IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 21, 2020, accepted February 19, 2020, date of publication March 2, 2020, date of current version March 12, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2977407*

# Friend Recommendation Based on Multi-Social Graph Convolutional Network

**LIANG CHEN[1,2], (Member, IEEE), YUANZHEN XIE[1,2], ZIBIN ZHENG[1,2], (Senior Member, IEEE), HUAYOU ZHENG[3], AND JINGDUN XIE[4]**

[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510000, China
[2]National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510000, China
[3]Guangdong Provincial Public Security Department, Guangzhou 510000, China
[4]State Key Laboratory of Oncology in South China, Collaborative Innovation Center for Cancer Medicine, Department of Anesthesiology, Sun Yat-sen University Cancer Center, Guangzhou 510060, China

Corresponding author: Jingdun Xie (xiejd@sysucc.org.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702568 and Grant U1711267, in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X355, in part by the Fundamental Research Funds for the Central Universities under Grant 17lgpy117, in part by the Natural Science Foundation of Guangdong under Grant 2018A030313092 and Grant 2018B030325001, and in part by the Research Foundation of Guangdong Medical Science and Technology under Grant A2018236.

**ABSTRACT** Friend recommendations based on social relationships have attracted thousands of research under the rapid development of social networks. However, most of the existing friend recommendation methods use user attributes or a single social network, while rarely integrating multiple social relationships to enhance the representation. This paper focuses on integrating various social relationships to guide the representation learning, and further generating personalized friend recommendations. We design an end-to-end framework based on multiple social networks to learn the potential features of users and construct a friend recommendation model named Multi-Social Graph Convolutional Network (MSGCN). It learns the features of higher-order neighbors from multiple social networks to enrich the representation of the target user based on the improved graph convolution neural network. In particular, some graph fusion strategies by adjusting and fusing the Laplace matrix of the graph are designed to integrate social relationships. Finally, we use Bayesian theory to transform friend recommendation into a sorting problem for personalized recommendation. The experimental results show that the proposed model outperforms the state-of-the-art methods.

**INDEX TERMS** Friend recommendation, multi-social graph convolutional network, social network.

## I. INTRODUCTION

In the past decade, social networks have experienced explosive growth. Twitter, Weibo, Youtube and other social networks have billions of users, who share opinions, photos and videos every day, breaking the traditional way of making friends offline. Friend recommendations based on online social networks have become an interesting research topic. A challenging issue is how to help these users find new friends effectively. As is known to all, useful personal information of users is difficult to collect and select. Intuitively, learning social networks is helpful for friend recommendation, while the structure information of a network contains rich potential user relationships. In real life, whether two people can become friends or not has a lot to do with their social circle.

The friend recommendation problem has been studied for a long time and several approaches were proposed such as content matching (CM), content plus link (CplusL), friends of friends (FOF). Friend recommendation methods that have emerged in recent years [1]–[6] can be divided into two main categories: methods based on matrix factorization (MF) [1]–[5] and methods based on deep learning [6]. Methods based on MF are often used for item recommendation like [7]–[10], but they ignore the complex relevance of users. Methods based on deep learning have superiority in feature

The associate editor coordinating the review of this manuscript and approving it for publication was Tallha Akram.

representation, standing an important place in the field of recommendation systems. Existing methods based on deep learning such as BayDNN [6], uses CNN to obtain user representations for friend recommendation, but ignoring the structure information of social networks.

The above methods use user attributes or a single social network only. However, with the continuous development of social networks, there are various social relationships in social media. Thus it is worth thinking about how to improve the correctness of friend recommendation through the comprehensive use of these social networks. At present, there are some learning methods of multi-relational network, such as [11]–[13], but most of them ignore the structural information fusion of networks.

In conclusion, current works on friend recommendation mainly faces the following **challenges**.

- (**C1**) How to better automate the presentation of user characteristics. As is known to all, collecting artificial features is not an easy task, and a better method is needed to learn the deep presentation of the user.
- (**C2**) How to make recommendation with sparse data and alleviate cold start. The friend information is very sparse relative to non-friend information.
- (**C3**) How to use multiple social relationships to improve the accuracy of friend recommendations. Compared with the product recommendation, the relationship between people is diverse and has a lot of interactive information.

In this paper, we mainly focus on friend recommendation with multiple social networks and propose our model named multi-social graph convolutional network (MSGCN), which considers the fusion of multi-social graph. The proposed method aims to mine and aggregate some social relationships to guide friend recommendation, alleviate the data sparsity, and improve the accuracy of friend recommendation. Recommendations based on multiple social networks can be divided into two tasks: one is to learn to integrate graph embedding based on multiple social relationships as user characteristics; the other is to rank and recommend friends based on these characteristics. First, in order to solve **C1**, we design an end-to-end deep learning method that eliminates the need to manually select user features which are difficult to collect and select. To alleviate the problem of cold start, we use additional social relationships to improve the accuracy of friend recommendations, which solves the **C2** mentioned above. In addition, we designed a method based on the graph convolutional neural network (GCN) architecture that can fuse multiple relationship graphs using the fusion strategy of multiple relationships to solve the **C3**. From the perspective of measurement strategy, in order to reduce the influence of data imbalance, this paper chooses bayesian sorting method. Compared with other models, it can better alleviate the data sparsity problem with multi-social networks. The MSGCN is suitable for representing different kinds of multi-relationship graphs' (unweighted, weighted, directed, undirected, etc) nodes.

To summarize, we make the following contributions:

- Fully considering the graph structure features, we design an end-to-end model to generate the user representation, and proposed a method that utilizes user interaction information to solve the data sparsity problem well.
- We innovate to combine the structural information of multiple social graphs and explore some ways of relationship integration to alleviate the cold start problem of recommendation.
- We conduct some experiments on three real-world datasets and our results demonstrate the superior performance of MSGCN over the state-of-the-art baselines.

**Organization**: The paper is divided into five sections. Section *II* expounds the related work from three aspects: friend recommendation, graph representation and multi-relational graph. Section *III* describes the problem definition and the principle of MSGCN model. Section *IV* introduces experimental parameters, settings, indicators, etc. Compared with other related models, we prove the validity of the model and discuss the influence factors such as the number of layers and the fusion mode. Section *V* summarizes the paper and proposes several directions for future research.

## II. RELATED WORK

Focusing on highly relevant work, we review some existing related work, explain differences in this paper, and then summarize common methods of recommendation system and deep learning methods on graph data representation.

### A. FRIEND RECOMMENDATION

There are some methods for friend recommendation like CM, CplusL, FOF, and so on. Nowadays, many social platforms use methods like the FOF algorithm as one of the social friend recommendation methods. FOF algorithm is based on ternary closure theory [14], which goes like this: within a social circle, those who have more mutual friends are more likely to become friends in the future. However, this method does not take into account the similarity of features among users, which is only recommended by friends through connection relationships. Reference [11] aligns the known nodes by network alignment, but it does not consider the characteristics of network structure and can only be applied to a single type of network. Reference [15] collects user portrait information on Weibo and set up temporal-topic model to recommend different friends at different times, but user portrait information may involve user privacy. These methods are basically not applicable to sparse data, while our method can work better by mining multiple graph structure information of social relationship to learn multi-layer connection relationships.

Some methods also widely used in recommendations are those based on matrix factorization, such as the BPRMF [2], SVD [16] and NCF [3]. BPRMF [2] combines with Bayesian theory and matrix factorization, which is effective in the recommended field. SVD [16] is mainly to learn the user's

score on the product, and the scoring matrix is obtained by learning data between the user and the commodity. The higher the natural score, the higher the recommendation index. However, this method is not suitable for data with a large number of commodities, because the user has the preferences of the products, and is often concerned with some high-priority goods in the recommendation process. NCF [3] solves this shortcomings by aggregating user and project embedding information and using multiple hidden layers to enhance learning. In conclusion, these methods based on matrix factorization ignore the complex relevance of humans, and our approach is to enhance user representation by learning the connection among users.

