2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

A Heterogeneous Network Community Detection Method Based on GCN and Social Recommendation

Zixiao Zhang[1,2], Jinglian Liu[1,2](✉), Shan Zhong[1,3], Yali Si[1,3], and Shengrong Gong[1,3]

[1] School of Computer Science and Engineering, Suzhou University of Technology, Suzhou, China
`liujinglian@cslg.edu.cn`
[2] School of Electronics and Information Engineering, Liaoning University of Technology, Jinzhou, China
[3] School of Computer Science and Technology, Soochow University, Suzhou, China

Abstract. To address the issue of sparse and singular connections between users and communities, we propose a novel heterogeneous network community detection method (GS-HCD) that combines GCN (graph convolutional neural network) with social recommendation. First, the heterogeneous information network is transformed into a user-user social network (G_{uu})and a user-community binary network (G_{uc}) based on predefined meta paths. The GCN model architecture was adjusted by adding regularization and appropriate activation functions, which achieves the network optimization of G_{uu} and G_{uc}. A labeling mechanism is then introduced to merge the two optimized networks and construct a user-community extension graph G_{cu}. Then, in the extended graph G_{cu}, meta paths that satisfy the criteria are selected between the target user node and the candidate community nodes, and the AvgSim similarity index is used to calculate the similarity based on these meta paths, forming candidate node pairs. Finally, input the vector information of user community candidate nodes into the social recommendation model based on deep learning, learn to capture the dynamic changes of user interests in social networks, and recommend the optimal communities for users. The experimental results on three classical datasets confirm the performance of the GS-HCD model and the accuracy of community detection. Compared with multiple representative methods, the GS-HCD model performs outstandingly in terms of *Precision*, *Recall*, and *F-score* values, and its F1 and AUC values generally exceed the comparative baselines, demonstrating its effectiveness in community detection tasks.

Keywords: Community discovery. Social recommendation. Graph Convolutional Neural Network. Heterogeneous Information Network. Meta path.

1 Introduction

Community detection, a task of grouping similar nodes together, can reveal the features and connections of its members that are different from those in other communities in complex network [1], it is an important problem in network science due to its diverse applications in various domains [2]. Detecting communities provides a deeper understanding of the network's functionality and structure, revealing hidden patterns within the network [3]. Many algorithms for community detection have been proposed, a majority of which mainly rely on network topology information [4], such as modularity optimization [5], hierarchical clustering [6], and label propagation [7]. These algorithms primarily focus on homogeneous networks. In real-world scenarios, a majority of complex networks are heterogeneous and incorporate a variety of node and edge types [8].However,most of the existing community detection methods are designed for homogeneous networks and cannot effectively handle heterogeneous structures and rich semantic information [9]. In recent years, the technique of deep learning has drawn a great deal of attention and has been demonstrated to have great power on a wide variety of problems, which also offered new perspectives for heterogeneous network community detection. GNN (graph neural networks) based on deep learning, can learn the vector representation of entities by utilizing a heterogeneous information network to fuse semantic and structural features [10]. GNN can effectively perform data fusion on large datasets with high-dimensional and diverse properties, which improves the performance and efficiency of community detection algorithms. Furthermore, social recommendation technology enriches the feature representation of nodes and mitigates the issue of data sparsity in heterogeneous networks by incorporating social relationship information among users as an additional data source. It helps to understand users more comprehensively, and enhance the accuracy of heterogeneous network community detection [11].

To address the issue of inaccurate community detection in heterogeneous networks, this paper proposes a heterogeneous network community detection method (GS-HCD) that combines GCN with social recommendation. The GS-HCD model framework is consists of three parts: user community extension graph construction, user community similarity measurement, and community result recommendation. This model combines the social interaction relationships between users with the similarity between users and communities, using GCN to solve the problem of sparse interaction data between users and communities. At the same time, it introduces social recommendation methods to personalize community recommendations to target users, solving the problem of single connections between users and communities and improving the precision of heterogeneous network community detection. The primary contributions of this article are as follows:

(1) By using meta paths to measure the similarity between users and communities, and introducing social recommendation methods to measure the different connection methods between users and communities, the problem of a single connection between users and communities has been solved.

(2) A heterogeneous network community detection method combining GCN with social recommendations is proposed, which introduces a labelling mechanism and utilizes message passing GCN to process and train heterogeneous information network data, capturing deep nonlinear relationships between nodes and alleviating data sparsity problems.

(3) Comparing with multiple representative methods on three datasets, the results show that the GS-HCD model has good performance in precision, recall, and F-score. The results are superior to other methods in F1 and AUC, and can effectively detect heterogeneous network communities.

2 Related Work

Due to the complexity of heterogeneous networks and the computing power of computers, traditional heterogeneous community detection methods tend to focus on simpler structured binary networks and multi-dimensional networks. The core idea of the heterogeneous network community detection method based on binary networks is to use modularity to measure the quality of community detection [12]. The higher the modularity, the tighter the connections within the community, and the sparser the connections between communities. By continuously adjusting the community division, the modularity is maximized. Liu et al. [13] assigned users to different communities based on modular scores. Use split clustering method to find the optimal community, which divides the graph until the maximum modularity score is reached. Reference [14] proposed an implementation on a weighted network with irregular topology, which is based on a stepwise path detection strategy, where each step finds a link to increase the overall strength of the detected path. However, in real-world networks, there are significant differences in data types, and traditional methods often struggle to effectively integrate these different types of data, resulting in information loss or insufficient utilization. The large amount of nonlinear structural information makes the traditional heterogeneous methods unsuitable for practical applications. graph neural network models have powerful modeling capabilities that can capture complex structural information in heterogeneous networks [15]. The GNN algorithm uses a recurrent graph neural network for maximizing modularity [16]. Wu et al. [17] proposed a heterogeneous Q&A community detection method based on graph neural networks to detect heterogeneous communities in community Q&A. The mechanism of automatically discovering hidden semantic communities from user social behavior lays the foundation for community construction and Q&A platforms.

