

Exploiting Group Information for Personalized Recommendation with Graph Neural Networks

ZHIQIANG TIAN, School of Management, Hefei University of Technology, China

YEZHENG LIU, School of Management, Hefei University of Technology and National Engineering Laboratory for Big Data Distribution and Exchange Technologies, China

JIANSHAN SUN and YUANCHUN JIANG, School of Management, Hefei University of Technology and Key Laboratory of Process Optimization and Intelligent Decision-making, China

MINGYUE ZHU, School of Management, Hefei University of Technology, China

Personalized recommendation has become more and more important for users to quickly find relevant items. The key issue of the recommender system is how to model user preferences. Previous work mostly employed user historical data to learn users' preferences, but faced with the data sparsity problem. The prevalence of online social networks promotes increasing online discussion groups, and users in the same group often have similar interests and preferences. Therefore, it is necessary to integrate group information for personalized recommendation. The existing work on group-information-enhanced recommender systems mainly relies on the item information related to the group, which is not expressive enough to capture the complicated preference dependency relationships between group users and the target user. In this article, we solve the problem with the graph neural networks. Specifically, the relationship between users and items, the item preferences of groups, and the groups that users participate in are constructed as bipartite graphs, respectively, and the user preferences for items are learned end to end through the graph neural network. The experimental results on the Last.fm and Douban Movie datasets show that considering group preferences can improve the recommendation performance and demonstrate the superiority on sparse users compared

CCS Concepts: • CCS Information systems → Recommender systems;

Additional Key Words and Phrases: Personalized recommendation, graph neural network, group preferences

This work is supported by the Major Program of the National Natural Science Foundation of China (91846201), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (71521001), the National Natural Science Foundation of China (71872060, 71722010, 91746302, 72071069, 71801069, and 71802068), the Fundamental Research Funds for the Central Universities of China (JZ2020HGPA0113), and the National Engineering Laboratory for Big Data Distribution and Exchange Technologies.

Authors' addresses: Z. Tian and M. Zhu, Hefei University of Technology, TunXi Road 193, Hefei, China; emails: tianzhiqiang2015@mail.hfut.edu.cn, zhuyang2737@163.com; Y. Liu, Hefei University of Technology, TunXi road 193, Hefei, National Engineering Laboratory for Big Data Distribution and Exchange Technologies, China; email: liuyezheng@hfut.edu.cn; J. Sun (corresponding author) and Y. Jiang (corresponding author), Hefei University of Technology, TunXi road 193, Hefei, Key Laboratory of Process Optimization and Intelligent Decision-making, China; emails: {sunjs9413, ycjiang}@hfut.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

1046-8188/2021/09-ART27 \$15.00

<https://doi.org/10.1145/3464764>

ACM Reference format:

Zhiqiang Tian, Yezheng Liu, Jianshan Sun, Yuanchun Jiang, and Mingyue Zhu. 2021. Exploiting Group Information for Personalized Recommendation with Graph Neural Networks. *ACM Trans. Inf. Syst.* 40, 2, Article 27 (September 2021), 23 pages.

<https://doi.org/10.1145/3464764>

1 INTRODUCTION

With the development of the Internet, users have entered an era of information explosion. As an important means to solve information overload [14, 41], recommender systems have been designed to tailor relevant items for users. However, traditional recommender systems are easily affected by the data sparsity problem since they mainly employ historical interaction data between user and items. The prevalence of online social networks promotes increasing online discussion groups, and users are often passively or actively engaged in these groups from different online communities, such as Baidu Tieba¹ and Douban Group.² The formation of these groups may be due to common cultural backgrounds, similar interest preferences, event discussions, and so forth. Although the reasons for the group composition are different, these groups often show the common preferences of participating users in a certain way.

When users have participated in groups, they interact with groups and their preferences are affected by groups. Shaw defines a group as a set of at least two people who influence each other [49]. People with the same interests are often more likely to form groups. For each user in the society, his or her individual preferences will also be affected by the group he or she joins [44]. Group identity theory can explain the reason for this behavior and it is developed from the theory of social identity [19]. Group identity represents a user's sense of identity to a specific group, and the group often brings specific values and meanings to the group members. Once a person regards himself or herself as part of a group, he or she will identify with the values of the group, gain self-esteem from having group membership, and adopt behavior consistent with the group [9]. Therefore, integrating the preferences of the user's groups can better model the user's preferences, thereby better achieving personalized recommendations.

Most of the existing personalized recommendation methods are based on collaborative filtering, which can be divided into two categories: neighborhood-based collaborative filtering methods [26, 53] and model-based collaborative filtering methods [27, 43]. Most of these studies did not consider the influence of group information on users, and only limited work used group preferences to improve personalized recommendation results [11, 38]. These limited works only employed the item information directly related to the group. In online communities, the relationships between groups and individuals are complex and mutually influential. On the one hand, different individual users form groups due to their similar interests and preferences, so group preferences are reflected in the aggregated individual user's preferences. The preference differences between groups are reflected in their different group compositions. This leads to the difficulty of how to use different user preferences to construct group preferences. On the other hand, individual users participate in a varying number of groups that reflect different aspects of preferences. When using group preferences to recommend new products for individual users, it is important to consider not only the influence of group preferences but also the influence of the preferences of other users in the group. For example, the same group of fans of science fiction movies may also have users who like comedy or action movies. Users with different preferences in the same group also have an

¹<https://tieba.baidu.com>.

²<https://www.douban.com/group>.

impact on the target user. This effect can be represented by the higher-level connectivity of nodes in graph structured data.

Existing group-information-enhanced recommendation methods mainly rely on the item information related to the group, which is not expressive enough to capture the complicated preference dependency relationships between group users and the target user. In this article, we solve the problem with the graph neural networks. In recent years, with the success of graph neural networks [24, 51] in graph structure data, researchers have begun to study the relationship between users and items as a bipartite graph [2, 16, 52] and thus realize the high-order connectivity of different users and items. Considering that the influence mechanism of users in the group on the target user corresponds to the high-order connectivity of the graph, we propose a **group-preference-enhanced graph neural recommendation model (GGRM)**. The model is a graph model based on deep learning, which constructs the relationship between users and items, item preferences of groups, and groups of users participating respectively into the form of a bipartite graph. Based on the graph neural network, the features of users and items are modeled from the perspective of individuals and groups, which not only alleviates the problem of data sparsity but also realizes sufficient modeling of user preferences and item features.

In sum, the main contributions of this article are as follows:

(1) This article not only uses the user's historical behavior to model the user's preferences but also aggregates the preferences of all groups in which the user participates. Different from the previous work of fusing group information for personalized recommendation, this article uses a deep graph neural network to model user preferences and group preferences. It not only aggregates item information directly related to the group but also models the influence of other users in the group on the target user.

(2) In order to verify the effectiveness of the GGRM, this article compares it with a variety of mainstream and classic recommendation models. Experimental results on the two real-world datasets of Last.fm and Douban Movie show that the GGRM is better than the state-of-the-art recommender methods. At the same time, the experimental results on different recommended list lengths and different latent factor vector dimensions show the robustness of the GGRM. It is worth noting that the GGRM shows more excellent results when facing more sparse users. This shows that the GGRM alleviates the sparsity problem in the implicit feedback recommendation.

The rest of the article is organized as follows: In Section 2, the related work is reviewed. The preliminaries and the proposed model are described in detail in Section 3. In Sections 4 and 5, extensive experiments are conducted and the results are analyzed in detail. Finally, the full article is summarized in Section 6.

2 RELATED WORK

2.1 Personalized Recommendation Based on Collaborative Filtering

An important goal of personalized recommendation methods is to tailor personalized products or items for users. Collaborative filtering [14] assumes that people with similar behaviors have similar preferences. Personalized recommendation methods based on collaborative filtering can be divided into neighborhood-based collaborative filtering methods and model-based collaborative filtering methods. Neighborhood-based collaborative filtering methods [1] also include two well-known collaborative recommendation methods: user-based collaborative filtering [20, 26] and item-based collaborative filtering [10, 31].