The model based on deep learning such as BayDNN [6] combines the advantages of Bayesian Personalized Ranking (BPR) and convolutional neural network (CNN) [17] that perform well in the field of image audio and video. But CNN has a problem that there is no way to learn the map information of higher-order neighbors.

### B. MULTI-RELATIONAL GRAPH

In this paper, we study the relationship between different networks. Multi-graph relationships have been studied in many fields, such as bio-informatics [18], Predicting Disease Outcomes [13], co-saliency detection [19], recommendation [11], [12], etc. Based on the maximum retention of the network structure, [11] uses tag and contact network by selecting important features from each network to align the related networks. This method ignores some important features and does not take into account the characteristics of network structure, and is only suitable for a single network. Our method focuses on the characteristics of network structure, which is suitable for many types of networks. Based on the joint recommendation framework of joint probability distribution and matrix factor factorization of multi-social networks, [12] extends personal portraits and time information to implicit social networks, improves performance, and solves the cold start problem based on social rules and network theory. Without considering the structural characteristics of social graph, each network needs to design a special scheme, and it takes a long time to study professional knowledge. Our method pays attention to the structural characteristics of social graph and designs the learning method of end-to-end adaptation of most type of graphs.

### C. GRAPH REPRESENTATION

From the study of biological protein structure to the recommendation of friends based on social relationships, the learning tasks in the graph data are wide in scope and have many demands. The key of graph data learning lies in how to learn the topology information and transform it into the representation vector for existing machine learning models. Stimulated by the massive demand in recent years, graph learning methods have been greatly developed, making it possible to transform for learning topology into learning low-dimensional

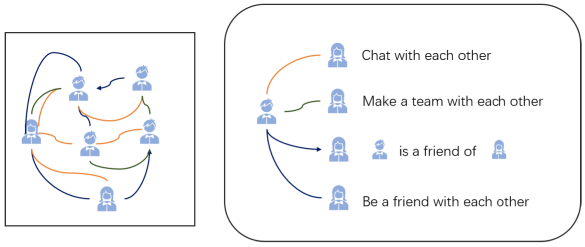


**FIGURE 1.** Overview of the multi-social graph, which contains multiple social relationships.

features. They can be divided into three methods: methods based on matrix, methods based on random walk like DeepWalk [20], LINE [21], Node2vec [22], metapath2vec [23] and methods based on graph neural network (GNN).

The emergence of GNN extents neural network to process graph data. Inspired by CNN's excellent performance in many tasks [24]–[27], GCN learns full convolution information by semi-supervised learning [28]; GraphSAGE [29] learns the node representation by aggregating neighbors by sampling subgraph; GAT [30] uses the attention mechanism to polymerize neighborhood characteristics. There are other GNN methods used in fields like [31]–[36], methods used recurrent multi-graph neural networks to solve geometric matrix completion like [37], and others which apply adversarial training thinking like GraphGAN [38].

However, previous approaches mostly focus on what to do with a single graph, rather than focusing on the problem of having multiple homogeneous data sources. Thus, we focus on solving the problem of integrating multiple graph data to learn user representation. In the paper, a variety of relationships are used to enhance the learning of user characteristics. GCN is used to learn the information of higher-order neighbors and fuse it into the node features of users, so as to make friend recommendation system based on the principle of bayesian personalized recommendation.

## III. MODEL

In this section, we elaborate the research problem at first, then introduce the architecture of Multi-Social Graph Convolutional Network referred to as MSGCN, including the process of the model's construction in detail, and finally give the method of model learning. The overall structure of the model is shown in Figure 2.

### A. BASIC CONCEPTS

To better describe the problem, we give the following definitions.

*Definition 1 (Social Network):* It is denoted as $G = \langle V, E \rangle$, where $V$ represents the user node, $E = \{e_{ij} | i \in V, j \in V\}$ represents the social relation among users, and $G$ represents the social relation graph among users (it can be chat, team, friend, etc.). $G(i, j) = 1$ represents that there is a corresponding

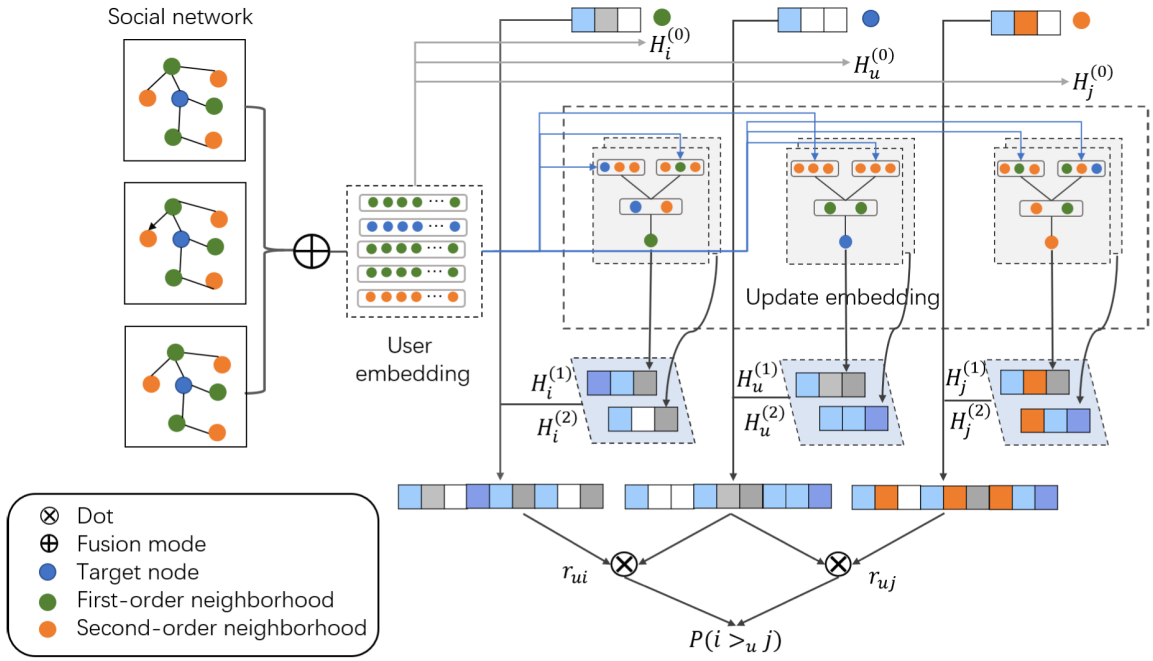


**FIGURE 2.** **Overview of the Multi-social Graph Convolutional Network Model for Friend Recommendation. In Friend Network, the line with arrow indicates a one-way friend relationship. The friend network is directed, where the chat or team network are undirected. In MSGCN, using some fusion modes based on GCN fuses multiple networks to get users' feature representation.**

social relationship between user $i$ and user $j$, and $G(i, j) = 0$ if there is no related relationship between user $i$ and user $j$.

*Definition 2 (R-Order Neighbor):* let $G = \langle V, E \rangle$ as a directed social network, the r-order neighbor of user $u$ is defined as $\Gamma_u^r = \{v : d(v, u) \leq r\}$, where $d(v, u)$ is the shortest path of $u \rightarrow v$.