Although graph neural networks have made significant progress in discovering heterogeneous network communities, they require high quality and diversity of data. Lack of sufficient labeled data can lead to performance degradation and inaccurate results in heterogeneous network community detection [18]. Social recommendations can introduce various information such as users' social relationships, interests, and preferences. In the process of community detection, this additional information can serve as important clues, helping to improve the precision of community segmentation. Satuluri et al. [19] proposed a community detection method based on Metropolis Hastings sampling for heterogeneous recommendations on Twitter, which is more accurate and faster than existing alternatives. Deng et al. [20] proposed a recommendation model for information element path discovery based on heterogeneous information networks and deep learning, which incorporates similarity into the deep learning model to predict community similarity. Social recommendations can improve community detection performance, but they also face some challenges. When new users or communities join, social recommendation systems may not be able to provide effective recommendations due to a lack of sufficient historical data [21]. This will also affect the performance of community detection, especially in the initial stage, where it is difficult to accurately classify the communities to which new users or communities belong [22]. By making reasonable use of social recommendation systems and addressing issues such as data cold start, the advantages of social recommendation can be fully utilized to improve the precision of community detection.

3 GS-HCD Model

3.1 Problem Definition

Definition 1: Heterogeneous Information Network [23] (HIN). An information network is a directed graph G = (V, E) with entity type mapping φ: V→A and link mapping ψ: E→R. Each object v ∈ V belongs to a specific entity type φ(v)∈A, and each link e ∈ E belongs to the set of relationship types R: The specific relationship type in ψ(e)∈R. If the object type |A|>1 or the relationship type |R|>1, the information network is called a heterogeneous information network. The user community heterogeneous information network shown in **Fig. 1**, with the circle on the right representing the community that attracts users on the left through direct or indirect contact. In HIN, project and tag entity types can serve as links connecting users with community entity types and forming various semantic paths.

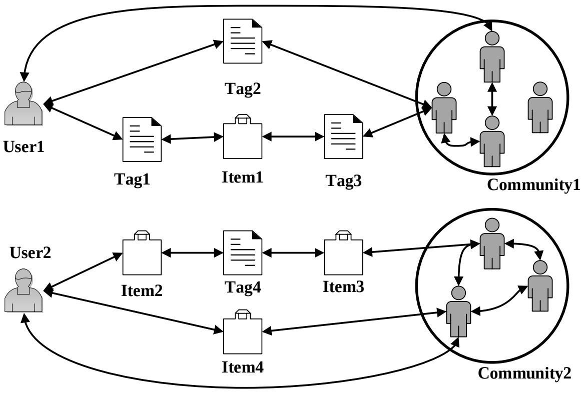

Fig. 1. User community HIN diagram

Definition 2: The network pattern [24]. The network pattern provides a clear understanding of the types of network entities(T) and the relationships(R), represented as TG = (T, R).

Define 3: Meta Path [23]. A meta path is a path defined on a network pattern that links two types of entities, formally defined as $P_m\colon A_1^{\overrightarrow{R_1}}A_2^{\overrightarrow{R_2}}\cdots^{\overrightarrow{R_n}}A_{n+1}$。 The meta path describes a type of connection between entities of types A1 and A_n+1, where different meta paths represent different semantic relationships. For example, the meta path UCU indicates that two users belong to the same community. The meta path UUC represents that a user can recommend their interest community to another user by following them. The meta path UIUI indicates that users who are interested in the same project as the user are also interested in other projects.

3.2 Model Architecture

The GS-HCD model, as shown in **Fig. 2**, consists of three modules: user community extension graph construction, user community similarity measurement, and community result recommendation.

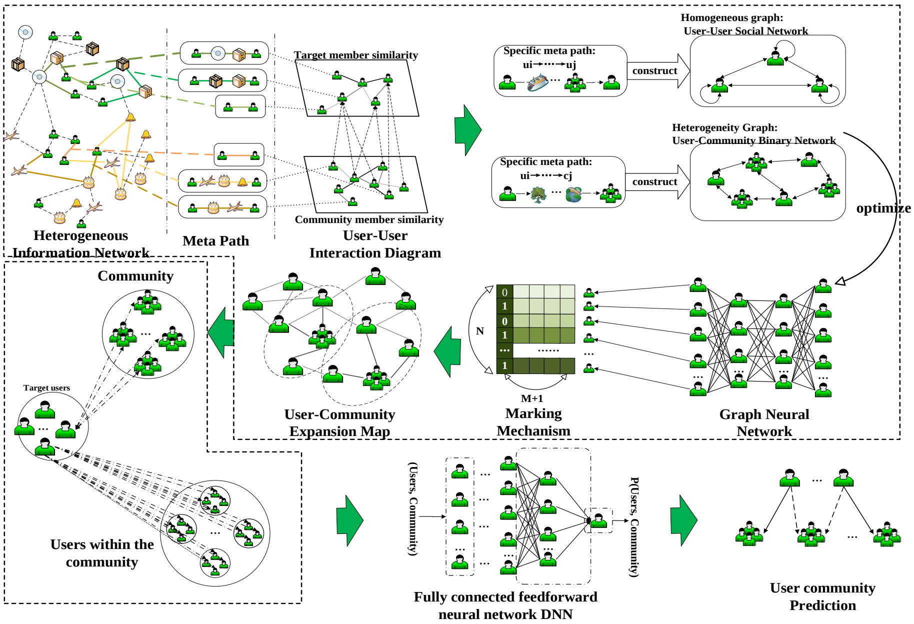

Fig. 2. GS-HCD model framework diagram

Firstly, a network model for heterogeneous information networks is constructed, which transforms the heterogeneous information network into a user-user social network G_{uu} and a user community binary network G_{uc} based on predefined meta paths. The GCN model architecture is adjusted to achieve network optimization, and a labeling mechanism is introduced to fuse the optimized two networks to construct a user community extension graph G_{cu}. Secondly, select meta paths that meet the criteria between the target user node and candidate community nodes in the extended graph, and use the AvgSim similarity metric to calculate similarity based on meta paths, forming candidate node pairs. Finally, using user community candidate nodes as input, the vector information is fed into a deep learning based social recommendation model for learning, capturing the dynamic changes in user interests in the social network and recommending the optimal community for users.

