: International Conference on Business Process Management, pp. 48–63 (2009). Springer

[15] Berretti, S., Del Bimbo, A., Vicario, E.: Efficient matching and indexing of graph models in content-based retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(10), 1089–1105 (2001)

[16] Srinivasa, S., Kumar, S.: A platform based on the multi-dimensional data model for analysis of bio-molecular structures. In: Proceedings of VLDB, pp. 975–986 (2003). Elsevier

[17] Raymond, J.W., Gardiner, E.J., Willett, P.: Rascal: Calculation of graph similarity using maximum common edge subgraphs. The Computer Journal **45**(6), 631–644 (2002)

[18] Willett, P., Barnard, J.M., Downs, G.M.: Chemical similarity searching. Journal of chemical information and computer sciences **38**(6), 983–996 (1998)

[19] Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: Proceedings of ICML, pp. 321–328 (2003)

[20] Vishwanathan, S.V.N., Schraudolph, N.N., Kondor, R., Borgwardt, K.M.: Graph kernels. Journal of Machine Learning Research **11**, 1201–1242 (2010)

[21] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? Proceedings of ICLR (2019)

[22] Leman, A., Weisfeiler, B.: A reduction of a graph to a canonical form and an algebra arising during this reduction. Nauchno-Technicheskaya Informatsiya **2**(9), 12–16 (1968)

[23] Bai, Y., Ding, H., Bian, S., Chen, T., Sun, Y., Wang, W.: Simgnn: A neural network approach to fast graph similarity computation. In: Proceedings of WSDM, pp. 384–392 (2019)

[24] Li, Y., Gu, C., Dullien, T., Vinyals, O., Kohli, P.: Graph matching networks for learning the similarity of graph structured objects. In: International Conference on Machine Learning, pp. 3835–3845 (2019). Proceedings of PMLR

[25] Su, Y., Zhang, R., Erfani, S., Gan, J.: Neural graph matching based collaborative filtering. Proceedings of SIGIR (2021)

[26] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Proceedings of ICLR (2013)

[27] Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., *et al.*: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for

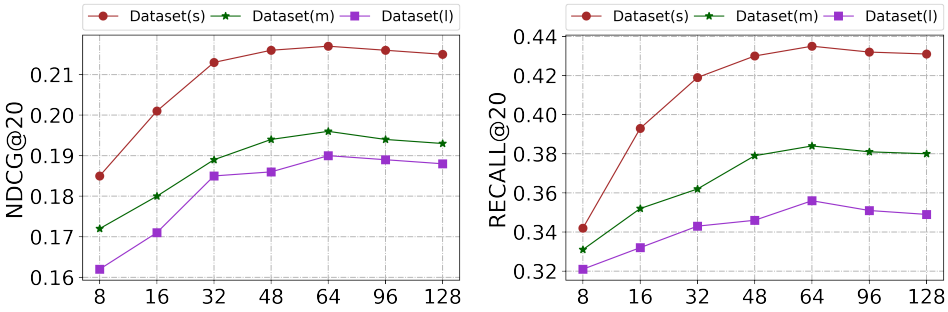
Recommender Systems, pp. 7–10 (2016)

[28] Yu, X., Ren, X., Gu, Q., Sun, Y., Han, J.: Collaborative filtering with entity similarity regularization in heterogeneous information networks. Proceedings of IJCAI **27** (2013)

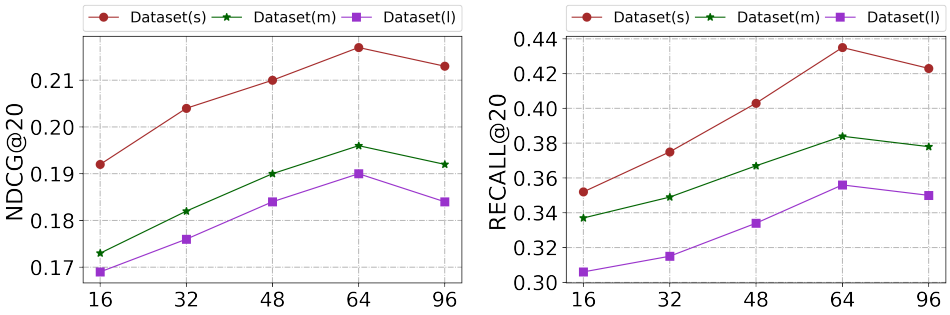
[29] Shi, C., Hu, B., Zhao, W.X., Philip, S.Y.: Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering **31**(2), 357–370 (2018)

[30] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010). JMLR Workshop and Conference Proceedings

[31] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. Proceedings of ICLR (2015)

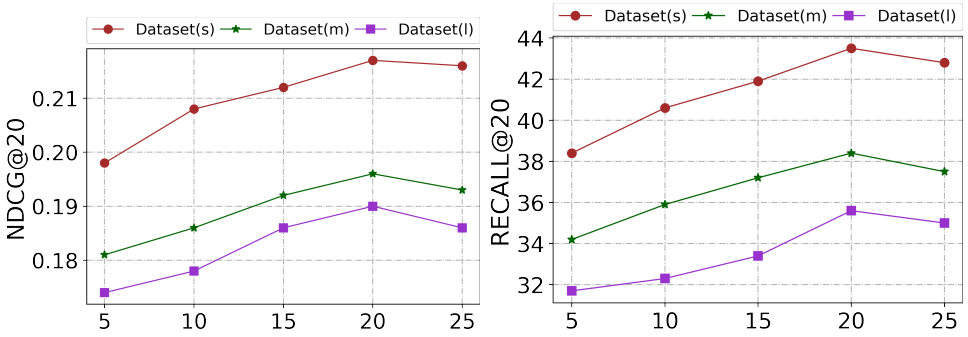


(a) # embedding dimension

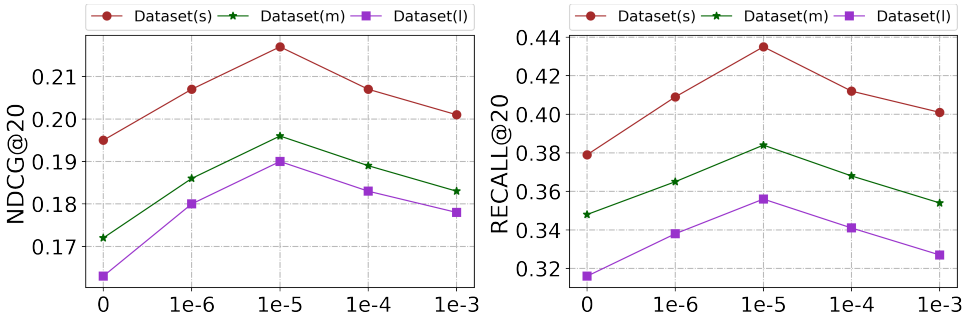


(b) # batch size

Fig. 4: Performance under different hyperparameters(i).



(a) # item numbers



(b) # regularization weight

Fig. 5: Performance under different hyperparameters(ii).