*Definition 3 ($i >_u j$):* It represents the triad $\langle u, i, j \rangle$, which is defined as that user $u$ is more likely to become friends with user $i$ than user $j$. A friend probability matrix $\boldsymbol{P}$ can be established for all users. $\boldsymbol{P}_{ui} = 1$ indicates that the probability of user $u$ and user $i$ becoming friends is 1. If user $u$ is easier to make friends with user $i$ between user $i$ and user $j$, then, $\boldsymbol{P}_{ui} > \boldsymbol{P}_{uj}$, can also be expressed as $\boldsymbol{P}(i >_u j) > 0$.

### B. PROBLEM

With the continuous development of deep learning, major companies have begun to introduce relevant technologies to the recommendations of finance, e-commerce, games and so on. The emergence of deep learning can solve more problems that cannot be solved well in the past. The model should have certain versatility and does not require complex feature engineering. Undoubtedly, its development has injected new vitality into the recommendation field of friends and brought new possibilities. The emergence of GCN makes learning of graph structures easy, and is often used to learn the underlying characteristics of images, text, users, and etc. Due to the nature of deep learning itself, methods based on deep learning models are more extendible than traditional recommendation methods and can be used in many different fields.

For these inspirations, three social relationship graphs of users in the datasets are extracted - chat, team and friend relationship graphs, respectively denoted as $C(i, j)$, $T(i, j)$ and $F(i, j)$ to construct MSGCN model based on deep learning methods. As shown in Figure 1, the friend relationship is a directed graph, and the others are undirected graphs. The reason for choosing these three graphs is that in MMORPGs, friends often chat and form teams with each other more frequently. Chat and team relationship can reflect the relationship among friends to a certain extent and add some disturbance of recommendation, so that the recommendation can meet people's diversified needs and avoid the oneness of general recommendation system. In summary, the purpose is to enhance the performance of friend recommendation based on a variety of relationships. The input and output are as follows:

**Input**: The multiple related graphs and a set of friends connections.

**Output**: User rating matrix for other users, showing the top $k$ ranking for each user $u$.

We should know how to measure whether a user is more suitable to be a friend of another user or not to establish the recommendation system. In most social graphs, there is no user's rating, so the information only presents the user's preference for friends. Therefore, ranking is chosen as the recommended metric.

Personalized recommendations should provide each user with an optional list of suggestions. For the user's friend relationship, only part of the friend information can be observed.

In traditional machine learning, unobserved missing values are classified into negative feedback domains. Here, the implicit preference information of user $u$ can be represented by the result of dichotomy, reflecting user $u's$ preference and aversion information. $D^+$ represents the friends set in the $F$, and $D^-$ represents the non-friends set in the $F$. A ternary representation $\langle u, i, j\rangle$ of a given user, is named as $i >_u j$, where $i$ is from $D^+$ and $j$ is from $D^-$. If user $i$ is the friend of user $u$ and user $j$ is not the friend of user $u$, we can say that $(u, i)$ ranks higher than $(u, j)$ because user $u$ prefer to be friend with user $i$. Finally, friend recommendation transforms as a sorting problem, which will not be affected by data imbalance as easily as the classification problem. The objective function of model optimization is:

$$loss = \sum_{\langle u,i,j\rangle \in D} p(i >_u j) + \lambda ||\Theta||^2 \tag{1}$$

### C. MULTI-SOCIAL GRAPH CONVOLUTIONAL NETWORK

#### 1) FEATURE LAYER

With the development of presentation learning, each user is often represented as a row vector, showing the feature representation to the user. In our work, $\boldsymbol{e}_i \in \boldsymbol{R}^{1\times k}$ represents a representation of user $i$. With the recent development of representation learning, network embedding technology studies how to encode the structural properties of networks into low-dimensional potential spaces. Therefore, the representation learning method can be used to conduct representation learning for all users and finally form a matrix $\boldsymbol{U} \in \boldsymbol{R}^{n\times k}$, where each row of the matrix $\boldsymbol{U}$ represents the feature of the corresponding vertex, $n$ represents the number of vertices in the graph, namely the number of users, and $k$ is the number of user characteristics. In the paper, we take the user's friend link relationship as the initial user characteristic.

#### 2) MSGCN LAYER

To fully exploit different social graphs that contain heterogeneous spatial correlation information, we propose a novel MSGCN layer in our neural network model. In feature learning process of graph structure, thanks to the development of deep learning in the structure of graph data, many excellent learning methods on how to process the graph structure data have emerged. The GCN can simultaneously learn the high-order neighbor information of the fusion graph and the characteristics of the user itself without using artificial features. Now it has been used in the recommended field, such as [39] to solve the problem recommended in the actual large-scale network. It can also be applied to some problems with sparse data, such as semi-supervised learning [28]. In this work, we consider many different types of relationships, including the fusion of symmetric graphs and asymmetric graphs. Given the user's friend relationship $F = \langle V, E_f\rangle$, chat relationship $C = \langle V, E_c\rangle$ and team relationship $T = \langle V, E_t\rangle$, the idea of GCN is to study the representation of their graph structure. The input of convolutional neural network layer is the vertex eigenmatrix and graph structure information, namely the adjacency matrix $\boldsymbol{A}$. Each row of $\boldsymbol{U}$ represents the characteristics of a user. In order to aggregate the characteristics of first-order neighbors, there are:

$$\boldsymbol{U}^1 = relu(\boldsymbol{L}\boldsymbol{U}\boldsymbol{W}^1) \tag{2}$$

In order to aggregate the characteristics of the r-order neighbors, there are:

$$\boldsymbol{U}^r = relu(\boldsymbol{L}\boldsymbol{U}^{r-1}\boldsymbol{W}^r) \tag{3}$$

where $r$ is the number of layers, $\boldsymbol{W}^r \in \boldsymbol{R}^{n\times k}$ is the parameter of the rth layer. Based on different fusion mode, $L$ is defined as:

$$\boldsymbol{L} = \tilde{\boldsymbol{A}_f} + \tilde{\boldsymbol{A}_c} + \tilde{\boldsymbol{A}_t} \tag{4}$$

where $\tilde{\boldsymbol{A}_f}$,$\tilde{\boldsymbol{A}_c}$,$\tilde{\boldsymbol{A}_t}$ are uniformly represented as $\tilde{\boldsymbol{A}_*}$, which is defined as:

$$\tilde{\boldsymbol{A}_*} = \boldsymbol{D}_*^{-\frac{1}{2}}(\boldsymbol{A}_* + \boldsymbol{I})\boldsymbol{D}_*^{-\frac{1}{2}} \tag{5}$$

Among them, $\boldsymbol{I} \in \boldsymbol{R}^{n\times n}$ is the user's self-connection relationship, which is a square matrix with all 1's on the diagonal and all 0's on others, and $\boldsymbol{A}_* \in \boldsymbol{R}^{n\times n}$ represents different relationship. $\boldsymbol{A}_f$ represents friend relationship. $\boldsymbol{A}_f(i, j) = 1$ means user $j$ is a friend of user $i$ (user $i$ may not be a friend of user $j$). $\boldsymbol{A}_c$ represents the chat relationship among users. At that time, $\boldsymbol{A}_c(i, j) = 1$ represents that user $i$ and user $j$ have chat relationship. $\boldsymbol{A}_t$ represents the team relationship between user $i$ and user $j$. Similarly, $\boldsymbol{A}_t(i, j) = 1$ represents the formation relationship between user $i$ and user $j$. $\boldsymbol{D} \in \boldsymbol{R}^{n\times n}$ is the degree matrix, which can be calculated by $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$.