3.3 User Community Expansion Diagram Construction Module

This module selects effective meta paths, and constructs a user-user social network G_{uu} and a user community binary network G_{uc}. Train and optimize G_{uu} and G_{uc} through GNN and labeling mechanisms, and integrate the two networks to construct the user community extension graph G_{cu}.

Build user-user social network G_{uu}

Define a network pattern to represent the types and relationships of entities involved in heterogeneous networks, select a specific meta path where both the starting and ending nodes are user entities, and construct a user-user social network G_{uu}.

Building a user community binary network G_{uc}

If there are community entity types, select the entity meta path from the starting node being the user to the ending node being the community, and construct the user community binary network G_{uc}. If there is no community entity type, the heterogeneous information network needs to be divided into initial communities before constructing the user community binary network G_{uc}. The specific steps are as follows:

Construct an initial set of seed nodes

Transforming heterogeneous networks into homogeneous networks, that is, selecting a meta path from the user type node to the user social homogeneous subgraph G that ends at the user type node. The initial seed node is a node whose degree of partition is not less than the degree of all nodes in its neighborhood. The set of direct neighbors of a node is called its neighborhood. Considering that small degree nodes in sparse regions cannot become seed nodes, the small degree nodes and their neighborhoods are deleted from the candidate seed set until the candidate seed set is empty, and the initial seed node set is selected.

Dynamically adjust the seed node set

Dynamically adjust the seed node set. After obtaining the initial set of seed nodes, dynamically adjust the seed nodes in the homogeneous network. To eliminate the influence of initial seed node selection on community detection results, the seed nodes are extended to communities, and the seed nodes of the community are updated to nodes with high consistency in community ownership within the neighborhood. If the nodes in the neighborhood of the node belong to the same community, the community ownership of the node is relatively determined. Identifying nodes with high certainty of community ownership as seed nodes can help improve the quality of community detection.

Forming the initial community

Forming the initial community. Use the adjusted seed nodes and their neighborhoods as the initial community for community detection. Re select the meta path from the starting node as the user entity to the ending node as the community entity, and construct the user community binary network G_{uc}.

GCN Optimization Network

Using node vectors to represent initialized user and community information, meta paths as features, iteratively training the user-user social network G_{uu} and user community binary network G_{uc} through message passing graph convolutional neural networks, aggregating information from their neighbors, and updating and optimizing node representations. Therefore, the vector representation of each user node can be represented as shown as Eq. (1) after going through k layers.

$$H^{(k+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(k)}W^{(k)}) \tag{1}$$

$$\tilde{A} = A' + I \tag{2}$$

$$\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij} \tag{3}$$

Among them, $H^{(k+1)}$ represents the feature vector of the node in the k-th+1st layer after k-layer convolution. $H^{(k)}$ represents the feature vector of the node in the k-th layer. $W^{(k)}$ represents the parameters of the k-th layer convolution. $\sigma(\cdot)$ represents the activation function. I is the identity matrix, where the diagonal is 1 and all other elements are 0. $\widetilde{D}^{-\frac{1}{2}}$ is the power of the diagonal elements, $\tilde{A}$ is the matrix obtained by adding self-loops to each node, can be calculated by Eq. (2). $\widetilde{D}$ is the plus 1 of each element in the D matrix, can be calculated by Eq. (3), which are Laplacian matrix. D is a diagonal matrix, where all other elements on the diagonal are 0. $\widetilde{D}_{ii}$ represents the number of nodes connected to node i. $\tilde{A}_{ij}$ is the value of the i-th row and j-th column elements in the matrix.

Users are influenced by adjacent user nodes and community nodes. By integrating the user-user social network G_{uu} with the user community binary network G_{uc}, a user community extension graph G_{cu} is constructed.

Marking Mechanism

The use of labeling mechanisms in message passing graph neural networks can enhance the feature representation of nodes in the model, and facilitate the generation of different heterogeneous information network subgraphs based on the entity types of neighboring nodes in heterogeneous information. Assign a one-dimensional label vector to each user node to specify its membership information. Given node x and node y, define the labels for x and y. Node types are not limited and can be either user or community types. The expression is shown as Eq. (4).

$$Label(x, y) = \begin{cases} 1, & if x = y \\ 0, & if x \neq \mathrm{y} \end{cases} \tag{4}$$

Assuming *x* and *y* are common variables for both target users and community users, if *a* and *b* are the same node, the value of *Label* (*x, y*) is 1. If *a* and *b* are different user nodes, the value of *Label* (*x, y*) is 0, and the similarity between target users and community users is 0.

By iteratively updating the graph neural network and labeling mechanism, the final embedding vector of user nodes is obtained, and the embedding vector is propagated in the user community extension graph G_{cu} to preserve rich user-user interaction information, which can generate embedding vectors for community nodes.

3.4 User Community Similarity Measurement Module

This module selects appropriate meta paths based on the user community extension graph G_{cu}. Measure the similarity between unallocated community target users and target communities based on meta paths, and obtain target user target community node pairs.