The model-based collaborative filtering methods mainly use machine learning techniques to model the relationships between users and items. Typical methods include Bayesian network-based models [23], matrix factorization models [27], and graph models [21]. Among them, the

matrix factorization model, as a typical latent factor model in the field of recommender systems, has been widely studied and applied due to its good performance [4, 18, 39]. The matrix factorization model decomposes the co-occurrence matrix of users and items into the user's latent feature matrix and the item's latent feature matrix, so that each user and item are respectively represented by a low-dimensional vector. In the prediction process, the product of the item of the target user's vector and the target item's vector is used as the target user's score on the target item. Another well-known matrix factorization model is **Probabilistic Matrix Factorization (PMF)** [43], which assumes that the user's latent feature vector and the item's latent feature vector obey the Gaussian prior distribution, and the user's score on the item also obeys the Gaussian distribution. Finally, the latent feature matrix of user and item is solved by the gradient descent method. For the implicit feedback problem, the most well-known method is **Bayesian Personalized Ranking (BPR)** [40]. BPR regards the data appearing in implicit feedback data as the items observed by users and the data not appearing as the items not observed by users. Different from previous models, the BPR model does not regard the recommendation model as a classification problem but assumes that users' preference for observed items is greater than that for unobserved items, which is more in line with the ranking situation.

In recent years, the deep learning models have been employed in the field of recommender systems [60]. The AutoRec model [47] combines the advantages of collaborative filtering and the auto-encoder model, takes the user (item) score as the input of the auto-encoder model, and reconstructs the input of the model through a hidden layer of neural network. On this basis, Wu et al. used a denoising auto-encoder to construct a recommendation model for the Top-N recommendation problem [56], and Liang et al. proposed the use of a variational autoencoder to overcome the linear factor model on the implicit feedback problem [30]. At the same time, He et al. [17] built a **general matrix factorization (GMF)** model with the help of the embedding representation in the neural network, which passes the latent factor matrix of users and products in the matrix decomposition through the embedding layer of the neural network for modeling. In comparison, Xue et al. [57] took into account the different effects of different products on the target product and used the attention mechanism to assign different weights to historically interactive products.

2.2 Personalized Recommendation Incorporating User Social Relationship

Users' purchase decisions are often not only based on their own preferences but also influenced by the information and suggestions provided by their social relations [48]. Research on social networks also finds that homophily (two users with social relations have certain similarity in behaviors) is an important principle in social relations [28, 35]. Therefore, researchers have begun to model users' social relationships to improve the effectiveness of the recommendation system [7, 34, 36]. At the same time, the use of social relations can alleviate the sparsity problem of the recommender systems.

Recent studies have proved that trust is strongly correlated with the homophily of social relations [13, 62]. Jamali and Ester proposed a TrustWalker algorithm to predict users' ratings of items [22]. The algorithm used random walk to add the trust value between users to the rating prediction model. It is a mainstream idea to integrate the trust relationship between users into the matrix factorization model. There are two main methods in this aspect. One is to add the social relationship between users into the matrix factorization method as a regular term, such as the SoRec algorithm [33]. Another way is to weighted-average the preferences of users and users' friends, such as the STE model [32]. Furthermore, Krohn-Grimberghe et al. [29] considered the social relations of users into the BPR model and used the matrix factorization model to jointly decompose, so as to improve the effect of the model on sparse users. Zhao et al. proposed the SBPR model based on the assumption that the products liked by users were better than those liked by friends,

which were better than those disliked by both users and friends [61]. With the development of deep learning, especially the graph neural network, researchers have begun to use the graph neural network to model the relationship between users. Wu et al. proposed the CNSR model [54] to integrate users' social network embedding into the neural network recommendation model and design an efficient joint learning algorithm. The experimental results showed that the model was superior to benchmark algorithms. The DGR model [55] was also proposed to model the social relationships and it employed graph attention networks to adaptively generate different weights to learn the influence of friends on the user's behavior.

In addition to the friend relationships directly related to users, users' social relations also include the online groups and communities that users participate in. These groups are important places for users to communicate, and subdivided groups maintain intra-group relevance and inter-group heterogeneity to a certain extent. Previous studies have found that online communities and groups are very important for the development and activity of the platform [42]. Meanwhile, with the rapid development and popularization of social media, groups and communities have gradually become important traffic portals of network platforms [37]. Therefore, researchers have begun to study the influence of group preference on individual preference. Pan and Chen proposed a GBPR model integrating group preferences [38]. The experimental results showed that fusing group preference could help achieve better personalized recommendation results. After that, Gao et al. not only integrated the social relations of users but also used the real groups that users participated in and the preference of groups for events to build a better event recommendation model [11]. All of these works have proved that group preference affects users' personality preference. Furthermore, because of the interaction between group and individual preferences, many models used for group recommendation tasks can also perform the task of personalized recommendation. For example, the AGREE model [5] proposed by Cao et al. can also perform personalized recommendation tasks using group information of user participation. Based on the AGREE model, Cao et al. [6] used social follower data to enhance the representation of users in a group and proposed the SoAGREE model to improve the effectiveness of recommendations.

2.3 Graph Neural Network

This work uses graph neural networks to model users, products, and groups. Therefore, in this section, we review the works of graph neural networks and their applications in recommendation systems.

Although deep learning models have brought huge improvements in the fields of computer vision and natural language processing, convolutional neural networks and recurrent neural networks can only process Euclidean spatial data. Graph data, a kind of non-Euclidean spatial data, has gradually attracted attention because of its universal existence. Graph data can naturally express data structures in real life, such as transportation networks, the World Wide Web, and social networks. In recent years, due to the ubiquity of graph data, researchers have begun to pay attention to how to construct deep learning models on graphs. Research on graph neural networks has become the most active and important research direction. Inspired by graph convolutional neural networks, Bruna et al. [3] proposed the first graph convolutional neural network. They started from the convolution theorem based on graph theory and defined graph convolution in the spectral space. In order to make the graph convolutional neural network work in the field of semi-supervised learning on the graph, Kipf et al. [24] simplified the Chebyshev network and proposed a first-order graph convolutional neural network. The **personalized propagation of neural predictions (PPNP)** based on personalized PageRank pointed out that as the number of model layers deepens, the network fitting ability increases, but the first-order graph convolutional neural network causes the node expression to be too smooth, which led to the indistinguishable

Table 1. Selected Works for Personalized Recommendations

Year	Model	Individual Preference	Group Preference	Basic Model			Fusion Strategy
				MF	NN	GNN	
2007	PMF [43]	✓		✓			
2009	BPR [40]	✓			✓		
2013	GBPR [38]	✓		✓	✓		Weighted Sum
2015	AutoRec [47]	✓				✓	
2017	NCF [17]	✓				✓	
2017	GCMC [59]	✓				✓	
2018	AGREE [5]	✓		✓		✓	Attention Model
2018	GraphSage [15]	✓				✓	
2019	DeepICF [57]	✓				✓	
2019	SoAgree [6]	✓		✓		✓	Attention Model
2019	NCCF [52]	✓				✓	
2020	LightGCN [16]	✓				✓	
	GGRM	✓		✓		✓	Graph Neural Network

nodes problem. Based on this, PPNP decoupled the dimensional transformation and feature propagation and introduced a personalized PageRank, first completing a small number of dimensional transformations on the input data and then performing feature propagation based on the personalized PageRank [25]. In a large-scale network, even if only considering second-order neighbors, the amount of calculation was too large. GraphSAGE [15] used a batch training method, combined with random sampling of neighbor nodes, to control the number of nodes required for each calculation within a certain range, making distributed training of large-scale graph data possible. When aggregating features of neighbor nodes, the **graph attention network (GAT)** [51] used the attention mechanism to determine the weight of each neighbor node in the node, and it can adaptively control the contribution of neighbor nodes to the target node. In addition to the innovation of the network model, the definition of the general framework in the graph neural network is a core issue. The **Message Passing Neural Network (MPNN)** [12] is based on the information dissemination and aggregation between nodes, and the framework is proposed by defining the general form of the aggregation function. The representation vector of the defined node is obtained after multiple rounds of message propagation through the message function and update function.

The graph neural network has been proven to be able to model the structural attributes and node feature information of the graph very well, and the recommender system can be regarded as a link prediction problem of bipartite graphs (users and items). Therefore, in recent years, researchers have begun to apply graph neural networks to the field of recommendation systems. Berg et al. [2] proposed the GCMC model, which regards the mid-matrix completion of the recommendation system as a bipartite graph link prediction problem, and used an end-to-end graph auto-encoder for modeling. Based on the high-level connectivity issues of users and products, Wang et al. proposed the NGCF model [52]. Based on the GraGraphSAGE model, Ying et al. proposed a data-efficient graph convolutional neural network algorithm PinSage. Compared with the traditional graph convolution method, it proposed an efficient random walk strategy modeling convolution and successfully applied the graph neural network to the super-large-scale recommendation system with 1 billion nodes [58].

Inspired by the work [45], we review previous important personalized recommendation works and summarize their employed information and basic models in Table 1. “NN” is the abbreviation of neural network and “GNN” is the abbreviation of graph neural network.

