Hindawi
Scientific Programming
Volume 2022, Article ID 9154712, 14 pages
https://doi.org/10.1155/2022/9154712

Research Article

A POI Recommendation Algorithm Based on the Heterogeneous Graph Convolution Network

Yue Li

Guangdong Communication Polytechnic, School of Information, Guangzhou 510650, China

Correspondence should be addressed to Yue Li; liyue@gdcp.edu.cn

Received 3 December 2021; Accepted 29 September 2022; Published 8 October 2022

Academic Editor: Zhihan Liu

Point-of-interest (POI) recommendation is a type of recommendation task, which generates a list of places that users may be interested in. There is a complex heterogeneous graph structure between users and points of interest. The current recommendation algorithms are generally based on Euclidean space data, and the algorithms based on graph structure also generally use homogeneous graph convolution. To solve these problems, the author proposes a heterogeneous graph convolution network algorithm based on hierarchical subgraphs (HGCNR). By constructing user-centered subgraph layers and interest point-cantered subgraph layers, respectively, the author performs heterogeneous graph convolution on different subgraphs to obtain more effective user node information and interest point node information and form recommendation result. Experiments on two public data sets show that HGCNR can effectively improve the recommendation performance of interest points and achieve better recommendation results.

1. Introduction

With the development of Internet technology, the data on the Internet are growing rapidly. These massive data not only facilitate people's life but also make it difficult for people to make accurate choices from massive data quickly. This phenomenon is called information overload, and it is an important problem that needs to be solved. The most common way to prevent information overload is to use information filtering. There are many ways to filter information, including classified index, search engine, and recommendation system.

A recommendation system is a kind of system that provides users with valuable information through effective information filtering. It is also an important method to solve information overload. The advantage of the recommendation system is that it can learn users' personalized preferences according to users' historical access records, help users to discover information they may be interested in, and provide users with personalized information services. The recommendation system has become an indispensable core technology in Internet products [1, 2]. Traditional recommendation methods include collaborative filtering, content-based recommendation, and hybrid recommendation methods. The content-based recommendation method uses items selected by users to find other items with similar attributes for recommendation, but this method requires effective feature extraction. The traditional shallow model relies on manual design features, and its effectiveness and scalability are very limited, restricting the performance of content-based recommendation methods. The collaborative filtering algorithm uses the interactive information between users and projects to recommend users. It is the most widely used recommendation algorithm at present. Because different recommendation algorithms have their own limitations, recommendation results are not ideal when one method is used alone in some scenarios. Choosing different recommendation algorithms to combine to form a hybrid recommendation algorithm is another widely used recommendation model. However, which models to mix and how to mix to produce more effective recommendations are also an important problem to be solved in the hybrid recommendation system [3, 4].

Solving the problems in recommendation systems based on deep learning is a new research direction in recent years.

Deep learning can characterize massive data related to users and items by learning a deep nonlinear network structure. It has a strong ability to learn the essential characteristics of data sets from samples and can obtain deep-level feature representations of users and items. On the other hand, deep learning uses automatic feature learning from multisource heterogeneous data, thereby mapping different data to the same hidden space, and can obtain a unified representation of the data. Combining traditional recommendation methods on the basis of deep learning can effectively utilize multisource heterogeneous data and alleviate data sparseness and cold start problems in traditional recommendation systems. Deep learning has made great progress in many fields such as image processing, natural language understanding, and speech recognition, and it has also brought new opportunities for the research of recommender systems. Recommendation techniques under deep learning include RNN-based recommendation, CNN-based recommendation, and GCN based recommendation [5, 6].

Recommendation systems can be divided into many different forms based on different recommended contents and purposes, including news-oriented recommendation, commodity-oriented recommendation, and video-oriented and music-oriented recommendation, and location-based recommendation is one of the special types. A location-based social network (LBSN) is a variant of the social network, which mainly establishes social connections with other users by collecting user's location information [7]. The check-in data of LBSN contain rich implicit information of users, which can be used to mine the content that users are interested in and realize valuable recommendations. With the development of LBSN, POI recommendation has attracted more attention. POI recommendation and traditional recommendation have some similarities but also have many differences. Recommendations based on interest points should consider many aspects, including geographical location, time factors, and text content [8–10].