#### 3) FUSION MODE

In previous studies, there are three ways to represent multi-source information fusion. The first is to set up models separately and use simple splicing as the final representation vector. The second is to establish the joint optimization objective function in the training process to achieve linear fusion. Third, the shallow model representation of common input can be used to extract features and realize nonlinear fusion. Inspired by the third idea, we take the normalized symmetric Laplacian matrices of the graph for fusion in the processing of multi-social graphs to solve the **C3**, which can enrich the learned structural information transmitted through nodes. We propose four fusion methods. The *add* strategy integrates the graph structure information through adding Laplacian matrices. The *min* and *max* strategies are similar to pooling operations. Compared with adding Laplace matrix results, the risk of overfitting will be reduced when we use *min* or *max* operations. The *multi* strategy integrates through trainable parameters to learn the importance of different relationships. Because there are some different relationship graphs, some fusion modes are proposed, such as MSGCN_add, MSGCN_min, MSGCN_max and MSGCN_multi. Mathematically, the difference among them is in the representation of $\boldsymbol{L}$. $\boldsymbol{L}_{add}, \boldsymbol{L}_{min}, \boldsymbol{L}_{max}, \boldsymbol{L}_{multi}$. They are defined as:

$$\boldsymbol{L}_{add} = \tilde{\boldsymbol{A}_f} + \tilde{\boldsymbol{A}_c} + \tilde{\boldsymbol{A}_t} \tag{6}$$

$$\boldsymbol{L}_{min} = minpool(\tilde{\boldsymbol{A}}_f, \tilde{\boldsymbol{A}}_c, \tilde{\boldsymbol{A}}_t) \tag{7}$$

$$\boldsymbol{L}_{max} = maxpool(\tilde{\boldsymbol{A}}_f, \tilde{\boldsymbol{A}}_c, \tilde{\boldsymbol{A}}_t) \tag{8}$$

$$\boldsymbol{L}_{multi} = \tilde{\boldsymbol{A}}_f \boldsymbol{W}_1 + \tilde{\boldsymbol{A}}_c \boldsymbol{W}_2 + \tilde{\boldsymbol{A}}_t \boldsymbol{W}_3 \tag{9}$$

where, $\boldsymbol{W_1}$, $\boldsymbol{W_2}$ and $\boldsymbol{W_3}$ are matrices of $R^{n\times n}$, *minpool* is similar to the minimum pooling idea, taking the minimum of each value in the matrices, while *maxpool* takes the maximum value. E.g, if $\tilde{\boldsymbol{A}}_f(i,j) = 1.0$, $\tilde{\boldsymbol{A}}_c(i,j) = 0.5$, $\tilde{\boldsymbol{A}}_t(i,j) = 0.8$, then $\boldsymbol{L}_{min} = 0.5$, $\boldsymbol{L}_{max} = 1.0$. Compared with [13] and [19], we fuse the Laplace matrix with the structure information of the graph directly instead of performing the convolution operation directly on a single graph to learn the features and then concat them. Our method may learn the structure information of different graphs better by considering the fusion of Laplace matrix.

The model in this paper has a certain degree of extensibility. If a new relation $x$ is needed, $\tilde{\boldsymbol{A}}_x$ can be added in $\boldsymbol{L}$. Modifying equation (4), multiple different relations can be processed simultaneously in the model.

#### 4) HIDDEN LAYER

In order to consider some deviation factors, we introduces the hidden layer. After passing through the MSGCN layer, the user representation $\boldsymbol{U}$ is obtained and for each user $\boldsymbol{U}$ have:

$$\boldsymbol{H}_u = ReLU(\boldsymbol{w}\boldsymbol{U}_u + \boldsymbol{b}) \tag{10}$$

The *ReLU* activation function is used for the consideration of the convergence speed and range. The $\boldsymbol{w}and\boldsymbol{b}$ are the weights and deviations of neurons.

#### 5) PREDICTION LAYER

After passing through the hidden layer, a representation set of different layers can be obtained. Once the user characteristics have been obtained, users need to be ranked in order to recommend friends to the target user. User $u's$ score for user $i$ can be predicted by:

$$\boldsymbol{P}_{ui} = \sigma(\boldsymbol{H}_u \boldsymbol{H}_i^T) \tag{11}$$

where, $\sigma(.)$ is the activation function. However, since the superposition of MSGCN layer will bring the risk of overfitting, the splicing operation is carried out for each layer in order to reduce this risk:

$$\boldsymbol{U}_u^* = concat(\boldsymbol{U}_u^1, \boldsymbol{U}_u^2, \ldots, \boldsymbol{U}_u^r) \tag{12}$$

$$\boldsymbol{H}_u^* = ReLU(\boldsymbol{w}^* \boldsymbol{U}_u^* + \boldsymbol{b}^*) \tag{13}$$

where *concat* is the splicing operation. Therefore, user $u's$ preference score for user $i$ becomes:

$$\boldsymbol{P}_{ui}^* = \sigma(\boldsymbol{H}_u^* \boldsymbol{H}_i^{*T}) \tag{14}$$

### D. MODEL LEARNING

After getting user ratings, a sorted list should be presented for recommendation based on users' interests and preferences which can be described from different perspectives based on their multiple social relationships.

Bayesian theory is widely used in the field of inference diagnosis, and provides a more reasonable solution for the friend recommendation. In details, friendship is divided into three categories: the positive, negative and unknown. Our goal is to predict the unknown friendship. At the time of training, the positive relationship is greater than the negative and unknown relationship to form Bayesian order. In the work of this paper, the user preference is implicit feedback. Inspired by the above thoughts, after given a group $\langle u, i, j\rangle$, which indicates that user $u$ prefer to be friends with user $i$ than user $j$, the Bayesian Personalized Ranking is adopted here to calculate the loss function of the model.

$$loss = \sum_{\langle u,i,j\rangle \in D} -ln\sigma(\boldsymbol{P}_{ui}^* - \boldsymbol{P}_{uj}^*) + \lambda||\Theta||^2 \tag{15}$$

where, $D = \{\langle u, i, j\rangle \mid \langle u, i\rangle \in D^+, \langle u, j\rangle \in D^-\}$ represents the triad of training, $D^+$ represents the positive friendship that user $i$ is a friend of user $u$, and $D^-$ represents the negative friendship that user $j$ is not a friend of user $u$. In order to avoid overfitting the model, a penalty term $\lambda||\Theta||^2$ is added.

In the experiment, the mini-batch learning method is adopted to learn and train model parameters, which costs little time and is more efficient. In each training iteration, positive samples $(i,j) \in D^+$ with a random size of batch size are selected. For each user, negative samples $(i,j) \in D^-$ of the same size are randomly selected as one batch. User characteristics are generated by MSGCN layer and Adam optimizer is applied to calculate the objective function in the experiment.

## IV. EXPERIMENT

In this part, the method proposed in this paper is compared with other methods on three real-world datasets of Netease games to evaluate the advantages and disadvantages of the proposed model. At the same time, the effect of different fusion methods and MSGCN layers on the model are discussed. All of them aim to answer the following research questions:

**RQ1** Can the MSGCN outperform the state-of-the-art models? Can the learning of aggregating multiple graphs improve performance?

**RQ2** Does the more layers, the better the model is? Is it better to learn r-order neighbor information at a higher order?

**RQ3** Which graph fusion method proposed has better performance? Why is that?

### A. THE EXPERIMENTAL SETUP

#### 1) DATASETS

The experimental datasets of this paper are from different online multiplayer role-playing computer games of Netease. The datasets are named Game 1, Game 2, and Game 3. In the experiment, just some active users are taken into account in the selection. Since inactive users are likely to have been separated from the game, users with more than one connection number in the friend relationship are selected in the datasets. The statistics of experimental datasets are

**TABLE 1. Statistics of experimental datasets.**

| Dataset | #Users | #Friends | #Chat | #Team | Density |
|---|---|---|---|---|---|
| Game 1 | 40279 | 251794 | 267819 | 524722 | 0.0155% |
| Game 2 | 36216 | 156160 | 447302 | 337666 | 0.0119% |
| Game 3 | 27065 | 86528 | 169245 | 206184 | 0.0118% |

shown in Table 1, including the number of users, friends, chat connections and team connections of the datasets. The densities of three datasets are low, which shows the sparsity of data.