As can be seen from Table 1, neural networks, especially graph neural networks, are the mainstream models used for personalized recommendation tasks in recent years. In terms of fusing

Table 2. List of Used Notations

Notation	Description
u, i, c	User, item and group
U, I, C	User set, item set, and group set
M, N, T	The numbers of users, items, and groups
y_{ui}, \hat{y}_{ui}	The user u 's true preference value and model prediction value of product i
G^{UI}, G^{CI}, G^{UC}	User-product, group-product, and user-group bipartite graph
A^{UI}, A^{CI}, A^{UC}	The adjacency matrix of user-product, group-product, and user-group
$P^U \in \mathbb{R}^{M \times K}, P^I \in \mathbb{R}^{N \times K}, P^C \in \mathbb{R}^{T \times K}$	The latent feature matrix of users, items, and groups
$e_u \in \mathbb{R}^K, e_i \in \mathbb{R}^K, e_c \in \mathbb{R}^K$	The latent feature vector of user u , item i , and group c
K	The dimension of latent feature
D	The depth of graph neural network

users' social friends, research work has been done to demonstrate that graph neural networks can better model the interactions between users compared to traditional methods. Unfortunately, despite the work on using group preferences to improve personalized recommendations, however, their influence between individual and group preferences has not been modeled in depth. The ability of graph neural networks to more deeply model the influence of correlations between nodes is precisely why this work chose to use graph neural networks to fuse group preferences and individual preferences.

3 PROPOSED MODEL

3.1 Preliminaries

In this section, we first introduce a formal introduction to the problem solved by the GGRM. According to the analysis in the first section, a user's individual preferences are affected not only by his or her own historical behavior but also by the preferences of the participating groups. Therefore, when the GGRM generates a recommendation list for users, it not only models the user's individual preferences but also aggregates the group preferences of the user's participation. Formally, there are user $u \in U$, products $i \in I$, and groups $c \in C$, and the GGRM models them to predict users' preferences for products. The mathematical notations used in this article and their descriptions are listed in Table 2.

In order to better characterize the relationship between users, products, and groups, first construct a user-product bipartite graph G^{UI} , a group-product bipartite graph G^{CI} , and a user-group bipartite graph G^{UC} , as shown in Figure 1. User-product bipartite graph G^{UI} contains all the relationships between users and products, and A^{UI} is the adjacency matrix of G^{UI} :

$$A_{ui}^{UI} = \begin{cases} 1, & \text{if user } u \text{ like product } i \\ 0, & \text{else.} \end{cases} \quad (1)$$

User-product bipartite graph G^{CI} contains all the relationships between users and products, and A^{CI} is the adjacency matrix of G^{CI} :

$$A_{ci}^{CI} = \begin{cases} 1, & \text{if group } c \text{ like product } i \\ 0, & \text{else.} \end{cases} \quad (2)$$

User-product bipartite graph G^{UC} contains all the relationships between users and products, and A^{UC} is the adjacency matrix of G^{UC} :

$$A_{uc}^{UC} = \begin{cases} 1, & \text{if user } u \text{ in group } c \\ 0, & \text{else.} \end{cases} \quad (3)$$

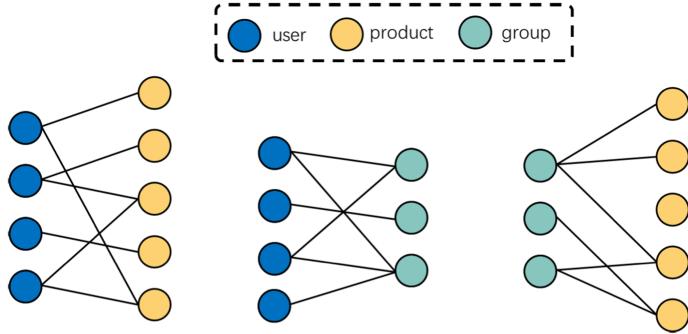


Fig. 1. The diagram of bipartite graph construction.

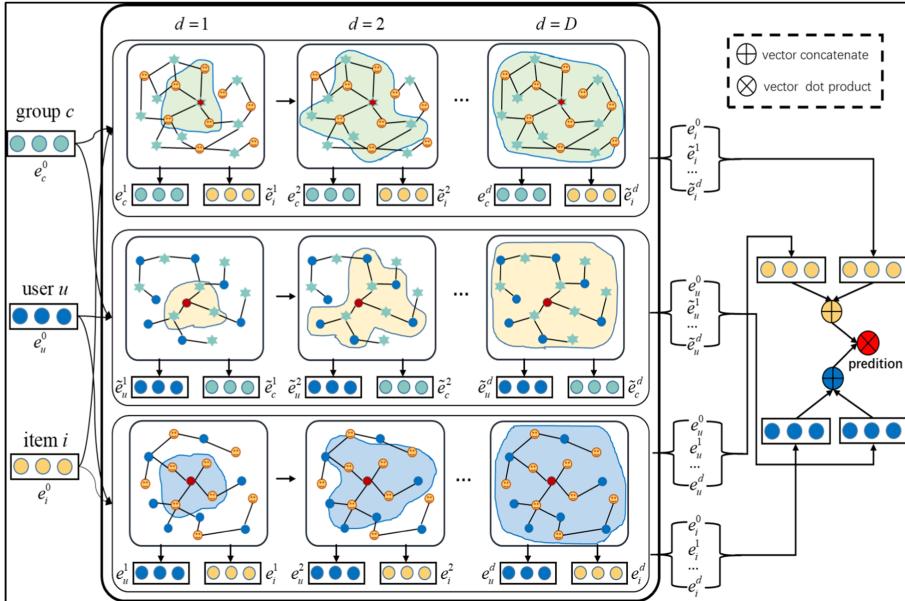


Fig. 2. An illustration of GGRM architecture.

3.2 GGRM

Previous studies have proved that graph neural networks can better mine the structure and content of bipartite graphs. Accordingly, we use graph neural networks to simultaneously model users' individual preferences and group preferences to provide users with more accurate recommendation lists. This chapter proposes the GGRM. The model structure is shown in Figure 2. In Figure 2, the dark blue circle represents the user, the yellow smiley face represents the product, the green hexagon represents the group, and the boxes represent the hidden feature representation vectors of the three. The GGRM can be roughly divided into three parts: the embedding layer maps users, products, and groups into low-dimensional dense vector representations; the graph neural network aggregates the feature of users, products, and groups by learning high-level connections in the bipartite graph; and the predictions layer aggregates the user's individual preferences and group preferences and outputs the user's preference score for the product. The core of the GGRM is the

graph neural network part. As shown in Figure 2, there are a total of three bipartite graphs from top to bottom as input to the model. They are the group-product bipartite graph, the user-group bipartite graph, and the user-product graphs. These three graphs model by graph neural network to learn the representation of users, products, and groups, respectively. Next, this section starts with the input of the model and introduces the modeling process of the GGRM in detail.

3.2.1 Embedding Layer. Like the LightGCN model, the role of the GGRM embedding layer is to map users, products, and groups into low-dimensional dense vector representations. Take the user as an example. The input of the GGRM is the user's index. If it is one-hot encoded, a high-dimensional and extremely sparse vector will be generated. This is to increase the complexity of model calculations and at the same time encounter the problem of dimensional disaster. Similar to the matrix factorization model, the embedding layer also uses a low-dimensional dense vector to represent users, which not only greatly reduces the computational complexity of the model, but also the learned user representation vector can calculate the similarity between users. It also means that the embedding layer has learned a certain degree of relationship characteristics. In the GGRM, $P^U \in R^{M \times K}$, $P^I \in R^{N \times K}$, $P^C \in R^{T \times K}$ denote the embedding matrix of users, products, and groups (the parameters of the matrix are updated with the learning of the model), respectively. Using the user index, the latent feature factor vector $e_u^0 \in R^K$ of the user u can be obtained. Similarly, through the product and group index, $e_i^0 \in R^K$, $e_c^0 \in R^K$ are the latent factor vectors of the product and the group and can be obtained respectively, where K represents the dimension of the latent factor vector.