In order to solve the POI recommendation and make full use of the heterogeneous data in interest point recommendation, the author proposes a graph neural network-based interest point recommendation algorithm. The model decomposes complex multisource heterogeneous data into different subgraph layers and then obtains richer user node information and interest point node information by performing graph convolution on the subgraphs. The rest of this paper is structured as follows: Section 2 introduces graph neural networks and knowledge related to POI recommendation, Section 3 introduces the proposed model, Section 4 demonstrates the effectiveness of the model through relevant implementations, and Section 5 summarizes the full paper.

2. Related Works

2.1. POI Recommendation. The task of recommending geographic locations that users may be interested in is called point-of-interest (POI) recommendation. POI recommendation is actually a special case of social networks. The core idea of common recommendation algorithms based on social networks is to capture users' interest preferences and friend information by using social network data and make personalized product recommendation, friend recommendation, and conversation recommendation of information flow for users according to the obtained data. Users' social information needs to be considered for geolocation-based social POI recommendation, but social information is not the decisive factor. Compared with traditional content recommendation, POI recommendation has its own uniqueness:

(1) Geographic location will affect the recommended results. Tobler's [11] first law of geography states that all attribute values on a geographic surface are related to each other, but closer values are more strongly related than are distant ones. In the selection of points of interest, users are also more willing to choose a location closer to the current location. Similarly, users may also prefer to visit places closer to their favorite locations. In the POI recommendation, the user's geographic location and the geographic location of the POI will greatly affect the user's decision. Therefore, the influence of geographic location information is the most critical feature that distinguishes POI recommendation from traditional recommendation systems [12].

(2) The point of interest lacks explicit evaluation information. For content recommendation, it is easier to obtain people's evaluation information about the content. Users are willing to write down their experience or ratings of the project after watching movies or listening to music. However, for points of interest, users usually do not express their clear preferences in the check-in process, and the comment information is sparse. Users' preferences for locations usually need to be obtained by converting implicit information.

(3) The user's social relationship will affect the user's choice of POI [13]. When faced with a choice in life, users may turn to their friends for advice, including the choice of points of interest. For example, users may ask their friends which restaurants or tourist attractions are worth going. Many users tend to go to places where their friends have signed in or have gone. In addition, when people travel together, users in the same group may also influence each other's decisions on POI. It can be seen that social factors have a great influence on location recommendation. In general, it can be assumed that friends are more likely to have common preferences. In order to improve the recommendation performance, the traditional recommendation system also considers the user's social relationship for the user's rating prediction. In some research results, fusion of users' social relationships has been proven to improve the performance of the recommendation system.

(4) The user's location is dynamic information that will move and change over time. Time factors will also affect the user's check-in and selection of POI [14].

> For example, the user's check-in place during the working day is generally an office place, and the check-in information for holidays is more likely to be restaurants, theaters, attractions, and other entertainment places. There is also a significant difference between the user's check-in point at 12 : 00 noon and the user's check-in point at 12 : 00 pm [15].

These differences determine that there are huge differences between the POI recommendation based on location-based social networking (LBSN) and the traditional recommendation. The traditional recommendation algorithm is not ideal to solve the recommendation of POI.

2.2. Graph Neural Network. No matter what the traditional linear model or neural network model is, Euclidean structure data are the main data to be processed. However, in the real world, much data is generated from non-Euclidean structure data. In the recommendation system, there is a complex graphic network structure, which is formed by the fusion of various network data, such as the social relationship network between users, the evaluation data network of users to projects, and the hierarchical network data between projects. Because there is no fixed relationship and position between graph data and the structure of the graph node is not uniform, the current neural network model is unsuitable for processing graph data.