Similarity Between Target Users and Community Users

AvgSim can measure the similarity between entities through meta paths of any length without the need for additional work. Determining the similarity between two users and the connected meta path can be calculated using AvgSim, which is circulated as Eq. (5).

$$Sim(u_i, u_j | P_m) = \frac{1}{2}[Label(u_i, u_j | P_m) + Label(u_j, u_i | P_m^{-1})] \tag{5}$$

$$Label(u_i, u_j | P_m) = \begin{cases} \frac{1}{|O(u_i|R_l)|} \sum_q^{|O(u_i|R_l)|} Label\left(O_q(u_i|R_l), u_j \middle| A_1 \overset{R_2}{\rightarrow} \dots \overset{R_l}{\rightarrow} User\right), & if |O(u_i|R_l)| \neq 0 \\ 0, & if |O(u_i|R_l)| = 0 \end{cases} \tag{6}$$

Among them, $O(u_i|R_l)$ represents the set of neighbors reached by u_i through R_1, $|O(u_i|R_l)|$ represents the size of the set, and $O_q(u_i|R_l)$ represents the *q*-th neighbor in the set. As shown as Eq. (6), if *Label* (*a*, *b*) is the same entity, its value is 1, otherwise it is 0. Because if *a* and *b* are the same entity and are directly connected to themselves, the similarity is 1 because they are of the same type, otherwise the similarity is 0 because they are different types of entities. In *sim* (•), the former tests the probability of u_i reaching u_j through P_m, while the latter measures the probability of u_j reaching u_i through P_m^{-1}, and then takes the average. This formula can symmetrically measure the similarity between u_i and u_j.

Similarity Between Target Users and Candidate Communities

Secondly, based on the similarity between the target user and the community users in the target community, the proximity between user u_i and community *c* is measured through the most similar community users in the target community and the meta path P_m, expressed as Eq. (7).

$$Similarity(u_i, c|P_m) = \frac{1}{|U_{c,i}|}\sum_{u_j \in U_{c,i}}[Sim(u_i, u_j|P_m)] \tag{7}$$

Among them, $U_{c,i}$ is the aggregation of community members that are most similar to u_i.

The above AvgSim similarity metric can also be directly used to calculate the proximity between users and communities, as shown as Eq. (8).

$$Sim(u_i, c_j|P_m) = \frac{1}{2}[Label(u_i, c_j|P_m) + Label(c_j, u_i|P_m^{-1})] \tag{8}$$

$$Label(u_i, c_j|P_m) = \begin{cases} \frac{1}{|o(u_i|R_l)|}\sum_q^{|o(u_i|R_l)|} Label\left(O_q(u_i|R_l), u_j\middle|A_1 \overset{R_2}{\rightarrow} \dots \overset{R_l}{\rightarrow} User\right), if|O(u_i|R_l)| \neq 0 \\ 0, \quad if|O(u_i|R_l)| = 0 \end{cases} \tag{9}$$

Among them, u_i and c_j are the source and target entities of the meta path P_m, respectively, and P_m^{-1} is the reverse meta path of P_m. Label (a, b), which is circulated by Eq. (9), is as same as Eq. (6).

3.5 Community Results Recommendation Module

This module trains a fully connected feedforward neural network (DNN) with multiple hidden layers on the target user candidate community nodes obtained, outputs the probability of the target user joining the target community, and applies the final prediction results to heterogeneous network community detection.

Let a^t be the output vector of the *l*-th layer in the *T*-layer DNN, a^{t-1} be the output vector of the *t-1*st layer in the *t-1*-layer DNN, a^1 be the input layer, and a^T be the output layer. The calculation Eq. (10) is as follows:

$$a_v^t = \phi_v^t(\sum_f \omega_{vf}^t a_f^{t-1} + b_v^t) \tag{10}$$

Among them, $\phi_v^t(\cdot)$ is the nonlinear activation function, $\omega_{vf}^t$is the matrix value of the *v*-th column and *f*-th row of the training weights, $b_v^t$is the biased *v*-th column matrix vector.

Using ReLU as the activation function for the hidden layer to avoid gradient vanishing, and using Sigmoid function as the non-linear activation function for the output layer, the output values can be constrained within the range of (0,1), thereby obtaining the relationship between the target user and the target community. Communities with Top-K prediction probabilities are recommended to the target user for matching, and whether the target communities overlap can be determined, achieving heterogeneous network community detection.

4 Experiment and Evaluation

4.1 Dataset

The selected datasets include the DBLP dataset with positive sample sparsity, as well as CiteULike and Last.fm datasets with a large amount of community information. The meta path method defined by previous researchers was used to evaluate the meta path based approach. The related meta path selection information for the datasets is shown in **Table 1**.

Table 1. Dataset meta path selection

Dataset	Node information	Relationship between nodes	Meta path selection
DBLP	author(A):4057 paper(P):14328 conference(C):20 term(T):7723	P-A:19645 P-C:14328 P-T:85810	APA APCPA APTPA
CiteULike	user(U):1016 community(C):401 article(A):6881 tag(T):6193	U-C:2940 U-A:16760 U-T:50481 A-T:32194	UCU、UAU、UTU UATAU UTATU
Last.fm	user(U):10013 community(C):1000 item(I):9651 tag(T):8254	U-U:154125 U-C:54832 U-I:1039875 U-T:796241 I-T:5697021	UU、UCU UIU、UTU UITIU UTITU

4.2 Comparing Baselines

The GS-HCD model integrates social recommendation technology into heterogeneous network community detection, outputs the probability of target users joining the community, and then divides the community. Therefore, evaluations will be conducted separately based on the community segmentation results and the user community recommendation situation of the experimental model.