3.2.2 User-Product Bipartite Graph Modeling. The products consumers like directly reflect their preferences. Intuitively, two consumer owners who like the same product have a certain similarity. Therefore, if most of the same products exist in the historical consumption data of two users, then the preferences of the two users are also similar. Similarly, two different products with a common user should have the same characteristics. This historical relationship between the two interactions can be modeled by mining the user-product bipartite relationship. Graph convolutional networks have achieved remarkable results on graph structure data. Similarly, the LightGCN model [16] based on graph convolutional neural networks uses graph convolutional networks to model the relationship between users and products and has achieved the best result in the current personal recommendation task. The GGRM models the user-product bipartite graph based on the LightGCN model. The modeling of graph neural networks is a process of message propagation and aggregation. In the GGRM the convolution operation on the relationship and characteristics of users and products is defined as follows:

$$\begin{aligned} e_u^{(d+1)} &= \sum_{i \in N_u^d} \frac{1}{\sqrt{|N_u^d|} \sqrt{|N_i^d|}} e_i^{(d)} \\ e_i^{(d+1)} &= \sum_{u \in N_i^d} \frac{1}{\sqrt{|N_u^d|} \sqrt{|N_i^d|}} e_u^{(d)}. \end{aligned} \quad (4)$$

In the above formula, $e_u^{(d)}$, $e_i^{(d)}$ denote the latent vectors of the user and product in the first-level convolution operation, respectively. d denotes the number of layers of the convolution operation, that is, the order in graph concept. Through d layer convolution, the neighbor features of the target vertex can be aggregated. N_u^d denotes the user's neighbors on the map and in the real world is a set of products that user u likes. Similarly, N_i^d denotes the set of users who like the product. As a normalization item, $1/\sqrt{|N_u^d|} \sqrt{|N_i^d|}$ follows the standard design of the graph convolutional network, and this normalization item also achieves good results on the recommendation system.

After the d layer convolution operation, the features of users and products at different levels can be obtained. At the same time, as the number of layers increases, the high-level feature will be smoother, which is not conducive to the recommendation task. Therefore, in order to fuse different features, the modeling input of the user-product bipartite graph is the fusion of different features of users (products). The formula for the end user and product output is as follows:

$$\begin{aligned} e_u &= \frac{1}{D} \sum_{d=0}^D e_u^{(d)} \\ e_i &= \frac{1}{D} \sum_{d=0}^D e_i^{(d)}. \end{aligned} \quad (5)$$

3.2.3 Group-Product Bipartite Graph Modeling. Similar to users, different groups also have group preferences. For example, Mr. Bean's fan groups mostly discuss Mr. Bean's films, while the content of the comedy film group discussion includes both Mr. Bean's comedy movies and movies of other comedians. These different groups generally have feature preferences due to their organizational principles. Groups with similar preferences also have similarities. For example, Mr. Bean's fan group and comedy movie group mentioned above have certain similar features. In order to maintain the unity of the GGRM structure, the graph convolutional network is also used when modeling the group-product bipartite graph, and the convolution operation on the relationship and characteristics of the group and the product is defined as follows:

$$\begin{aligned} e_c^{(d+1)} &= \sum_{i \in N_c^i} \frac{1}{\sqrt{|N_c^i|} \sqrt{|N_i^c|}} \tilde{e}_i^{(d)} \\ \tilde{e}_i^{(d+1)} &= \sum_{c \in N_i^c} \frac{1}{\sqrt{|N_c^i|} \sqrt{|N_i^c|}} e_c^{(d)} \\ \text{with, } \tilde{e}_i^{(0)} &= e_i^0. \end{aligned} \quad (6)$$

In the above formula, $e_c^{(d)}$ denotes the feature vector of the group after d layer convolution operation. In order to distinguish the difference between individual preference and group preference, we use $\tilde{e}_i^{(d)}$ to represent the latent vector of the product in the group preference during the first layer convolution operation. N_c^i denotes the set of products that the group c likes. Similarly, N_i^c denotes the set of groups that like the product i .

Similar to the user-product bipartite graph modeling, the group-product bipartite graph modeling groups and products obtain different levels of features. In order to integrate the characteristics of different levels, the output of the group-product bipartite graph is

$$\begin{aligned} e_c &= \frac{1}{D} \sum_{d=0}^D e_c^{(d)} \\ \tilde{e}_i &= \frac{1}{D} \sum_{d=0}^D \tilde{e}_i^{(d)}. \end{aligned} \quad (7)$$

3.2.4 User-Group Bipartite Graph Modeling. Consumers do not exist independently on the Internet; they will actively or passively participate in the Internet community. Similarly, a user may participate in different groups. These groups all explicitly reflect the interest preferences of users to a certain extent. Different users participating in the same group have interest tags marked by

this group, which is very useful for modeling user representation. When the GGRM uses the collectively participating groups to model users and also uses graph convolutional networks to act on the user-group bipartite graph, the convolution operation on the relationship and characteristics of users and groups is defined as follows:

$$\begin{aligned}\tilde{e}_u^{(d+1)} &= \sum_{c \in N_u^c} \frac{1}{\sqrt{|N_u^c|} \sqrt{|N_c^u|}} \tilde{e}_c^{(d)} \\ e_c^{(d+1)} &= \sum_{u \in N_c^u} \frac{1}{\sqrt{|N_u^c|} \sqrt{|N_c^u|}} e_u^{(d)} \\ \text{with, } \tilde{e}_u^{(0)} &= e_u^0.\end{aligned}\tag{8}$$

In the above formula, $e_u^{(d)}$, $\tilde{e}_c^{(d)}$ respectively represent the feature vectors of the user and the group after d layer convolution operation. N_u^c denotes the user's neighbors on the graph, that is, the groups in which user u participated. Similarly, N_c^u denotes the set of all users in the group.

On the user-group bipartite graph, features at different levels are also aggregated as the output of the user-group bipartite graph:

$$\begin{aligned}\tilde{e}_u &= \frac{1}{D} \sum_{d=0}^D \tilde{e}_u^{(d)} \\ e_c &= \frac{1}{D} \sum_{d=0}^D e_c^{(d)}.\end{aligned}\tag{9}$$

3.2.5 Prediction Layer. The GGRM builds the user's latent vector and product latent vector through graph convolutional network modeling of users' historical data and builds the group latent vector and product latent vector through a group-product bipartite graph. The participating groups aggregate the preferences of the user participating groups. In the final prediction, the GGRM aggregates the preferences of individual users and the group preferences in which users participate to jointly predict users' preferences for products. The prediction formula is defined as follows:

$$\hat{y}_{ui} = (e_u \oplus \tilde{e}_u) \cdot (e_i \oplus \tilde{e}_i)^T,\tag{10}$$

where \hat{y}_{ui} represents the GGRM predicting the user's preference score for the product i , and \oplus represents the splicing operation of two vectors.

3.2.6 Model Learning. In the implicit feedback problem, the only data that exists is the product data that users like. Products that the user has not observed are not necessarily products that the user does not like and may be those that the user has not yet browsed. Based on this, the GGRM chooses to compare the learning ranking method to construct the loss function of the GGRM. The basic idea is that the existing user's preference for products is greater than the unobserved products. The mathematical definition of the loss function is as follows:

$$L = - \sum_{u=1}^M \sum_{i \in N_u} \sum_{j \in N_u} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\Theta\|^2,\tag{11}$$

where L represents the overall loss function value of the model, and λ is the intensity of regularization of model parameters. It should be noted that the parameters of the GGRM are only the embedded matrix parameters of users, products, and groups; that is, compared with matrix decomposition, only one more group matrix vector parameter needs to be learned. Therefore, the computational complexity of the GGRM is basically equivalent to matrix decomposition. In the actual

training of the model, the small batch Adam optimization method is used to update the parameters of the learning model. Compared with other gradient descent methods, the Adam method has the advantages of small memory footprint and high computational efficiency and is very suitable for large-scale data environments.

3.3 Model Analysis

To present the powerfulness and uniqueness of the proposed GGRM, we discuss the relationships between GGRM and two similar related works (LightGCN [16] and **Bundle Graph Convolutional Network (BGCN)** [8]). The LightGCN model is a SOFT personalized recommendation model based on graph neural networks. However, the LightGCN model only utilizes user-product interaction data without incorporating other auxiliary information. Based on the LightGCN model, our work proposes a GGRM that incorporates group preferences, taking into account the influence of the group preferences that users are in. BGCN is a graph neural network model specifically designed for the bundle recommendation task. Although both BGCN and GGRM are graph neural network models incorporating group information, the two models do not address the same problem. The GGRM is designed to better portray the impact of personalized preferences of the group preferences that the user is in. Furthermore, in the model design, the input of the GGRM is three bipartite graphs between three entities (users, groups, and products), while the input of the BGCN model is a heterogeneous graph fusing the three.