Because the association information such as edge and graph structure in the graph network plays an important role in capturing the hidden relationship and mining the eigenvalues of nodes, higher quality recommendation results can be obtained by directly calculating the graph structure data [16, 17]. In the LBSN, there are many complex and diverse data structures, such as the association relationship between users, the association relationship between users and locations, the location relationship between locations, and the relationship between users and comments, and it is a complex heterogeneous network structure, as shown in Figure 1.

A typical graph structure contains two parts of information, one is the information of graph nodes and the other is the structural information between nodes. The attributes of a node include its explicit or implicit characteristics, which are inherent. The structure information describes the association between nodes in the graph structure data. This kind of information not only characterizes the attribute characteristics of nodes but also characterizes the structural expression of the whole graph.

A graph convolutional neural network is a generalization of the convolution neural network in the graph structure, which can learn node features and structure features end to end at the same time [18]. Compared with the traditional convolution network, the graph convolution network has the same properties [19, 20]. The convolution operator in the graph convolutional network is applicable to each node, and the operator is shared everywhere on different nodes. The receptive field of the model is proportional to the number of layers. At the beginning of convolution, each node contains the information of direct neighbors. When the second layer of convolution is calculated, the information of the second-order neighbors can be included so that the information involved in calculation is more sufficient. The receptive field of the model is proportional to the number of convolutional layers. The more convolutional layers, the more information involved in the operation. The graph convolution network also has the characteristics of deep learning. For example, the graph convolution network has a hierarchical structure, and features are extracted layer by layer. The graph convolution network increases the expression ability of the model through nonlinear transformation, and it can realize end-to-end training without defining any rules.

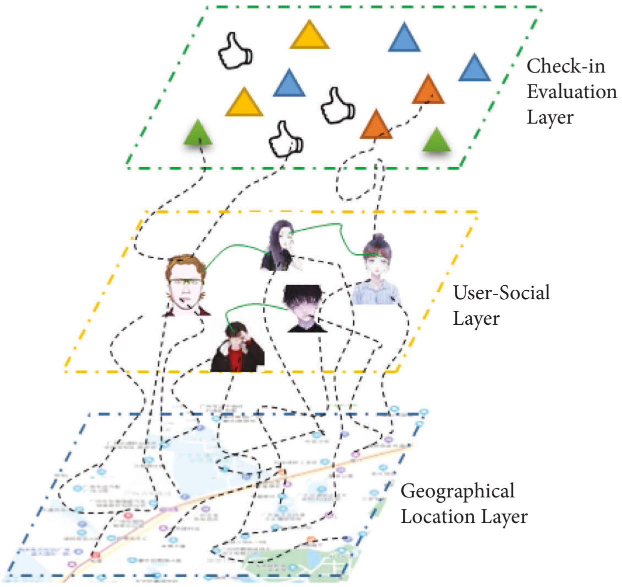


Figure 1: Heterogeneous network structure of LBSN.

The graph convolution network has been used to solve the problem of POI recommendation. However, these research studies mainly use the classical graph convolution method to process the graph data. Firstly, the relationship between users is extracted to obtain user features, and the relationship between interest points is extracted to construct the interest point graph structure. Then, the interest points are extracted by graph convolution. Finally, obtained user features and interest point features are used for interest point prediction. These methods decompose the complex heterogeneous graph into the homogeneous graph through manual design and extraction and then calculate them by graph convolution. The disadvantage of this method is that it will lose the important heterogeneous interaction information and complete graph structure information in the original heterogeneous graph. A heterogeneous network is a kind of network which comprises many different types of nodes and edges. Networks that are composed of heterogeneous information are more common in our real world [21]. For example, an academic citation network is composed of authors, papers, and journals; an e-commerce network is composed of users, commodities, and evaluation; and a film information network is composed of films, directors, and actors, and these are all heterogeneous graph networks. At present, the graph neural network used in POI