Comparison of Community Division Results with Baseline

The research objective is to partition heterogeneous network target nodes into communities. In order to verify that the application of recommendation in community detection can improve the performance of community detection, the proposed GS-HCD will be compared with the state-of-the-art community detection baseline methods: unsupervised heterogeneous method Metapath2vec [25], HetGNN [26], DMGI [27], and semi supervised heterogeneous method HAN [28].

Metapath2vec formalizes the random walks of meta paths, constructs heterogeneous neighbors of nodes, and uses a heterogeneous skip cell model to achieve node mapping.

HetGNN can efficiently combine heterogeneous structure information with heterogeneous content information to achieve efficient collaboration among nodes. DMGI not only increases the mutual information between blocks in the graph, but also enhances the overall expression of the graph. HAN has given sufficient attention to the importance of nodes and meta paths by studying the focus points at both the node and semantic levels.

Data preprocessing: For methods based on random walks (Metapath2vec, HetGNN), set the number of walks for each node to 40, the walk length to 100, and the window size to 5. Test all meta paths for DMGI and HAN, and use the settings from their original paper for other parameters. For fair comparison of all methods, the embedding dimension is set to 64, randomly run 10 times, and report the average result. For each dataset, only use the original attributes of the target node and assign one hot ID vectors to other types of nodes as needed.

Recommended Community Model Comparison Baseline

Compare the model with five recommended models applied to the community: CF、CB、GB、MF、HIN [20]. Further validate the quality of the model.

CF model: using direct interaction between users and communities for recommendation. CB model: using labels to represent community features and user preferences. GB model: Representing data as homogeneous graphs, entities as nodes of the same type, and relationships as edges of the same type, thus transforming suggestions into link prediction problems. MF model: Aggregate various information into a combined user item matrix, factorize the matrix to obtain a low dimensional dense vector representing the item. HIN model: Integrating heterogeneous information and merging similarities into deep learning models to generate recommendations.

Data preprocessing: For CB and GB methods, there are no parameters that need to be tuned. The validation and training sets are used as inputs for these two methods, and the user recommended community list in the test set is used as output. To determine the parameters of the other four methods, the training set is first used as input for these methods, and a community list is recommended to the users in the validation set. By changing these parameter values, different recommendation lists can be generated, so different F-scores can be calculated by comparing the recommended communities with the communities that users join in the validation set. When the F-score reaches its maximum, the optimal solution for the parameters is obtained. Once the parameters are determined, the validation set and training set will be used as inputs for these methods, and the community will be recommended to the users in the test set. By comparing the recommended communities with the communities where users in the test set joined, the recommendation performance of these methods can be measured.

4.3 Evaluation Indicators

Precision: Regarding the judgment result, the proportion of true positive instances (TP) among the samples predicted as positive class (TP+FP) is 1. The calculation is as Eq. (11).

$$Precision = \frac{TP}{TP+FP} = \frac{\text{Positive samples correctly predicted}}{\text{Predicted } Positive \text{ samples}} \tag{11}$$

Recall: Recall, also known as recall rate, is better if the recall is higher. The proportion of correctly judged positive instances (TP) in the total positive instances (TP+FN) for the sample. The calculation equation is as Eq. (12).

$$Recall = \frac{TP}{TP+FN} = \frac{\text{Positive samples correctly predicted}}{Total\ number\ of\ Positive \text{ samples}} \tag{12}$$

F1-score: Taking into account both precision and recall, it is the harmonic average of the two. The closer the F1 score is to 1, the better the overall performance of the model in terms of recall and precision, while in Precision or Recall, once one term approaches 0, the F score will be very low. The calculation equation is as Eq. (13).

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{13}$$

NMI (Normalized Mutual Information): The higher the value of NMI, the better the correlation between clustering effect and real categories.

$$NMI(\Omega, C) = \frac{2 \cdot I(\Omega; C)}{H(\Omega) + H(C)} \tag{14}$$

Among Eq. (14), I (Ω; C) represents mutual information, and H (Ω) and H (C) represent the information entropy of Ω and C, respectively. The calculation equation is as Eq. (15).

$$I(\Omega; C) = \sum_k \sum_j P(\omega_k \cap c_j)\, log \frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_j)} = \sum_k \sum_j \frac{|\omega_k \cap c_j|}{N}\, log \frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_j)} \tag{15}$$

$P(\omega_k)$, $P(c_j)$, and $P(\omega_k \cap c_j)$ can be respectively regarded as the probability that the sample belongs to cluster ω_k, category c_j, and both.

ARI (Adjusted Rand Coefficient)

$$ARI = \frac{\sum_{i,j} C_{n_{ij}}^2 - [\sum_i C_{a_i}^2 \sum_j C_{b_j}^2]/C_n^2}{C_n^2} \tag{16}$$

Among Eq. (16), $C_{n_{ij}}^2$represents the logarithm of samples belonging to both categories i and j in both the real category and clustering results, $C_{a_i}^2$ and $C_{b_j}^2$ represent the logarithm of samples belonging to categories i and j in the real category and clustering results, respectively, while C_n^2 represents the combination of all sample pairs. The higher the ARI value, the more consistent the clustering results are with the real situation.

Roc and AUC curves: AUC (Area Under Curve) is defined as the area enclosed by the coordinate axis under the ROC curve, and the value of this area will not exceed 1. Moreover, since the ROC curve is generally above the line y=x, the range of AUC values is between 0.5 and 1. The closer the AUC is to 1.0, the higher the authenticity of the detection method; When it is equal to 0.5, the authenticity is the lowest and there is no practical value.

4.4 Comparative Analysis of Community Division

In this study, the K-means algorithm was used to achieve learning for each node, NMI and ARI are used to evaluate its clustering performance. On this basis, simulate this process more than 10 times to reduce instability caused by changes in initial values. As can be seen from the data visualization in **Fig. 3** and **Table 2**.