Then, we discuss the matrix form of the GGRM and its complexity analysis. In the training process of the GGRM, there are two parts in total that require computational resources. The first part is the update of the model parameters. The total number of parameters of the GGRM is only related to the number of users, products, and groups. The total number of parameters of the GGRM is $(M + N + T) \times K$, where K is the size of the embedding vector. Another part that consumes computational resources is the computational process of the graph neural network, and we give the implementation of this process in matrix form. Taking the user-group bipartite graph as an example, it is first transformed into the following form:

$$A = \begin{pmatrix} 0 & A_{uc}^{UC} \\ (A_{uc}^{UC})^T & 0 \end{pmatrix}. \quad (12)$$

The computational process in the graph neural network is as follows:

$$E^{(d+1)} = \left(W^{-\frac{1}{2}} A W^{-\frac{1}{2}} \right) E^{(d)}, \quad (13)$$

where W is a $(M + T) \times (M + T)$ diagonal matrix, $E^{(0)} \in R^{(M+T) \times K}$.

4 EXPERIMENTS

4.1 Datasets

We perform experiments on two real-world datasets of Last.fm [46] and Douban Movie [50]:

Last.fm is a well-known music community website. On this website, users can explore the latest music singles or the most popular music creators and label these music or artists. At the same time, users can create new groups and invite others to join or join existing groups to discuss and share favorite music. The Last.fm dataset collects user data of the Last.fm website in the first half of 2009. In the actual experiment operation, the group with only one user was filtered out, and then the data of 21,571 users corresponding to 3,997 groups were randomly selected. On average, each user participated in 2.99 groups. The experiment also removed the music which was marked less than 5 times (music quality is not high or new music), and kept 100,502 music, an average of 78.55 music marked by each user.

Table 3. Statistics of the Evaluation Datasets

Dataset	#User	#Item	#Records	#Groups	#Sparsity
Last.fm	21,571	100,502	1,694,352	3,997	99.92%
Douban Movie	12,845	12,677	1,068,275	2,753	99.34%

The Douban Movie dataset comes from a well-known online community in China: Douban. The Douban community allows users to rate their favorite movies, books, and music. It also organizes different groups to allow users to participate in group discussions. As one of the channels, Douban Movies' ratings and discussions on movies have become the most important source of movie reviews in the Chinese community. Since the Douban Moviedata set is scoring data, in order to adapt to the experimental environment of this chapter, it is transformed into implicit feedback data. The experiment only uses user scores greater than 3 as positive samples. Like the Last.fm dataset, groups with only one user and movies with fewer than five ratings are removed. In the end, 12,845 users' preferences for 12,677 movies were obtained. On average, one user liked 83.17 movies. At the same time, there were data on user participation in 2,753 groups, and each user participated in 44.38 groups on average.

The statistics of the two datasets are shown in Table 3.

4.2 Baselines

To prove the advantage of our proposed GGRM, we compare GGRM with the following representative baseline methods:

(1) BPRMF [40]: The **Bayesian Personalized Ranking (BPR)** model assumes that users' preferences for different products are independent of each other. The BPR model uses the user's preference for products as a sorting algorithm; that is, the user's rating of the product that he or she likes is greater than the rating of the product that he or she dislikes. This idea is very suitable for implicit feedback data.

(2) CMF [11]: This model uses the idea of collaborative matrix decomposition to integrate users' social preferences and group preferences into the recommendation model. To be fair, we only tested the group preference part of the model. This model calculates the group's product preference score and individual score and then balances the roles of the two in the recommendation model through parameters.

(3) NCF [17]: Neural Collaborative Filtering is the first to use the embedded layer of neural networks to construct the hidden feature matrix of users and products. At the same time, unlike the matrix factorization model that uses the inner product of two vectors to calculate the user's preference for products, the NCF model inputs the vectors to a multilayer neural network by splicing the vectors. Experimental results show that this method is more suitable for product ranking in recommendation.

(4) AGREE [5]: AGREE is a neural network model for group recommendation tasks using attention mechanisms. Although it is a group recommendation model, it can also be used for personalized recommendation tasks because it utilizes user-product interaction data. We use it as a personalized recommendation model using group information and report its experimental results.

(5) NGCF [52]: In the face of implicit feedback problems, Neural Graph Collaborative Filtering uses the user-product matrix as a bipartite graph to mining; considering that user products can establish relationships through high-level connections, NGCF uses the graph neural network to model the features of different depth graphs, so as to splice the features learned from different depths to predict the user's preference for products.

(6) LightGCN [16]: The LightGCN model is an improved version of the NGCF model. This model does not adopt the method of splicing the features of different depths but adopts the method of averaging. The experimental results show that this method not only improves the effect of the experiment but also saves a lot of model calculation time. Therefore, this is also the basis of the GGRM.

4.3 Evaluation Metrics

In the experimental phase, we use 20% of the products that each user likes as the test set and the remaining 80% as the training set. Since there are no negative samples of users in the implicit feedback problem, in the experiment, products are sampled from those users did not interact with and the group did not interact with as negative samples of the training set according to the ratio of positive and negative samples of 1:1. In the evaluation, all products that users have not interacted with and the products in the test set are used for prediction and ranking. For the metrics, we employ Recall and **Normalized Discounted Cumulative Gain (NDCG)** on the ranking of all non-purchased items for a user.

4.4 Experiments Setting

In the experiment, an important issue was to choose products preferred by the group. In order to distinguish between group preference and individual preference, in the specific experiments, we used the products preferred by more than two people in the group as the products that the group liked. On the other hand, including the benchmark algorithm and the GGRM proposed, there are many parameters that need to be adjusted, such as the dimension of the latent factor matrix. The experiment analyzes the effect of different dimensions in the experimental results. In addition, the NCF model is consistent with the original paper. The experiment uses a three-layer tower-shaped neural network structure to test the effects of different neural networks and show the best results. For the balance hyper-parameter value in CMF, the experiment adjusted different values, and the results showed that the effect was the best at 0.2. In addition, the other parameter settings of the GGRM are as follows: using the xvaier initialization method, the default learning rate is 0.01, and the regularization intensity is adjusted to 0.001; at the same time, the size of each small batch is set to 2,048, using the Adam algorithm default parameters. The GGRM is implemented by TensorFlow.

5 EXPERIMENTAL RESULTS AND ANALYSIS

5.1 Performance Comparison

In order to verify the effectiveness of the GGRM proposed in this work, we conducted comparative experiments with many benchmark algorithms. Among these algorithms, there are the classic personalized recommendation algorithms, BPRMF and CMF, and the deep-learning-based algorithms: NCF, NGCF, LightGCN, and GGRM. NGCF, LightGCN, and GGRM are all algorithms based on graph neural networks. At the same time, CMF, AGREE, and GGRM are models that integrate group preferences, while other models only consider individual preferences. Table 4 shows the experimental results of different comparison algorithms on the two real-world datasets Last.fm and Douban Movie. The “Improv” column focuses on comparing the effects of the GGRM model compared to the benchmark algorithms.

The following can be inferred from the results in Table 4:

(1) Models based on graph neural networks (NGCF, LightGCN, and GGRM) are better than other models, whether it is traditional models (BPRMF and CMF) or deep learning models (NCF, AGREE). This shows that compared with other latent factor recommendation models, the graph neural network can more accurately model the representation of users and products by modeling

Table 4. Comparison of Overall Performance among GGRM and Competing Methods
When the Length of Recommended List Is 10 and the Latent Vector Dimension Is 64

	Last.fm				Douban Movie			
	Recall	Improv	NDCG	Improv	Recall	Improv	NDCG	Improv
BPRMF	0.0511	29.59%	0.0516	24.69%	0.0997	28.89%	0.1327	19.45%
CMF	0.0552	19.96%	0.0539	19.29%	0.1081	18.85%	0.1340	18.31%
NCF	0.0310	113.46%	0.0297	116.68%	0.0959	33.97%	0.1059	49.68%
AGREE	0.0565	17.37%	0.0575	11.79%	0.1124	14.38%	0.1403	13.00%
NGCF	0.0542	22.19%	0.0528	21.77%	0.1067	20.45%	0.1367	15.93%
LightGCN	0.0568	16.73%	0.0587	9.56%	0.1127	14.01%	0.1491	6.33%
GGRM	0.0663	-	0.0643	-	0.1285	-	0.1585	-

high-level node connectivity, so as to achieve better results in personalized recommendation. This is why this work chooses the graph neural network to model three bipartite graphs.

(2) Second, among all the models, the CMF and AGREE model that integrates group preferences is better than the traditional BPRMF model, and even the CMF and AGREE model is better than the NGCF model on the Douban Movie dataset and slightly better than the LightGCN model. At the same time, in the recommendation model based on graph neural networks, the GGRM model fused with group preferences has achieved better results than both the NGCF model and the LightGCN model on the Last.fm and Douban Movie datasets. This also proves that the use of user-participated group information can help better model users' likes of the product.