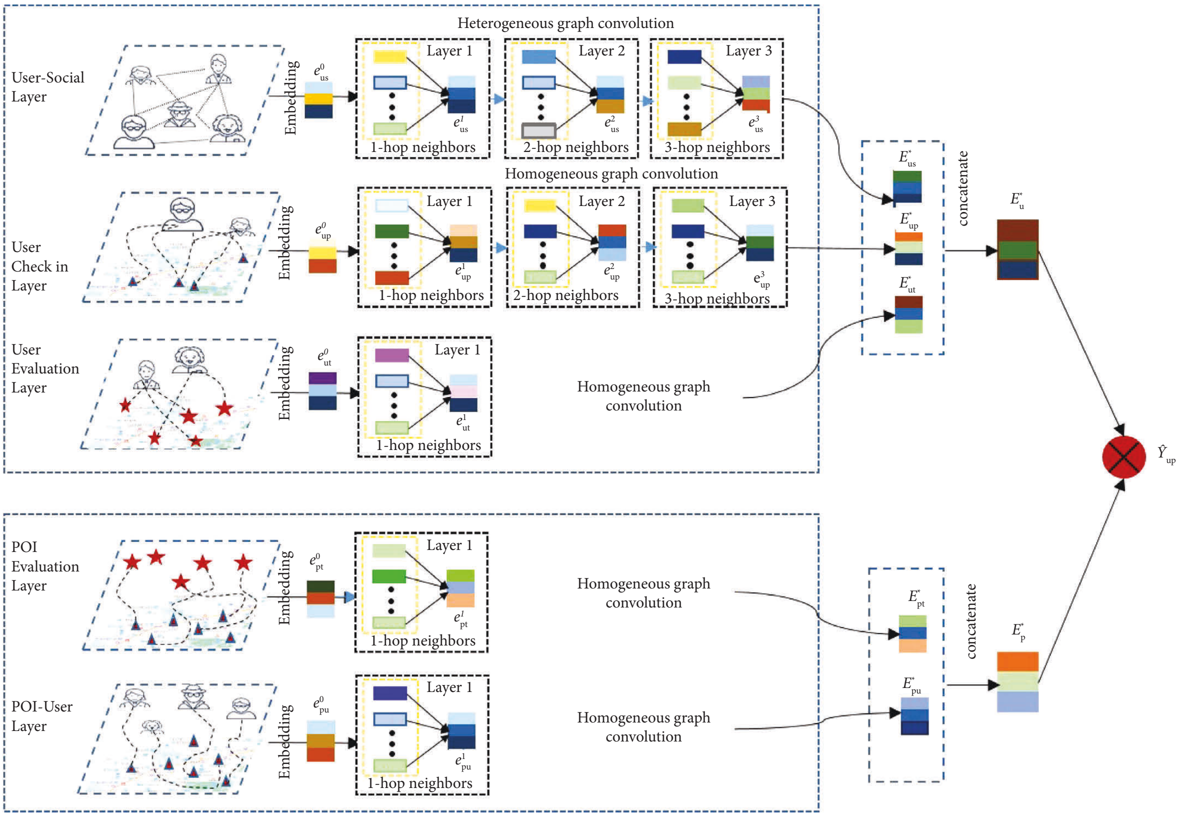


Figure 2: The framework of the model.

recommendation often does not use the heterogeneous graph network to represent node characteristics. In an original user-POI network, a central node often contains many types of neighbor nodes. Compared with the central node, its neighbor nodes may be homogeneous nodes or heterogeneous nodes. Different types of neighbor nodes have different effects on the characteristic calculation of the target node. If heterogeneous networks are disassembled into homogeneous networks for calculation, much important information of nodes and structures will be lost, and the performance of recommendation will be reduced [22].

3. POI Recommendation Based on the Heterogeneous Graph Convolution Network

Although there are many deep learning models that can achieve good recommendation performance, most of these models cannot achieve good results when applied to POI recommendation. The main reason is that in POI recommendation, it is difficult to obtain the attribute characteristics of users for the reason of user privacy protection. At the same time, the label of POI is also simple, and users' evaluation of POI is often missing, so the data in POI recommendation are sparse. A time factor is also very important in POI recommendation. The location of users changes over time, and the interests of users also vary in different time periods [23]. These features are obviously different from the traditional recommendation system and make POI recommendation complicated and difficult.