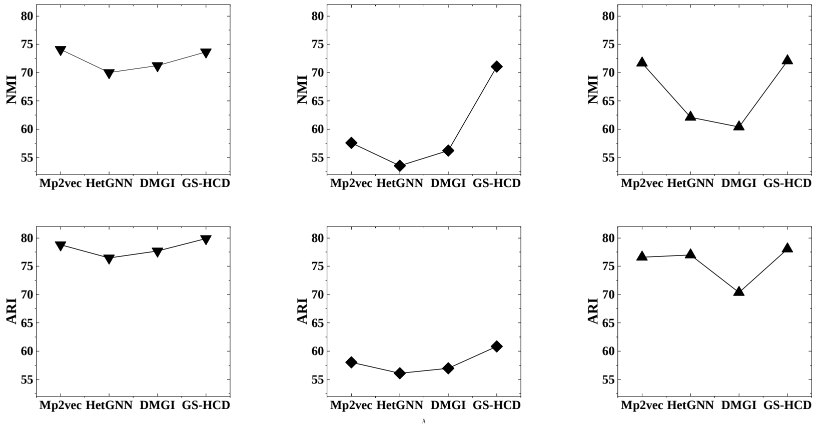

Fig. 3. Visualization of clustering effect

The Metapath2vec model (hereinafter referred to as Mp2vec) maps nodes in heterogeneous networks to a single vector space, which may result in information loss and affect the model's understanding and representation of the network. Although the HetGNN model avoids information loss caused by different node types or differences in the number of neighbors, simple feature concatenation or linear transformation may not fully capture the complex relationships and interactions between features, which affects the precision of node feature representation. The DMGI model minimizes the divergence between node embeddings of specific relationship types, allowing node embeddings of different relationship types to coordinate with each other. However, its ability to handle multimodal data may be limited, making it difficult to fully utilize the rich information in multimodal data to improve the quality of node embeddings.

Table 2. Clustering performance of different datasets under different baselines

Datasets	DBLP		CiteULike		Last.fm	
Metrics	NMI	ARI	NMI	ARI	NMI	ARI
Mp2vec	74.10	78.80	57.59	58.03	71.66	76.58
HetGNN	70.01	76.45	53.54	56.08	62.12	77.01
DMGI	71.22	77.70	56.21	56.98	60.39	70.34
GS-HCD	**73.64**	**79.88**	**71.06**	**60.85**	**72.08**	**78.05**

Selecting a small number of labeled nodes (such as 20) can test the generalization ability of the model in data scarcity scenarios, simulating the high cost of labeling certain categories in reality. Gradually increasing to 40/60 nodes, the improvement trend of model performance with the increase of training data volume can be observed to verify data efficiency. Selecting 20/40/60 nodes for each category ensures category balance and avoids bias in the model due to excessive data in certain categories. A larger validation/test set (1000 nodes) can reduce the variance of evaluation results and ensure higher confidence in metrics such as accuracy and F1 score. The training set and validation set are used to adjust the parameters of the proposed method and benchmark. On this basis, multiple algorithms are used to generate recommendation effects, and the F-scores and AUC values of each algorithm are obtained by comparing them with experimental data. The experimental results are shown in **Table 3**, **Table 4**, **Table 5**.

Table 3. The F-scores and AUC values on DBLP dataset.

Dataset	Metric	Split	Mp2vec	HetGNN	HAN	DMGI	**GS-HCD**
DBLP	Ma-F1	20	89.65±0.2	90.65±2.1	90.54±1.3	89.25±0.2	**91.83±0.2**
		40	90.26±0.3	89.15±1.9	89.69±0.2	87.69±0.4	**91.04±0.3**
		60	92.43±0.4	88.32±1.0	89.99±0.4	89.14±0.6	**91.30±0.2**
	Mi-F1	20	90.25±0.1	89.68±1.0	90.65±1.2	88.02±0.6	**92.87±0.5**
		40	90.14±0.2	89.21±0.9	89.21±0.9	89.36±0.4	**91.70±0.6**
		60	91.36±0.6	90.31±0.8	90.44±0.8	90.29±0.5	**91.45±0.2**
	AUC	20	89.65±0.2	90.65±2.1	90.54±1.3	89.25±0.2	**91.83±0.2**
		40	90.26±0.3	89.15±1.9	89.69±0.2	87.69±0.4	**91.04±0.3**
		60	92.43±0.4	88.32±1.0	89.99±0.4	89.14±0.6	**91.30±0.2**

Table 4. The F-scores and AUC values on CiteULike dataset.

Dataset	Metric	Split	Mp2vec	HetGNN	HAN	DMGI	**GS-HCD**
CiteU Like	Ma-F1	20	73.14±0.2	82.64±0.6	83.57±1.2	87.26±3.5	**89.20±1.5**
		40	77.69±0.4	78.74±0.9	87.36±0.7	88.02±2.4	**88.65±0.9**
		60	75.78±0.1	83.25±0.7	86.44±0.9	87.36±2.1	**87.59±0.8**
	Mi-F1	20	76.98±0.1	86.69±0.8	87.69±0.8	88.95±1.2	**89.01±1.2**
		40	81.45±0.2	83.14±0.5	92.44±0.6	92.58±0.5	**93.26±1.3**
		60	78.25±0.1	85.31±0.7	91.01±1.2	92.47±0.6	**92.58±0.5**
	AUC	20	81.31±1.0	90.14±1.2	93.10±0.3	**94.20±0.8**	93.98±0.6
		40	85.16±0.8	89.11±1.3	95.36±0.5	**96.13±2.6**	95.86±0.4
		60	84.25±0.5	91.36±1.6	94.01±1.2	95.18±2.1	**95.80±0.9**

Table 5. The F-scores and AUC values on Last.fm dataset.