In three datasets, the number of chat connections and team connections are greater than the number of friends. To some extent, the team and chat relationship reflects the relationship among friends. Two users who often organize a team and chat with each other may become friends. So aggregating those information may be helpful for friend recommendations. In addition, the matrix stored in the experiment consumes a lot of memory due to the large number of users. Considering that the connections among users are relatively sparse, we use the sparse matrix.

#### 2) DATA PREPROCESSING

We collect data sets based on time and do some processing on them. In Game 1, the training data is obtained from June 30, 2018 to July 4, 2018, and the team formation and chat relationship pairs are obtained in the same time period. Testing data is available from July 5, 2018 to July 6, 2018. In Game 2, the training data is obtained from July 5, 2018 to July 10, 2018. Meanwhile, the team formation and chat relationship pairs in the same period are acquired. Testing data is available from July 11, 2018 to July 12, 2018. In Game 3, the training data and testing data are consistent with Game 2. The training data and the testing data are divided according to the number of days. In general, we use six days of datum to predict the change in the next two days.

#### 3) EVALUATION METRICS

We employ leave-one-out evaluation method. Instead of testing all non-friends, 199 non-friends are randomly selected for a target user $u$. In order to evaluate the performance of the first $k$ recommendation ranking models, Recall@k and Hits@k are selected to evaluate the performance of positive feedback items, and MAP@k and NDCG@k are selected to evaluate the excellence of the ranking.

- **MAP**: Mean Average Precision (MAP) is the average accuracy of correct recommendations, while MAP@k refers to calculating the average accuracy of the first k recommendations.

$$MAP@k = \frac{1}{|U_t|} \frac{\sum_{u \in U^t} P(k) \times rel(k)}{\sum_{u \in U^t} rel(k)} \tag{16}$$

- **Recall**: Recall represents the percentage of all items that are correctly matched against all items which should be recommended.

$$Recall@k = \frac{1}{|U_t|} \sum_{u \in U^t} hit(u)@k \tag{17}$$

where, $rel(k)$ represents correlation, expressed as 0 and 1 respectively.

- **Hits**: Hits indicates whether the correct recommendations have been made to the test user.

$$Hits@k = \frac{1}{|U|} \sum_{u \in U} hit(u)@k \tag{18}$$

where, $|U|$ represents the number of test users, hit(u) represents whether the recommendation is hit or not. The formula is as follows:

$$hit(u)@k = \begin{cases} 1 & if\ hit \\ 0 & if\ not\ hit \end{cases} \tag{19}$$

- **NDCG**: Normalized Discounted Cumulative Gain (NDCG) is usually used to measure and evaluate search results algorithm, which is based on a ranking index.

$$NDCG@k = \frac{DCG@k}{IDCG@k} \tag{20}$$

$$IDCG@k = \sum_{i=1}^{min(k,n)} \frac{2^{y^i} - 1}{log_2(i+1)} \tag{21}$$

#### 4) BASELINES

In order to prove the validity of the model, the model in this paper is compared with some existing friend recommendation models in the experiment. In the selection of comparison methods, the time sequence of the relevant methods and the applicability of the method to the problems proposed in this paper are considered. The baselines are MF [40], BPRMF [2], NCF [3], a friend recommendation method based on CNN, named BayDNN [6], GCN-cat [19], and the comparison of several single relationship graph of GCN. In summary, there are three methods based on matrix factorization and five methods based on neural networks. Traditional friend recommendation methods such as FOF are not suitable for solving the recommendation problem of sparse data. The number of common friends in the datasets is very rare. Through the experiment, it is found that almost all indexes of the FOF method are 0, so it is not selected as one of the baselines.

- **MF**: Matrix factorization is a potential factor model, which is widely used in the field of sequencing prediction. The MF method proposed by [40] takes positive connection as 1 and negative connection as 0 to solve the problem of link prediction.
- **BPRMF**: BPRMF [2] is a matrix factorization model relying on bayesian loss pair optimization, which is proved to be effective in the field of the friend recommendation.
- **NCF**: NCF [3] is an optimization model of matrix factorization, which aggregates user representation and uses

**TABLE 2.** Performance comparison of our methods and other methods when k = 3.

| Method | Game 1 | | | | Game 2 | | | | Game 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Recall@3 | MAP@3 | Hits@3 | NDCG@3 | Recall@3 | MAP@3 | Hits@3 | NDCG@3 | Recall@3 | MAP@3 | Hits@3 | NDCG@3 |
| MF | 0.7335 | 0.6638 | 0.7046 | 0.6818 | 0.8104 | 0.7342 | 0.5331 | 0.7538 | 0.8053 | 0.7303 | 0.5085 | 0.7496 |
| BPRMF | 0.8766 | 0.7843 | 0.8544 | 0.8081 | 0.8992 | 0.818 | 0.5988 | 0.8389 | 0.8332 | 0.7288 | 0.52 | 0.7556 |
| NCF | 0.8471 | 0.7598 | 0.8303 | 0.7823 | 0.8762 | 0.7967 | 0.5867 | 0.8172 | 0.8502 | 0.7405 | 0.5765 | 0.7687 |
| BayDNN | 0.7327 | 0.6044 | 0.7078 | 0.6374 | 0.8312 | 0.7158 | 0.5528 | 0.7455 | 0.8661 | 0.789 | 0.5604 | 0.8088 |
| GCN-cat | 0.3366 | 0.2854 | 0.3877 | 0.2985 | 0.2380 | 0.1903 | 0.1988 | 0.2026 | 0.4208 | 0.3518 | 0.3503 | 0.3695 |
| MSGCN_f | 0.9221 | 0.8465 | 0.9153 | 0.866 | 0.923 | 0.8479 | 0.631 | 0.8673 | 0.9225 | 0.8472 | 0.6153 | 0.8666 |
| MSGCN_c | 0.916 | 0.8421 | 0.9118 | 0.8611 | 0.9141 | 0.8406 | 0.6283 | 0.8595 | 0.9227 | 0.8456 | 0.6151 | 0.8655 |
| MSGCN_t | 0.9119 | 0.8316 | 0.9089 | 0.8523 | 0.9191 | 0.8434 | 0.6283 | 0.8629 | 0.9197 | 0.8432 | 0.6149 | 0.863 |
| MSGCN_best | **0.9247** | **0.8503** | **0.9196** | **0.8695** | **0.9308** | **0.8582** | **0.6384** | **0.8769** | **0.9354** | **0.8612** | **0.6241** | **0.8803** |
| %Improv. | 5.49% | 8.42% | 7.63% | 7.60% | 3.51% | 4.91% | 6.61% | 4.53% | 9.38% | 14.77% | 8.26% | 13.24% |