According to the advantage that the heterogeneous graph convolution network can restore the original data structure in POI recommendation, a POI recommendation system based on the heterogeneous graph convolution network is proposed, which can effectively preserve the graph structure information in user data. The structure information between users and locations is obtained by heterogeneous graph convolution. The model proposed in this paper is shown in Figure 2.

The focus of the model is to extract the hidden layer information of users and hidden layer information of interest points from the user-interest point heterogeneous graph structure. The model predicts the interest location of user through this information and recommends it to users. The most important aspect of the algorithm is to obtain high-quality user-implicit layer information and interest point implicit layer information. In the proposed model, the complex user-interest point heterogeneous graph structure is first divided into the user-user social graph layer, user-interest point check-in graph layer, user-interest point evaluation graph layer, interest point-evaluation graph layer, and interest point-user check-in graph layer based on different focus points. Except for the user-user social graph layer, which is a homogeneous graph structure, all other graph layers are heterogeneous graph networks [24, 25].

The purpose of performing this decomposition is that the relationships between users and points of interest in the real world are very complex and the amount of data is huge, and most of the nodes in this complex mesh are heterogeneous from each other. This decomposition can better model relationships between data and consequently obtain richer hidden layer information [26].

First, for each of the three sublayers, the aggregation of neighbors is performed using graph convolution to obtain the embedding representation of the central node. Among them, in the user-user social layer, the neighbor nodes of the user are also users and edges are represented as social relationships between users and users, such as friend relationships. Then, the homogeneous graph convolution method is used to aggregate the first-order neighbors, second-order neighbors, and third-order neighbors of the user in turn, and finally, the vector of embedding expressions of user social attributes under the user-user social homogeneous graph is generated. For the layer of user-interest point check-in and the layer of user-interest point evaluation, because the information between user and check-in and user and evaluation is different categories of information, both layers are heterogeneous graph structures. Using the method of heterogeneous graph convolution, the neighbor nodes of the central node are aggregated, and finally, the embedding expressions of the central node are obtained as the user check-in attribute embedding expression subvector and the user evaluation attribute embedding expression subvector, respectively. The three subvectors are concatenated to obtain the overall embedding expression structure of the user. In the same way, for the interest point-evaluation layer and interest point-user check-in layer, heterogeneous graph convolution is used to obtain the subvectors under the interest point evaluation attribute and the interest point-user check-in attribute, and then, the embedding expression vectors of interest points are obtained by the splicing operation. Finally, the obtained user embedding vector and interest point embedding vector are used for prediction to get the interest points that users may be interested in for recommendation.

3.1. Information Extraction for Users. In point-of-interest recommendation, there is an association relationship between users and users, which can build the user network for users based on social relationships of location. Users will have behaviors such as check-in and evaluation on points of interest. Points of interest are refined expressions that are outlined based on the semantics and functions of locations. Besides, the important information that the point of interest should include is the address location of the point of interest, which can usually be expressed in terms of latitude and longitude [27]. In point-of-interest recommendation, multilayer data models can be constructed, such as user social layer, geographic information layer, and evaluation data layer, but the core of all layers of data combined is the user. Therefore, in point-of-interest recommendation, the user is the central point of the data graph structure of each layer.

3.1.1. User Social Layer. In many location-oriented social systems, users are associated with each other through social relationships, which are usually friends or groups. According to the reasoning of "birds of a feather fly together," a user's friends can characterize the current user [28]. For example, a user who likes food tagging locations may also have friends who like to clock in on food locations. The hidden layer information of the current user can be obtained by computing the graph convolutional aggregation of friend users who are associated with the current user nodes in the user's social relationship graph. Although the graph convolutional neural network can stack multiple convolutional layers to obtain information of more distant users, according to social theory, users who are too far away are actually not similar to the current user, so the model in this paper only needs to aggregate information of neighbors within 3 hops.

The model first constructs the social relationship matrix between users and users and extracts the hidden layer information of users based on the graph neural network.