Dataset	Metric	Split	Mp2vec	HetGNN	HAN	DMGI	**GS-HCD**
Last.fm	Ma-F1	20	64.54±0.2	80.24±0.6	86.03±1.3	89.98±0.6	**90.31±1.8**
		40	75.65±0.3	88.36±0.8	92.15±1.6	91.45±0.5	**92.64±1.0**
		60	72.14±0.2	87.14±0.9	91.10±0.9	90.36±0.9	**91.59±0.6**
	Mi-F1	20	80.26±0.1	80.36±0.7	88.06±2.6	89.99±3.4	**90.64±0.9**
		40	89.26±0.1	85.18±0.8	90.10±2.4	90.58±2.6	**91.47±0.8**
		60	84.24±0.2	87.34±1.2	89.35±2.3	90.02±1.5	**90.98±0.6**
	AUC	20	89.15±1.5	90.21±1.3	88.21±2.5	**91.24±0.2**	89.78±0.5
		40	91.21±1.0	87.71±2.3	91.36±1.9	**92.03±0.9**	91.59±0.9
		60	90.62±0.9	89.33±2.1	90.14±2.6	**92.15±1.7**	91.76±0.7

Mp2vec embeds all types of nodes into the same vector space, which may result in inaccurate representation of different types of nodes in vector space, as different types of nodes may have different features and semantics. HetGNN can capture information from different types of neighbors, and avoid information loss caused by differences in node types or the number of neighbors, but simple feature concatenation or linear transformation may not fully capture the complex relationships and interactions between features, affecting the precision of node feature representation. DMGI minimizes the divergence between node embeddings of specific relationship types, allowing node embeddings of different relationship types to coordinate with each other. However, its ability to handle multimodal data may be limited, making it difficult to fully utilize the rich information in multimodal data to improve the quality of node embeddings. When dealing with heterogeneous graphs, HAN may still overlook some high-order structural information, resulting in a less comprehensive understanding of the graph structure.

The results show that compared with the semi supervised HAN algorithm, GS-HCD performs well on all samples and all regions. The experimental results show that the performance of GS-HCD is better than that of most DMGI, and the performance of DMGI is better than other benchmarks. Because DMGI can cluster a large number of target nodes as overlapping nodes, GS-HCD selects the community with the highest probability and assigns it to the target nodes, mostly in non-overlapping communities. Therefore, in some evaluation metrics, it is slightly lower than DMGI. The GS-HCD algorithm performs better than the HAN algorithm that utilizes label information in various situations. The use of connections among community members can effectively improve the performance of recommendation systems on heterogeneous networks.

4.5 Performance Comparison of Recommendation Community Models

GS-HCD method is compared with five baselines that apply recommendations to communities. The comparison results are shown in **Fig. 4** and **Fig. 5**.

From the visualization results of the table and graph, it can be concluded that the CF model only uses member information to generate recommendation lists, the CB model provides services to users and communities based on tags related to users, and the GB model models entities and relationships as homogeneous graphs. All three models operate on homogeneous graphs, indicating that incorporating heterogeneous information into community recommendations is effective.

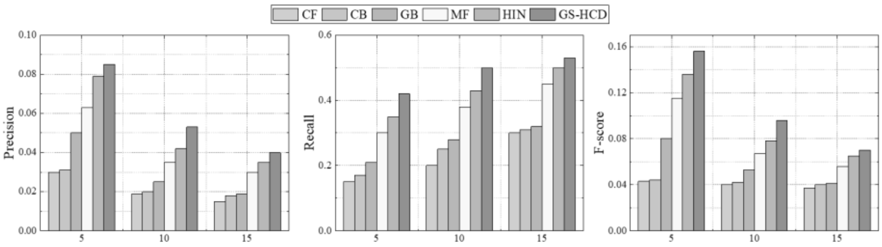

Fig. 4. Visualization on CiteULike dataset

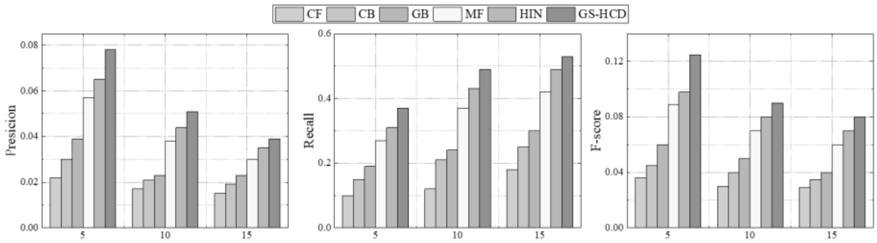

Fig. 5. Visualization on Last.fm dataset

Both the MF model and the HN model are performed on heterogeneous datasets, but the MF model factorizes the matrix and the HN model collaboratively filters the experimental results. Compared with the GS-HCD model that uses deep learning, in extremely sparse data, it can cause data cold start and seriously affect prediction precision. Compared with the above baseline, the GS-HCD model has better recommendation performance in precision, recall, and F-score. This method is robust on different numbers of recommendations and datasets. When the number of recommendations is small, the improvement is higher, which is advantageous because users tend to view a limited number of recommendations.

5 Conclusions

This article proposes a new community detection method based on GCN and social recommendations. selecting appropriate meta path information from heterogeneous information networks for community recommendation. At the same time, the use of deep learning technology effectively alleviates the problem of data sparsity, and social recommendation solves the problem of single connection between users and communities, improving the precision of community segmentation. The experimental results on three real datasets show that the GS-HCD model is more accurate in community detection on heterogeneous information networks and can effectively partition communities. By recommending communities that users are interested in, it is possible to more effectively discover information of interest in related communities.