(3) The two indicators of the GGRM on the two datasets are better than all benchmark algorithms, which shows that the GGRM based on graph convolution operation and fusion of group preferences is conducive to the construction of better user and product representations. Users provide a more accurate recommendation list. In addition, compared to the Douban Movie dataset, the effect of the GGRM on Last.fm is higher. The possible reason is that GGRM may be more effective in the case of sparse data. At the end of this section, the effect of the GGRM model on different types of users is analyzed in detail.

5.2 Sensitivity to Hyper-Parameter

In this part, we focus on the hyper-parameter investigation of the proposed methods on two datasets.

5.2.1 Recommended List Length. Generally speaking, as the length of the recommendation list increases, the more likely it is that personalized recommended products will cover user interests. In order to verify the robustness of the GGRM, we tested the performance of the GGRM and the comparison algorithm in different recommended list lengths. The experimental results of different models on the Last.fm dataset and the Douban Movie dataset are shown in Figure 3.

As can be seen from Figure 3, on both datasets, as the length of the recommendation list increases, all models perform better and better, which proves that increasing the length of the recommendation list can cover more user preferences. The GGRM proposed performs better than other model algorithms at different recommended list lengths. In addition, the performance difference in recall of BPRMF, CMF, AGREE, and NGCF is not obvious when the recommendation list is short, for example, when it is 10. On a longer recommendation list, the difference between these models will increase. At the same time, although the difference in ranking metrics among models is greater than the performance difference in recall metric, this indicates that partial order modeling is more suitable for ranking problems in recommendation systems. This is also the basis for

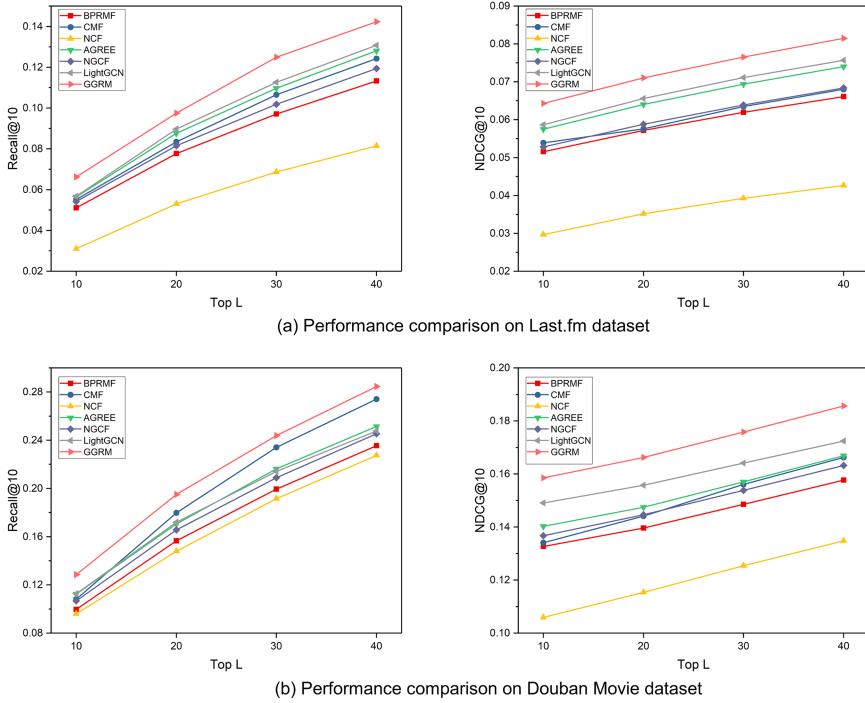


Fig. 3. Performance comparison over the different recommendation list lengths on different datasets.

the GGRM to design the loss function with partial order. The same conclusion can also be proved on the Douban Movie dataset.

5.2.2 The Dimension of Latent Factor Vector. The personalized recommendation algorithms compared in the experiment are all based on the latent factor model; that is, the users and products in the personalized recommendation system are represented by latent factor vectors. Therefore, in these models, the dimension of the latent factor vector is an important hyper-parameter. Generally speaking, the larger the hidden feature dimension, the greater the information that can be modeled and the more accurate the model. However, too large dimensions will also bring about overfitting problems. In order to measure the robustness of the GGRM, we simultaneously compared the performance in four different dimensions with the comparison algorithm. The result is shown in Figure 4.

As can be seen from Figure 4, whether it is on the last.fm or Douban Movie dataset, as the latent factor vector increases, the effect of most models becomes better. The only fluctuation is the NCF model with the performance of the NDCC metric on Douban Movie. Another point to note is that on the Last.fm model, when the dimensionality is small, the effect of CMF and AGREE is better than the NGCF and LightGCN models based on graph neural networks, but this phenomenon does not exist on the Douban Movie dataset. The results of the analysis may be related to the sparseness of the data. Compared with the Douban Movie dataset, the Last.fm model is more sparse and requires other information when the dimensionality is small. As the dimensionality increases, the LightGCN model can model more information and has also achieved better results. The specific analysis will be obtained below. But it is worth mentioning that no matter how many dimensions,

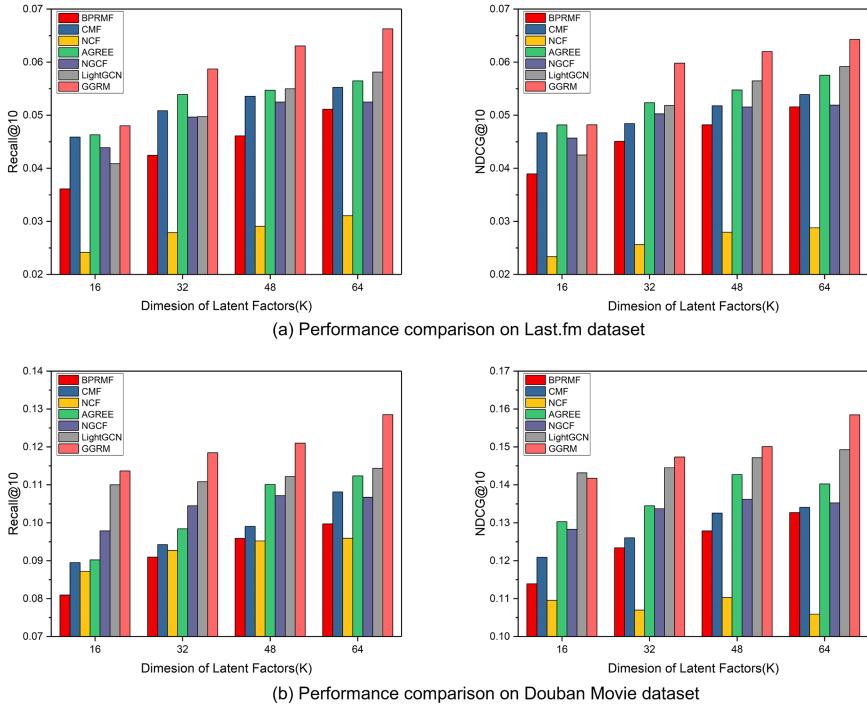


Fig. 4. Performance comparison over the different dimensions of the latent factor vector on different datasets.

the result of the GGRM model is the best among all models, which also proves the robustness of the GGRM model.

5.2.3 The Depth of Graph Neural Network. Although deep learning uses a deep neural network to learn the characteristics of the input, does the increase in depth definitely bring about an improvement in the effect, especially in graph structure data? In the GGRM model, the depth of the graph neural network is an important parameter. The one-layer graph neural network can aggregate the features of the first-order neighbors in the network, but when the depth increases, it may cause the problem of “over-smoothing.” Therefore, we tested the impact of different depths in the GGRM. The results are shown in Table 5. The “D” in Table 5 represents the depth of the graph neural network. Table 5 describes the results of different depths of the GGRM on the two datasets.

It can be seen that on the Last.fm dataset, the effect of the GGRM is the best (except for the Recall@20 indicator) when $D = 2$. The Douban Movie dataset is different; except for NDCG@40, the other indicators are the best at the time when $D = 3$. This shows that (1) the depth of the network can improve the effect, but the deeper graph neural network will damage the final result, and (2) for different datasets, different depths should be tested according to the different data structures. The influence of the model finds the optimal hyper-parameter value.