One layer of graph convolution can obtain the node information of the first-order domain of the current central node, and by multilayer graph convolution, it can obtain information such as second-order neighbors and third-order neighbors of the user. According to the theory of the third degree of influence and the property that stacking too many layers of the graph convolutional network will cause oversmoothing, stacking up to 3 layers is sufficient. The calculation formula is as follows:

$$e_{\mathrm{us}}^{l} = \mathrm{LeakyReLUAGG}\left(\frac{1}{\sqrt{\left|N_u\right|}}\sum_{n\in N_u} W_1^l e_n^{l-1}\right), \tag{1}$$

where N_u denotes all first-order neighbors of the user u in the social graph layer, e_{us}^{0} is the original input of the user social graph layer, W_1^l is the weight parameter of the convolution of the layer l in the user social graph layer, and e_n^{l-1} is the neighbor nodes of the convolution of user u in the layer of $l-1$ in the user social graph layer.

3.1.2. Layers of User's Access. User check-ins to points of interest are heterogeneous information networks, and at the same time, user check-ins to points of interest may be multiple, sequential check-ins; therefore, the user-interest access layer is a multiorder network structure [29].

We need to model the check-in behavior of users in a certain time period because check-in visits may be continuous, so we can capture continuous visit records recorded by the system in a certain time period for multiorder network modelling, or we can calculate the top-k of user check-in frequency for modelling, top-1 is transformed into direct neighbors, top-2 is transformed into second-order neighbors, and so on. However, this implicit conversion method removes the direct information of user check-in and performs artificial transformation and extraction, which changes the content information and graph structure information of the original graph structure and may therefore affect the accuracy of the recommendation results, so this model uses the continuous check-in structure within a certain time period as the original input of the graph

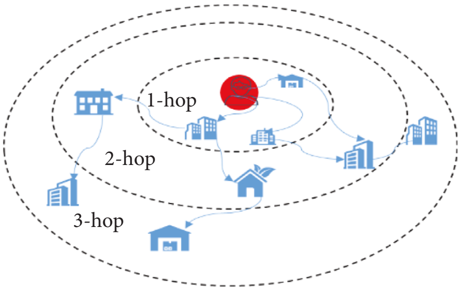


Figure 3: The 3-hop structure.

structure. Considering that the content after multihop information decays layer by layer for portraying the information value of the current user, this paper only extracts the access points within the 3-hop structure [29], which is illustrated in Figure 3.

For the user embedding expression of the user-interest point check-in graph layer, the calculation formula is as follows:

$$e_{\mathrm{up}}^{l}=\mathrm{LeakyReLU}\left(W_1^l e_{\mathrm{up}}^{l-1}+\frac{1}{\sqrt{|N_i|}}\sum_{i\in N_{\mathrm{up}}}\left(W_1^l e_i^{l-1}+W_2^l\left(e_i^{l-1}\odot e_{\mathrm{up}}^{l-1}\right)\right)\right), \tag{2}$$

where e_{up}^{l} is the original input of the user-interest point check-in layer and e_i is the first-order neighbor node of the currently calculated user node, which is the 1-hop check-in location of the current user in the user-interest point check-in layer. By stacking multiple convolutional layers, the information of the 2-hop check-in location, the 3-hop check-in location, and the more distant check-in locations of the user can be aggregated. Similarly, considering the information attenuation of nodes being too far away for inscribing the current node, this model also adopts only a three-layer heterogeneous graph convolutional stacking. W_1^l and W_2^l are the weight parameters of the lth layer convolution in the user check-in layer, and e_i^{l-1} is the neighbor nodes of interest points that the user has checked in in the layer of $l-1$ convolution in the user check-in layer.