In the future, the detection of heterogeneous network communities can be extended from different directions, and more accurate dynamic demonstration models can be established to describe the process of heterogeneous network communities changing over time. At the same time, it can combine big data technology to collect and analyze the real-time flow of users within the community, improve the efficiency of community information dissemination, and strengthen the social atmosphere within the community.

References

1. Su X, Xue S, Liu F Z, et al. A Comprehensive Survey on Community Detection with Deep Learning. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(4): 4682-4702.
2. T. Chakraborty, N. Park, A. Agarwal, and V. S. Subrahmanian. Ensemble Detection and Analysis of Communities in Complex Networks. ACM/IMS Transactions on Data Science, 2020, 1(1): 2:1-2:34.
3. D.Jin, X.Wang, D. He, and W. Zhang. Robust Detection of Link Communities with Summary Description in Social Networks. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(6): 2737-2749.
4. D.Jin, Z.Z.Yu, P. F. Jiao, et al. A Survey of Community Detection Approaches: From Statistical Modeling to Deep Learning. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(2): 1149-1170.
5. E. Jokar, M. Mosleh, and M. Kheyrandish, et al. GWBM: an algorithm based on grey wolf optimization and balanced modularity for community discovery in social networks. The Journal of Supercomputing, 2022, 78(5): 7354-7377.
6. V. H. Bui, and H. T. Phan. The Computational Complexity of Hierarchical Clustering Algorithms for Community Detection: A Review. Vietnam Journal of Computer Science, 2023, 10(4): 409-431 (2023).
7. Y.B.Yue, G.Y.Wang, J. Hu, and Y. Li. An improved label propagation algorithm based on community core node and label importance for community detection in sparse network. Applied Intelligence, 2023, 53(14): 17935-17951.
8. Y. K.Wang, D.Jin , D. X.He , K. Musial, and J. Dang. Community Detection in Social Networks Considering Social Behaviors. IEEE Access, 2022, 10: 109969-109982 (2022).
9. Y. Zhao, W. Li, F. Liu, J. Wang, and A. M. Luvembe. Integrating heterogeneous structures and community semantics for unsupervised community detection in heterogeneous networks. Expert Systems with Applications, 2024, 238(Part D): 121821.
10. Wu, Yongliang, et al. "Heterogeneous question answering community detection based on graph neural network." Inf. Sci. 621(2023):652-671.
11. G. Wang, L. Zhou, J. Gong, and X. Zhang. Heterogeneous graph neural network with hierarchical attention for group-aware paper recommendation in scientific social networks. Applied Soft Computing, 2024, 167: 112448.
12. S. Aref, and M. Mostajabdaveh. Analyzing modularity maximization in approximation, heuristic, and graph neural network algorithms for community detection. Journal of Computational Science, 2024, 78: 102283.

13. Z. Akbar, J. Liu, and Z. Latif. Mining social applications network from business perspective using modularity maximization for community detection. Social Network Analysis and Mining ,2021, 11(1): 115.
14. S. Souravlas, A. Sifaleras, and S. Katsavounis. A Parallel Algorithm for Community Detection in Social Networks, Based on Path Analysis and Threaded Binary Trees. IEEE Access 7, 2019, 20499-20519.
15. L. Luo, Y. Fang, X. Cao, X. Zhang, and W. Zhang. CP-GNN: A Software for Community Detection in Heterogeneous Information Networks. Software Impacts, 2021, 10: 100169 (2021).
16. S. Sobolevsky, A. Belyi, Graph neural network inspired algorithm for unsupervised network community detection, Appl. Netw. Sci. 7 (1) (2022)1–19.
17. Y. Wu, Y. Fu, J. Xu, H. Yin, Q. Zhou, and D. Liu. Heterogeneous question answering community detection based on graph neural network [J]. Information Science, 2023, 621: 652-671.
18. K. Sharma, Y. Lee, S. Nambi, A. Salian, S. Shah, S. Kim, et al. A Survey of Graph Neural Networks for Social Recommender Systems. ACM Computing Surveys, 2024, 56(10): 265.
19. V. Satuluri, Y. Wu, X. Zheng, Y. Qian, B. Wichers, Q. Dai, et al SimClusters: Community-Based Representations for Heterogeneous Recommendations at Twitter. KDD, 2020: 3183-3193.
20. W. Deng, W. Du, and C. Han. Incorporating heterogeneous information in deep learning with informative meta-paths for community recommendations. Journal of Information Science, 2023 49(5): 1309-1324.
21. Z. Zhang, F. Song, B. Wang, and C. Dong. Extract Implicit Semantic Friends and Their Influences from Bipartite Network for Social Recommendation. Data Science and Engineering, 2024, 9(3): 278-293.
22. P. R. de Souza, F. A. Durão. Exploiting social capital for improving personalized recommendations in online social networks. Expert Systems with Applications. 2024, 246: 123098.
23. Y. Z.Sun , J. W.Han, X. F.Yan , P.S.Yu, T.Y.Wu. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. Proc. VLDB Endow, 2011,4(11): 992-1003.
24. X. L.Sun,H. F.Lin,K.Xu. A social network model driven by events and interests. Expert Systems with Applications, 2015, 42(9): 4229-4238.
25. Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. KDD 2017: 135-144.
26. C. X. Zhang, D.J.Song, C.Huang, A. Swami, and N. V. Chawla. Heterogeneous Graph Neural Network. KDD 2019: 793-803.
27. C.Park, D. Kim, J.W.Han, and H.Yu. Unsupervised Attributed Multiplex Network Embedding. AAAI 2020: 5371-5378.
28. X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, et al. Heterogeneous Graph Attention Network. WWW 2019: 2022-2032.

✉ liujinglian@cslg.edu.cn