5.3 Result Analysis of Different Type Users

Based on the analysis in the previous section, this section analyzes the effect of GGRM on different types of users, especially users with different sparsity. Since there are two entities in the dataset, product and group, we use users on the two datasets to be divided into users with four roles,

Table 5. Performance Comparison over the Different Depths of Graph Neural Network on Different Datasets

D	Last.fm				Douban Movie			
	1	2	3	4	1	2	3	4
Recall@10	0.0558	0.0663	0.0617	0.0624	0.1126	0.1218	0.1285	0.1220
Recall@20	0.0895	0.0975	0.0953	0.0941	0.1711	0.1816	0.1950	0.1844
Recall@30	0.1073	0.1290	0.1188	0.1169	0.2191	0.2390	0.2438	0.2394
Recall@40	0.1245	0.1423	0.1378	0.1376	0.2569	0.2706	0.2845	0.2742
NDCG@10	0.0595	0.0643	0.0618	0.0614	0.1426	0.1529	0.1585	0.1526
NDCG@20	0.0629	0.0710	0.0681	0.0674	0.1448	0.1620	0.1662	0.1641
NDCG@30	0.0662	0.0765	0.0756	0.0748	0.1574	0.1717	0.1758	0.1730
NDCG@40	0.0725	0.0815	0.0787	0.0785	0.1691	0.1878	0.1857	0.1843

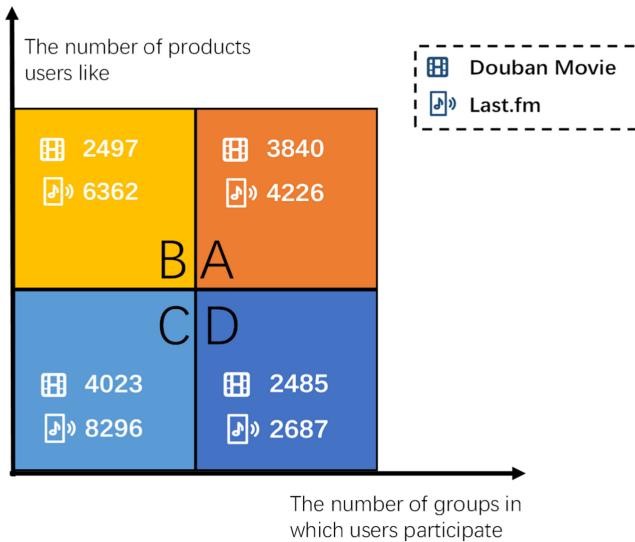


Fig. 5. The diagram of user division in different roles.

A, B, C, and D. The dividing standard is the median of the number of users who liked the product and the median of the participating groups. The two medians in the Last.fm dataset are 12 and 2, respectively, while the two numbers in the Douban Movie dataset are 10 and 27, respectively. The schematic diagram of the division is shown in Figure 5

The white numbers in Figure 5 represent the number of users with different roles. Among the four user roles, A and B types of user-representative historical data already contain a large amount of product preference data, and C and D users are the opposite; at the same time, A and D users indicate that users participate in many groups, and B and C do not. Through this division, the experiment measures the indicator effects for each user separately and then averages them by role. Table 6 shows the results of users with different roles on two different datasets of Last.fm and Douban Movie. The results are obtained under the experimental settings of latent factor dimension 64 and recommended list length 10. In order to be more clear about the influence of group preference, the last “Improv” row of each dataset in Table 6 indicates the improvement percentage of the GGRM relative to the LightGCN model.

Table 6. Performance Comparison over the Different Roles of Users on Different Datasets

	Last.fm							
	A		B		C		D	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPRMF	0.0584	0.0899	0.0625	0.0750	0.0419	0.0234	0.0411	0.0228
CMF	0.0607	0.0995	0.0614	0.0822	0.0501	0.0291	0.0479	0.0297
NCF	0.0322	0.0515	0.0352	0.0410	0.0293	0.0154	0.0249	0.0127
AGREE	0.0613	0.0995	0.0652	0.0822	0.0503	0.0291	0.0482	0.0297
NGCF	0.0595	0.0889	0.0645	0.0757	0.0456	0.0254	0.0482	0.0264
LightGCN	0.0616	0.1007	0.0674	0.0834	0.0491	0.0284	0.0478	0.0277
GGRM	0.0642	0.1049	0.0718	0.0868	0.0653	0.0364	0.0593	0.0326
Improv	4.24%	4.20%	6.52%	4.13%	33.10%	28.32%	23.93%	17.74%
	Douban Movie							
BPRMF	0.0979	0.2081	0.1200	0.2077	0.0795	0.0490	0.0802	0.0491
CMF	0.1019	0.2124	0.1258	0.2138	0.1024	0.0556	0.1096	0.0599
NCF	0.0740	0.1520	0.0962	0.1583	0.1088	0.0614	0.1056	0.0581
AGREE	0.1021	0.2119	0.1367	0.2214	0.1087	0.0662	0.1099	0.0707
NGCF	0.1035	0.2116	0.1297	0.2146	0.0992	0.0615	0.1005	0.0647
LightGCN	0.1109	0.2323	0.1396	0.2368	0.1010	0.0644	0.1074	0.0694
GGRM	0.1115	0.2341	0.1458	0.2449	0.1272	0.0779	0.1396	0.0852
Improv	0.47%	0.79%	4.45%	3.42%	25.84%	21.05%	30.01%	22.68%

The following can be seen from Table 6: (1) Compared with the LightGCN model, GGRM only has small improvements on category A and B users. In contrast, GGRM has achieved significant improvements on category C and D users. (2) Similarly, the effect of the CMF and AGREE models compared with the BPRMF model on the C and D users is more significant than that of the A and B users. From the above analysis, it can be concluded that the model that integrates group preferences has a more obvious effect on users with high sparseness, while the number of participating groups has little effect on the model. Therefore, in the actual personalized recommendation problem, when facing users who show less product preference, the groups that users participate in and the group's preference for products should be regarded as important personalized recommendation modeling information. When user activity increases and interactions with platform products increase, the weight of group preferences for personalized recommendation modeling can be reduced, and the user's personal historical data can be focused to create product recommendation lists.

6 CONCLUSIONS

This article first analyzes the influence of groups that the user participates in on the user's preference by using the theory of social identity and group identity, and then proposes a method to help model sparse users by using group preference. Specifically, this article proposes GGRM to model both individual and group preferences of users. When using the graph neural network to aggregate the preferences of various groups that users participate in, the GGRM model can not only aggregate the direct preferences of the group but also model the effects of other users of the group on the target users. The experimental results on two real-world datasets show that GGRM performs better than other baseline methods. Especially for users with sparse historical records, GGRM achieves better improvements. This shows that the use of group preferences can help make more effective recommendations to users with fewer historical records. This work not only proposes new personalized recommendation models and enriches the research on

personalized recommendation methods but also confirms that individual user preferences are influenced by group behavior from a data perspective. At the same time, the experimental results of this work can provide recommendation tools and design strategies for online community platforms. When facing new users, different groups should be actively promoted for users to participate in, so as to identify more accurate user preferences.

In future research work, we will extend our work in two directions: first, this work does not explicitly model users who are influenced by group preferences in varying degrees. In the real world, different users are influenced by group preferences differently; e.g., users with specialized knowledge are more likely to purchase products based on their own judgment, while users who are not knowledgeable about product performance are more likely to be influenced by group preferences. In the research of personalized recommendations, this phenomenon should be significantly taken into account in the modeling process. Second, the interplay of users and groups is not static but evolves dynamically. It is important to model the dynamic influence process of users and groups through a time-series model, which not only is helpful to improve the effectiveness of personalized recommendations but also can help to understand the complex relationship between users and groups.

REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749. <https://doi.org/10.1109/TKDE.2005.99>
- [2] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).
- [3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR'14)*, CBLS.
- [4] D. Cai, X. He, J. Han, and T. S. Huang. 2011. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 8 (2011), 1548–1560. <https://doi.org/10.1109/TPAMI.2010.231>
- [5] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. 2018. Attentive group recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 645–654.
- [6] Da Cao, Xiangnan He, Lianhai Miao, Guangyi Xiao, Hao Chen, and Jiao Xu. 2019. Social-enhanced attentive group recommendation. *IEEE Transactions on Knowledge and Data Engineering* 33, 3 (2019), 1195–1209.
- [7] James Caverlee, Ling Liu, and Steve Webb. 2010. The Socialtrust framework for trusted social information management: Architecture and algorithms. *Information Sciences* 180, 1 (2010), 95–112.
- [8] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Bundle recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1673–1676.
- [9] Yan Chen and Sherry Xin Li. 2009. Group identity and social preferences. *American Economic Review* 99, 1 (March 2009), 431–57. <https://doi.org/10.1257/aer.99.1.431>
- [10] Mukund Deshpande and George Karypis. 2004. Item-based Top-N recommendation algorithms. *ACM Transactions on Information Systems* 22, 1 (Jan. 2004), 143–177. <https://doi.org/10.1145/963770.963776>
- [11] Li Gao, Jia Wu, Zhi Qiao, Chuan Zhou, Hong Yang, and Yue Hu. 2016. Collaborative social group influence for event recommendation. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM'16)*. Association for Computing Machinery, New York, NY, 1941–1944. <https://doi.org/10.1145/2983323.2983879>
- [12] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*. JMLR.org, 1263–1272.
- [13] Jennifer Golbeck. 2009. Trust and nuanced profile similarity in online social networks. *ACM Transactions on the Web* 3, 4, Article 12 (Sept. 2009), 33 pages. <https://doi.org/10.1145/1594173.1594174>
- [14] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM* 35, 12 (Dec. 1992), 61–70. <https://doi.org/10.1145/138859.138867>
- [15] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., Red Hook, NY, 1025–1035.