**TABLE 3.** Performance comparison of our methods and other methods when k = 5.

| Method | Game 1 | | | | Game 2 | | | | Game 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Recall@5 | MAP@5 | Hits@5 | NDCG@5 | Recall@5 | MAP@5 | Hits@5 | NDCG@5 | Recall@5 | MAP@5 | Hits@5 | NDCG@5 |
| MF | 0.7795 | 0.6744 | 0.7531 | 0.7007 | 0.8613 | 0.7459 | 0.5722 | 0.7748 | 0.8518 | 0.7408 | 0.5468 | 0.7687 |
| BPRMF | 0.9322 | 0.7972 | 0.919 | 0.8311 | 0.9426 | 0.8279 | 0.6386 | 0.8568 | 0.9034 | 0.7449 | 0.5863 | 0.7846 |
| NCF | 0.9017 | 0.7723 | 0.89 | 0.8048 | 0.9211 | 0.807 | 0.623 | 0.8357 | 0.9148 | 0.7554 | 0.6174 | 0.7954 |
| BayDNN | 0.8198 | 0.6245 | 0.8005 | 0.6734 | 0.8906 | 0.7294 | 0.6 | 0.77 | 0.918 | 0.8008 | 0.6027 | 0.8302 |
| GCN-cat | 0.3884 | 0.2966 | 0.4412 | 0.3193 | 0.2795 | 0.1998 | 0.2297 | 0.2197 | 0.4626 | 0.3613 | 0.3787 | 0.3867 |
| MSGCN_f | 0.9549 | 0.8541 | 0.9505 | 0.8796 | 0.9554 | 0.8554 | 0.6536 | 0.8807 | 0.9564 | 0.855 | 0.6393 | 0.8807 |
| MSGCN_c | 0.9508 | 0.8502 | 0.9472 | 0.8756 | 0.9513 | 0.8492 | 0.6537 | 0.8749 | 0.9549 | 0.8531 | 0.6378 | 0.8788 |
| MSGCN_t | 0.9492 | 0.8403 | 0.9478 | 0.8678 | 0.952 | 0.8509 | 0.6529 | 0.8765 | 0.9556 | 0.8515 | 0.6394 | 0.8778 |
| MSGCN_best | **0.9576** | **0.8579** | **0.9547** | **0.8831** | **0.9606** | **0.8651** | **0.6595** | **0.8893** | **0.9633** | **0.8676** | **0.6447** | **0.8919** |
| %Improv. | 2.72% | 7.61% | 3.88% | 6.26% | 1.91% | 4.49% | 3.27% | 3.58% | 5.61% | 13.65% | 4.42% | 11.38% |

multiple hidden layers to mine higher-order features of users.

- **BayDNN**: BayDNN [6] models the friends problem as a sort problem and utilizes multi-layer CNN to extract user features. It employs the Bayesian theory to construct the objective function.
- **GCN-cat**: GCN-cat [19] work well in image filed for Co-Saliency Detection. In dealing with multiple graphs, it uses GCN operations on single graph, and then concatenates the results of those graphs.

The MSGCN is also compared with three single-relationship-based models, including the MSGCN_f based on friend relationship, MSGCN_c based on chat relationship and MSGCN_t based on team relationship.

#### 5) HYPER-PARAMETER SETTINGS

The MSGCN model and baseline methods are implemented by python, and relevant experiments are mainly completed by using the Pytorch framework. Adam optimizer is selected, which is more convenient for adjusting the learning rate. The selection range of the overparameter is listed below: The batch size of all models is fixed to 256. The learning rate is selected from [0.01, 0.001, 0.0001, 0.00001]. The $L_2$ regular term is selected from [0.1, 0.01, 0.001, 0.0001]. The dropout ratio in NCF is selected the best from [0.1, 0.2, 0.3, 0.4, 0.5]. The default layer number of MSGCN correlation model is 2 if there is no description, while in the layer number experiment it is selected from [1, 2, 3, 4, 5].

### *B. PERFORMANCES COMPARISON (RQ1)*

The experiment results of all models are showed in Table 2 and Table 3, where $k$ is set to 3 and 5. The MSGCN_best in Table 2 and 3 is the proposed model which has the best performance.

- We can find that among the three models related to matrix factorization, BPRMF and NCF have higher performance than MF, indicating that the optimized matrix factorization model performs better in the field of friend recommendation, and the aggregation of higher-order user characteristics can improve the performance.
- Compared with other neural networks and matrix factorization methods, BayDNN does not work well on three datasets, which indicates that BayDNN model may not work well in sparse data. The performance of Game 2 and Game 3 have been improved, indicating that it is more suitable for the recommendation work which is based on the accumulation of certain data in the system.
- Compared with our methods, GCN-cat, which used in image field, does not work well on three datasets. It indicates that this model maynot work well in discrete data. Rather than learning from a single graph structure to aggregate different features, it is better to aggregate multiple graph structure as features directly.
- MSGCN_f, MSGCN_c, and MSGCN_t make use of the information of graph structure, and perform better than the previous method due to the use of other information to solve the problem of data sparsity. It indicates that the use of some structure information can better capture user preferences. It also shows that friend, team and chat graph structure information are helpful to friend recommendation. It can be found that the performance of the three, MSGCN_f is the best, MSGCN_c is the second and MSGCN_t is the third. It shows that friend relationship is more effective and chatting