3.1.3. User Evaluation Subgraph Layers. Similarly, there is a heterogeneous graph network between users and evaluations. Since there is no multiorder connection between users and evaluations, only the convolution of the first-order graph structure of user-evaluation needs to be considered. The specific calculation formula is as follows:

$$e_{\mathrm{ut}}^{l}=\mathrm{LeakyReLU}\left(W_1^l e_{\mathrm{ut}}^{l-1}+\frac{1}{\sqrt{|N_i|}}\sum_{i\in N_{\mathrm{ut}}}\left(W_1^l e_i^{l-1}+W_2^l\left(e_i^{l-1}\odot e_{\mathrm{ut}}^{l-1}\right)\right)\right), \tag{3}$$

where e_{ut}^{0} is the original input of the user evaluation graph layer, e_i is the first-order neighbor nodes of the currently calculated user node, which contains all evaluation information of the current user in the user evaluation graph layer. W_1^l and W_2^l are the weight parameters of the convolution of the layer l in the user evaluation graph layer, and e_i^{l-1} is the interest point neighbor node that the user has evaluated in the layer$l-1$ of convolution. The three user embedding expressions obtained by convolution for the above three layers are fused to obtain the user embedding vector under the user-interest point-evaluation heterogeneous graph structure. The formula is as follows:

$$e_u^l = e_{\mathrm{us}}^l \| e_{\mathrm{up}}^l \| e_{\mathrm{ut}}^l, \tag{4}$$

where $\|$ indicates the splicing operation of vectors.

3.2. Information Extraction of Interest Points. Considering interest points as central nodes, neighboring nodes of interest points also contain two kinds of information, one is the evaluation information against interest points and the other is the information of users who have ever checked in on interest points [30]. There should also be an association relationship between interest points, and this association is mainly a distance relationship in location. Because the information of the interest point itself contains location coordinates, the distance relationship between interest points is not modelled separately in this model. For the information extraction of interest points, it mainly comes from two parts, the interest point evaluation layer and the interest point-user layer, and the heterogeneous graph convolution method is used for the information extraction of the two layers, respectively.

3.2.1. Interest Point Evaluation Subgraph Layer. This layer is used to extract the implicit features of the interest point, and the labels and evaluations of the interest point can be used to find out what kind of interest point it is [31, 32]. For example, the interest point "Bing-Sheng Restaurant" is labeled as a restaurant, and evaluation information includes positive and negative information. This is a one-hop network structure, where all evaluation information is normalized to a number between [0, 1], where 1 means like and 0 means dislike, and the larger the number, the higher the positive evaluation. Because the interest points and evaluation properties are not the same, this is also a heterogeneous network, and the information extraction of interest points is performed by using heterogeneous graph convolution, which is calculated as follows:

$$e_{\mathrm{pt}}^{l}=\mathrm{LeakyReLU}\left(W_1^l e_{\mathrm{pt}}^{l-1}+\frac{1}{\sqrt{|N_i|}}\sum_{i\in N_{\mathrm{pt}}}\left(W_1^l e_i^{l-1}+W_2^l\left(e_i^{l-1}\odot e_{\mathrm{pt}}^{l-1}\right)\right)\right), \tag{5}$$

where e_{pt}^{0} is the original input of the interest point evaluation layer and e_i is the first-order neighbor node of the currently calculated interest point node, which contains all evaluation information of the current position in the interest point evaluation layer. W_1^l and W_2^l are the weight parameters of the lth layer convolution in the interest point evaluation layer, and e_i^{l-1} is the neighbor node of the evaluation information of the layer $l-1$ convolution of the interest point.

3.2.2. POI-User Subgraph Layer. The check-in data on points of interest originate from users, and graph structures can be constructed for points of interest as well as users who have checked in on points of interest. Similarly, because of different nature between interest points and users, this is also a heterogeneous graph network. Heterogeneous graph convolution is used to extract information from interest points, and the calculation formula is as follows:

$$e_{\mathrm{pu}}^{l} = \mathrm{LeakyReLU}\left(W_1^l e_{\mathrm{pu}}^{l-1} + \frac{1}{\sqrt{|N_i|}} \sum_{i \in N_{\mathrm{pu}}} \left(W_1^l e_i^{l-1} + W_2^l \left(e_i^{l-1} \odot e_{\mathrm{pu}}^{l-1} \right) \right) \right), \tag{6}$$

where e_{pu}^{0} is the original input of the point-of-interest user layer and e_i is the first-order neighbor node of the currently calculated point-of-interest node, that is, all signed-in users at the current position in the point-of-interest user layer. W_1^l and W_2^l are the weight parameters of the lth layer convolution in the point-of-interest user layer and e_i^{l-1} is the signed-in user neighbors in the layer $l-1$ convolution of the point-of-interest nodes.