- [16] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'20)*. Association for Computing Machinery, New York, NY, 639–648. <https://doi.org/10.1145/3397271.3401063>
- [17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW'17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. <https://doi.org/10.1145/3038912.3052569>
- [18] X. He, J. Tang, X. Du, R. Hong, T. Ren, and T. Chua. 2020. Fast matrix factorization with nonuniform weights on missing data. *IEEE Transactions on Neural Networks and Learning Systems* 31, 8 (2020), 2791–2804. <https://doi.org/10.1109/TNNLS.2018.2890117>
- [19] Tajfel Henri and John C. Turner. 1986. The social identity theory of intergroup behavior. *Psychology of Intergroup Relations* 2 (1986), 7–24.
- [20] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. 1995. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*. ACM Press/Addison-Wesley Publishing Co., 194–201. <https://doi.org/10.1145/223904.223929>
- [21] Zan Huang, Wingyan Chung, and Hsinchun Chen. 2004. A graph model for E-commerce recommender systems. *Journal of the American Society for Information Science and Technology* 55, 3 (2004), 259–274. <https://doi.org/10.1002/asi.10372> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.10372>
- [22] Mohsen Jamali and Martin Ester. 2009. <i>TrustWalker</i>: A random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Paris, France) (KDD'09)*. Association for Computing Machinery, New York, NY, 397–406. <https://doi.org/10.1145/1557019.1557067>
- [23] Rong Jin and Luo Si. 2004. A Bayesian approach toward active learning for collaborative filtering. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI'04)*. AUAI Press, Arlington, VA, 278–285.
- [24] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [25] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997* (2018).
- [26] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. 1997. GroupLens: Applying collaborative filtering to usenet news. *Communications of the ACM* 40, 3 (March 1997), 77–87. <https://doi.org/10.1145/245108.245126>
- [27] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37. <https://doi.org/10.1109/MC.2009.263>
- [28] Gueorgi Kossinets and Duncan J. Watts. 2009. Origins of homophily in an evolving social network. *American Journal of Sociology* 115, 2 (2009), 405–450.
- [29] Artur Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2012. Multi-relational matrix factorization using Bayesian personalized ranking for social network data. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM'12)*. Association for Computing Machinery, New York, NY, 173–182. <https://doi.org/10.1145/2124295.2124317>
- [30] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference (Lyon, France) (WWW'18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 689–698. <https://doi.org/10.1145/3178876.3186150>
- [31] G. Linden, B. Smith, and J. York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80. <https://doi.org/10.1109/MIC.2003.1167344>
- [32] Hao Ma, Irwin King, and Michael R. Lyu. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*. Association for Computing Machinery, New York, NY, 203–210. <https://doi.org/10.1145/1571941.1571978>
- [33] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. SoRec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08)*. Association for Computing Machinery, New York, NY, 931–940. <https://doi.org/10.1145/1458082.1458205>
- [34] Sabine Matook, Susan A. Brown, and Johanna Rolf. 2015. Forming an intention to act on recommendations given via online social networks. *European Journal of Information Systems* 24, 1 (2015), 76–92.
- [35] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. 2001. Birds of a feather: Homophily in social networks. *Annual Review of Sociology* 27, 1 (2001), 415–444.

- [36] Simon Meyffret, Lionel Médini, and Frédérique Laforest. 2012. Trust-based local and social recommendation. In *Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the Social Web (RSWeb'12)*. Association for Computing Machinery, New York, NY, 53–60. <https://doi.org/10.1145/2365934.2365945>
- [37] Michael Muller, Kate Ehrlich, Tara Matthews, Adam Perer, Inbal Ronen, and Ido Guy. 2012. Diversity among enterprise online communities: Collaborating, teaming, and innovating through social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12)*. Association for Computing Machinery, New York, NY, 2815–2824. <https://doi.org/10.1145/2207676.2208685>
- [38] Wei Pan and Li Chen. 2013. GBPR: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*. AAAI Press, 2691–2697.
- [39] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD Cup and Workshop*, Vol. 2007. 5–8.
- [40] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI'09)*. AUAI Press, Arlington, VA, 452–461.
- [41] Paul Resnick and Hal R. Varian. 1997. Recommender systems. *Communications of the ACM* 40, 3 (March 1997), 56–58. <https://doi.org/10.1145/245108.245121>
- [42] Inbal Ronen, Ido Guy, Elad Kravi, and Maya Barnea. 2014. Recommending social media content to community owners. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. 243–252.
- [43] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'07)*. Curran Associates Inc., Red Hook, NY, 1257–1264.
- [44] Matthew J. Salganik, Peter Sheridan Dodds, and Duncan J. Watts. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *Science* 311, 5762 (2006), 854–856.
- [45] Sagar Samtani, Hongyi Zhu, and Hsinchun Chen. 2020. Proactively identifying emerging hacker threats from the dark web: A diachronic graph embedding framework (D-GEF). *ACM Transactions on Privacy and Security (TOPS)* 23, 4 (2020), 1–33.
- [46] Rossano Schifanella, Alain Barrat, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. 2010. Folks in folksonomies: Social link prediction from shared metadata. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*. 271–280.
- [47] Suwash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web (WWW'15 Companion)*. Association for Computing Machinery, New York, NY, 111–112. <https://doi.org/10.1145/2740908.2742726>
- [48] Upendra Shardanand and Pattie Maes. 1995. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*. ACM Press/Addison-Wesley Publishing Co., 210–217. <https://doi.org/10.1145/223904.223931>
- [49] Marvin E. Shaw. 1981. *Group Dynamics: The Psychology of Small Group Behavior*. McGraw-Hill College.
- [50] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S. Yu, Yading Yue, and Bin Wu. 2015. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. 453–462.
- [51] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [52] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. Association for Computing Machinery, New York, NY, 165–174. <https://doi.org/10.1145/331184.3331267>
- [53] S. Wei, N. Ye, S. Zhang, X. Huang, and J. Zhu. 2012. Item-based collaborative filtering recommendation algorithm combining item category with interestingness measure. In *2012 International Conference on Computer Science and Service System*. 2038–2041. <https://doi.org/10.1109/CSSS.2012.507>
- [54] L. Wu, P. Sun, R. Hong, Y. Ge, and M. Wang. 2018. Collaborative neural social recommendation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51, 1 (2018), 464–476. <https://doi.org/10.1109/TSMC.2018.2872842>
- [55] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference (WWW'19)*. Association for Computing Machinery, New York, NY, 2091–2102. <https://doi.org/10.1145/3308558.3313442>
- [56] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for Top-N recommender systems. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM'16)*. Association for Computing Machinery, New York, NY, 153–162. <https://doi.org/10.1145/2835776.2835837>

- [57] Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. Deep item-based collaborative filtering for Top-N recommendation. *ACM Transactions on Information Systems*, 37, 3, Article 33 (July 2019), 1–25. <https://doi.org/10.1145/3314578>
- [58] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.
- [59] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'18)*. Association for Computing Machinery, New York, NY, 974–983. <https://doi.org/10.1145/3219819.3219890>
- [60] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys* 52, 1, Article 5 (Feb. 2019), 38 pages. <https://doi.org/10.1145/3285029>
- [61] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM'14)*. Association for Computing Machinery, New York, NY, 261–270. <https://doi.org/10.1145/2661829.2661998>
- [62] Cai-Nicolas Ziegler and Jennifer Golbeck. 2007. Investigating interactions of trust and interest similarity. *Decision Support Systems* 43, 2 (2007), 460–475.

Received November 2020; revised March 2021; accepted April 2021