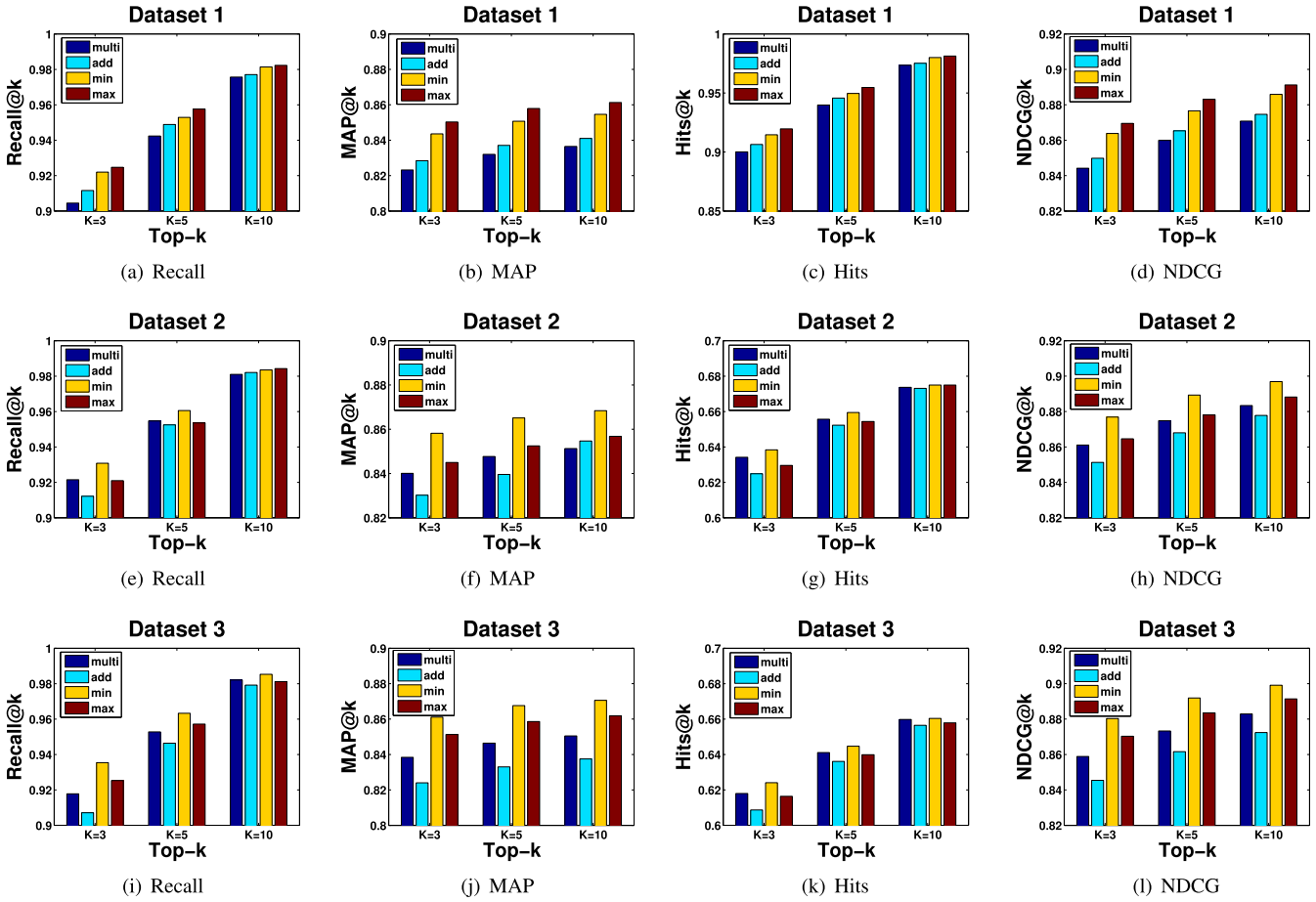


**FIGURE 3.** Impact of fusion mode. (a)-(d), (e)-(h), (I)-(l) respectively represent the performance comparison of various fusion methods on Game 1, Game 2 and Game 3 under different k values; Multi, add, min and max respectively represent MSGCN_multi, MSGCN_add, MSGCN_min and MSGCN_max.

**TABLE 4.** Impact of the MSGCN layer number in MSCGN_max.

| #Layer | Game 1 | | | | Game 2 | | | | Game 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Recall@3 | MAP@3 | Hits@3 | NDCG@3 | Recall@3 | MAP@3 | Hits@3 | NDCG@3 | Recall@3 | MAP@3 | Hits@3 | NDCG@3 |
| 1 | 0.581 | 0.4592 | 0.5695 | 0.4904 | 0.5947 | 0.4677 | 0.4003 | 0.5003 | 0.5807 | 0.4551 | 0.3856 | 0.4873 |
| 2 | 0.9247 | 0.8503 | 0.9196 | 0.8695 | 0.921 | 0.845 | 0.6296 | 0.8646 | 0.9253 | 0.8513 | 0.6164 | 0.8703 |
| 3 | 0.9907 | 0.9672 | 0.9886 | 0.9733 | 0.9882 | 0.962 | 0.6789 | 0.9688 | 0.988 | 0.9618 | 0.6625 | 0.9686 |
| 4 | 0.9976 | 0.9902 | 0.9971 | 0.9921 | 0.9988 | 0.9914 | 0.6862 | 0.9933 | 0.9983 | 0.9868 | 0.6698 | 0.9898 |
| 5 | 0.9969 | 0.9692 | 0.9956 | 0.9764 | 0.9965 | 0.9569 | 0.6846 | 0.9672 | 0.9931 | 0.9483 | 0.666 | 0.9599 |

among friends is more common than forming teams in games.

- It can be seen that MSGCN model is superior to the baselines, which proves the effectiveness of the model proposed in this paper and proves that MSGCN can better aggregate user characteristics when data is sparse from the results of the three datasets. The improvement of this effect stems from the aggregation of multiple social networks to learn the higher-order features of aggregated users.
- Compared with MSGCN_f, MSGCN_c and MSGCN_t, the improvement indicates that MSGCN integrates multiple user relationships, from which it learns the deeper diversified representation, such as the structure information and the link information of multiple graphs.
- MSGCN has a more significant improvement effect than other models on Game 1, the data of which is generated at the beginning of the game, indicating that the MSGCN model can effectively solve the recommendation problem on sparse data to some extent.

### C. IMPACT OF LAYERS (RQ2)

As mentioned above, due to the limitations of graph convolutional model, too many convolutional layers will lead to the risk of overfitting. The relevant experimental results are shown in Table 4 and Table 5. The layer number range of the experiment is from 1 to 5, and k is set as 3. MSGCN_max and MSGCN_min models are selected for exploration due to their good performances.

- According to the result, it can be found that the performance of the two models increases from layer 1 to

**TABLE 5.** Impact of the MSGCN layer number in MSCGN_min.

| #Layer | Game 1 | | | | Game 2 | | | | Game 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Recall@3 | MAP@3 | Hits@3 | NDCG@3 | Recall@3 | MAP@3 | Hits@3 | NDCG@3 | Recall@3 | MAP@3 | Hits@3 | NDCG@3 |
| 1 | 0.5745 | 0.453 | 0.5691 | 0.4842 | 0.5998 | 0.4716 | 0.4052 | 0.5045 | 0.5842 | 0.462 | 0.385 | 0.4933 |
| 2 | 0.922 | 0.8436 | 0.9146 | 0.8638 | 0.9308 | 0.8582 | 0.6384 | 0.8769 | 0.9354 | 0.8612 | 0.6241 | 0.8803 |
| 3 | 0.9866 | 0.9648 | 0.9855 | 0.9704 | 0.9901 | 0.9687 | 0.68 | 0.9743 | 0.9878 | 0.9633 | 0.6632 | 0.9697 |
| 4 | 0.9992 | 0.9926 | 0.9981 | 0.9943 | 0.9991 | 0.9932 | 0.6865 | 0.9947 | 0.9991 | 0.9921 | 0.6707 | 0.9939 |
| 5 | 0.9999 | 0.9977 | 0.999 | 0.9983 | 0.9997 | 0.997 | 0.687 | 0.9977 | 0.9998 | 0.9971 | 0.6713 | 0.9978 |

layer 4, because more MSGCN layers can aggregate the information of higher-order neighbors to enrich user representation.

- When there are five layers of MSGCN, the performance of MSGCN_max model is not improved compared with MSGCN_max model that has four layers. The reason may be that noise is introduced in the process of introducing many layers. Another possible reason is that models with too many layers are more likely to fall into the risk of overfitting.
- The more layers there are, the longer the corresponding training time will be spent. Generally, two-layer models with good effects and efficiencies are selected for experiments.

### D. IMPACT OF FUSION MODE (RQ3)

Several simple ideas in fusion mode are proposed. First, based on a user representation, three parts of the graph are simply added for direct fusion (*add*). Second, we perform similar minimum and maximum pooling operations (*min* and *max*, respectively, and take the minimum or maximum of the three parts) for multiple graphs. The third (*multi*) is to compute the representation of different users based on different graphs and then do the fusion of the representation. The above mentioned methods, including *multi*, *add*, *min* and *max*, are compared and the results are shown in Figure 3.

- According to the results, the overall performance is relatively complex. Four evaluations are considered comprehensively. In Game 1, MSGCN_max has the best performance and MSGCN_multi has the worst performance. In Game 2 and Game 3, MSGCN_min has the best performance, followed by MSGCN_max, while MSGCN_add has the poor performance. After these discussion, it can be found that MSGCN_max and MSGCN_min are relatively effective.
- Compared with simple addition, *min* and *max* avoid adding more noise on the basis of using three kinds of graph structures to improve the characterization. They are much simpler than MSGCN_multi, learning a single user representation uniformly, and introducing less noise.
- The Recall and Hits of MSGCN_multi in Game 2 and Game 3 are relatively good, but not good in Game 1, which shows that the performance of MSGCN_multi has been improved after a period of time. At the beginning of game opening, the four methods have little difference and MSGCN_max has slightly higher performance. MSGCN_min has the highest performance in Game 2 and Game 3.

## V. CONCLUSION

The paper mainly focuses on the problem of friend recommendation with sparse data, and proposes a friend recommendation model based on multi-social graph convolution, named as MSGCN. With the idea of GCN, we design a strategy integrating topological structure information from multiple graphs to obtain user's representation. By using bayesian theory, the friend recommendation is transformed into a ranking problem of personalized recommendation. In this paper, three real-world datasets from Netease games are used in the experiments, and the results show that MSGCN model is better than some existing friend recommendation models. The MSGCN model proposed in this paper has the advantages of considering multiple social networks and strong expansion ability to solve data sparsity issue. This is the first time that deep learning is used to enhance user representation based on multiple social graph structures in the field of friend recommendation. The model can solve the problem of the friend recommendation with sparse data effectively, and it has certain mobility and flexibility.

This work explores the potential of network representation in the field of friend recommendation and points to several directions in the future. First, the attention mechanism will be used to enhance the user feature learning in the graph convolution layer, so as to improve the performance of friend recommendation. The second is to comprehensively process the user's friend deletion information, take into account the negative information and time factors, use the reinforcement learning content plus the user's decision layer, and make decisions on whether he will make friends with others or will delete a friend. Third, it can also be combined with explainable recommendation system to strengthen the interpretability of the deep learning model of friend recommendation. Better user representation is expected to get to complete the tasks recommended by friends efficiently.

## REFERENCES

[1] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2016, pp. 549–558.

[2] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.

[3] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

[4] K. Zhou, S.-H. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf.*, 2011, pp. 315–324.
[5] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. Proc. 17th ACM Conf. Inf. Knowl. Mining (CIKM)*, 2008, pp. 931–940.
[6] D. Ding, M. Zhang, S.-Y. Li, J. Tang, X. Chen, and Z.-H. Zhou, "BayDNN: Friend recommendation with Bayesian personalized ranking deep neural network," in *Proc. ACM Conf. Inf. Knowl. Manage. (CIKM)*, 2017, pp. 1479–1488.
[7] X. Ren, M. Song, H. E, and J. Song, "Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation," *Neurocomputing*, vol. 241, pp. 38–55, Jun. 2017.
[8] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 59–66.
[9] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "IDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization," *Future Gener. Comput. Syst.*, vol. 66, pp. 30–35, Jan. 2017.
[10] L. Yao, X. Wang, Q. Z. Sheng, B. Benatallah, and C. Huang, "Mashup recommendation by regularizing matrix factorization with API co-invocations," *IEEE Trans. Services Comput.*, to be published.
[11] S. Huang, J. Zhang, L. Wang, and X.-S. Hua, "Social friend recommendation based on multiple network correlation," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 287–299, Feb. 2016.
[12] W. Yu and S. Li, "Recommender systems based on multiple social networks correlation," *Future Gener. Comput. Syst.*, vol. 87, pp. 312–327, Oct. 2018.
[13] J. Valenchon and M. Coates, "Multiple-graph recurrent graph convolutional neural network architectures for predicting disease outcomes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3157–3161.
[14] M. S. Granovetter, "The strength of weak ties1," *Amer. J. Sociol.*, vol. 78, no. 6, pp. 1360–1380, 1973.
[15] S. Huang, J. Zhang, D. Schonfeld, L. Wang, and X.-S. Hua, "Two-stage friend recommendation based on network alignment and series expansion of probabilistic topic model," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1314–1326, Jun. 2017.
[16] G. Guo, J. Zhang, and N. Yorke-Smith, "Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 123–129.
[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
[18] M. H. Goodfellow, J. Wilson, and E. Hunt, "Biochemical network matching and composition," in *Proc. 1st Int. Workshop Data Semantics*, 2010, p. 40.
[19] B. Jiang, X. Jiang, J. Tang, B. Luo, and S. Huang, "Multiple graph convolutional networks for co-saliency detection," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 332–337.
[20] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 701–710.
[21] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web (WWW)*, 2015, pp. 1067–1077.
[22] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, pp. 855–864.
[23] Y. Dong, N. V. Chawla, and A. Swami, "Metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2017, pp. 135–144.
[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[25] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: http://arxiv.org/abs/1408.5882
[26] J. Yu, C. Hong, Y. Rui, and D. Tao, "Multitask autoencoder model for recovering human poses," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 5060–5068, Jun. 2018.
[27] J. Yu, Z. Kuang, B. Zhang, W. Zhang, D. Lin, and J. Fan, "Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1317–1332, May 2018.
[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907
[29] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[30] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, andY. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*. [Online]. Available: http://arxiv.org/abs/1710.10903
[31] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," 2018, *arXiv:1812.08434*. [Online]. Available: http://arxiv.org/abs/1812.08434
[32] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," 2018, *arXiv:1801.10247*. [Online]. Available: http://arxiv.org/abs/1801.10247
[33] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4800–4810.
[34] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," 2018, *arXiv:1806.03536*. [Online]. Available: http://arxiv.org/abs/1806.03536
[35] W. Huang, T. Zhang, Y. Rong, and J. Huang, "Adaptive sampling towards fast graph representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4558–4567.
[36] Z. Li, Q. Chen, and V. Koltun, "Combinatorial optimization with graph convolutional networks and guided tree search," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 539–548.
[37] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3697–3707.
[38] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, andM. Guo, "Graphgan: Graph representation learning with generative adversarial nets," in *Proc. 23nd AAAI Conf. Artif. Intell.*, 2018, pp. 2508–2515.
[39] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, andJ. Leskovec, "Graph convolutional neural networks for Web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2018, pp. 974–983.
[40] A. K. Menon and C. Elkan, "Link prediction via matrix factorization," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Springer, 2011, pp. 437–452.

**LIANG CHEN** (Member, IEEE) received the bachelor's and Ph.D. degrees from the Advanced Computing and System Laboratory, College of Computer Science and Technology, Zhejiang University, China, in 2009 and 2015, respectively. He is currently a Distinguished Research Fellow with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His research areas including services computing, social network, recommendation systems, heterogenous information networks and graph attacks, and perturbations, with a focus on solving traditional challenges via heterogeneous data sources and data mining techniques.

**YUANZHEN XIE** received the bachelor's degree from Sun Yat-sen University, Guangzhou, China, in 2019, where she is currently pursuing the master's degree in computer technology with the School of Data and Computer Science. Her research interests include social network, recommendation systems, and graph neural network.

**ZIBIN ZHENG** (Senior Member, IEEE) received the Ph.D. degree from the Chinese University of Hong Kong, in 2011.

He serves as the Chairman of the Software Engineering Department. He is currently a Professor with the School of Data and Computer Science, Sun Yat-sen University, China. He has published more than 120 international journals and conference papers, including three ESI highly cited articles. According to Google Scholar, his articles have more than 7000 citations, with an H-index of 42. His research interests include blockchain, services computing, software engineering, and financial big data. He was a recipient of several awards, including the Top 50 Influential Papers in Blockchain of 2018, the ACM SIGSOFT Distinguished Paper Award at ICSE2010, and the Best Student Paper Award at ICWS2010. He has served as BlockSys'19 and the CollaborateCom'16 General Co-Chair, SC2'19, ICIOT'18, and the IoV'14 PC Co-Chair.

**HUAYOU ZHENG** graduated in 2007. He was assigned to the Guangdong Provincial Public Security Department. He has been engaged in the Internet Security Management for a long time and has extensive Internet management experience. He is currently an Expert of the Ministry of Public Security, Provincial Public Security Department.

**JINGDUN XIE** received the M.D. and Ph.D. degrees in medicine from Sun Yat-sen University, Guangzhou, China, in 2006 and 2016, respectively. He is currently a Distinguished Research Fellow with the Sun Yat-sen University Cancer Center, China. He has published articles in prestigious journals in neuroscience, e.g., the *Journal of Chemistry* and *Neuropharmacology*. His research interests include neuroscience, brain machine interface, anesthesiology, and pain.