The two interest point embedding vectors obtained from the heterogeneous graph convolution for the above two layers are fused to obtain the information of the interest points under the user-interest point-evaluation graph structure, which is calculated as follows:

$$e_p^l = e_{\mathrm{pu}}^l \| e_{\mathrm{pt}}^l, \tag{7}$$

where $\|$ indicates splicing.

3.3. Interest Point Prediction and Recommendation. The model finally uses e_p^l and e_u^l obtained in the above graph convolution to predict the rating of the user u for the interest point p. The calculation formula is as follows:

$$\hat{y}_{\mathrm{up}} = (E_u)^T E_p, \tag{8}$$

where E_u is the splicing of the embedding expression vector of each graph layer user obtained in the above step and E_p is the splicing of the embedding expression vector of each graph layer interest point obtained in the above step. The user's rating for the interest point is sorted in descending order, and the top K interest points are recommended to the user.

The loss function of the model uses the BPR loss [33], whose formula is defined as follows:

$$\mathrm{Loss} = \sum_{(u,i,j) \in D_S} \ln\, \sigma\left(\hat{y}_{\mathrm{ui}} - \hat{y}_{\mathrm{uj}} \right) - \lambda_{\Theta} \|\Theta\|^2, \tag{9}$$

where D_S is defined as follows:

$$D_S = \{(u,i,j) | (u,i) \in \mathscr{R}^+ \wedge (u,j) \in \mathscr{R}^- \}, \tag{10}$$

where $\mathscr{R}^+$ denotes the data where the user u rates the interest point i higher than the interest point j, $\mathscr{R}^-$ denotes the data where the user rates the interest point i lower than the interest point j. $\hat{y}_u$ denotes the user's prediction score for the interest point, λ_{Θ} denotes the parameter that controls the strength of $L2$ regularization to prevent overfitting, and Θ denotes all trainable parameters in the model.

By using the above approach, the features of users and points of interest under different layers are obtained by using heterogeneous graph convolution for different layers, and then, the embedding expressions of users and points of interest in the global graph structure are obtained; finally, the obtained final embedding expressions are used for prediction and recommendation of points of interest that users may be interested in.

4. Experiments

4.1. Baseline. The following algorithm is selected as the comparison baseline in the experiments.

(i) *UCF* [34]: This research attempts to facilitate a POI recommendation service in location-based social networks. Its idea is to incorporate user preference, social influence, and geographical influence into recommendation, and the research proposes a unified POI recommendation framework, which fuses user preference to a POI with social influence and geographical influence.

(ii) *BPR* [33]: This paper provides a generic learning algorithm for optimizing models with respect to BPR-Opt. The learning method is based on stochastic gradient descent with bootstrap sampling.

(iii) *GEOM* [35]: This model first proposes to exploit weighted matrix factorization for recommender tasks since it usually serves collaborative filtering with implicit feedback better, and the model augments users' and POIs' latent factors in the factorization model with activity area vectors of users and influence area vectors of POIs, respectively.

(iv) *GEOIE* [36]: This paper exploits POI-specific geographical influence to improve POI recommendation. It models the geographical influence between two POIs using three factors: the geoinfluence of POI, the geosusceptibility of POI, and their physical distance. Geoinfluence captures POI's capacity at exerting geographical influence to other POIs, and geosusceptibility reflects POI's propensity of being geographically influenced by other POIs.

4.2. Data Sets and Evaluation Metrics. The experimental comparison is based on two common public data sets, Foursquare and Gowalla.

Foursquare [37]: Founded in 2009, Foursquare has worked with worldwide collection and distribution of location data. The data set contains check-in data collected mostly from the USA and Tokyo. This data set also contains the list of all friends of each user in the LBSN.

Gowalla [38]: Gowalla is a location-based social media platform dedicated to location check-ins. Gowalla was primarily a mobile application that allowed users to

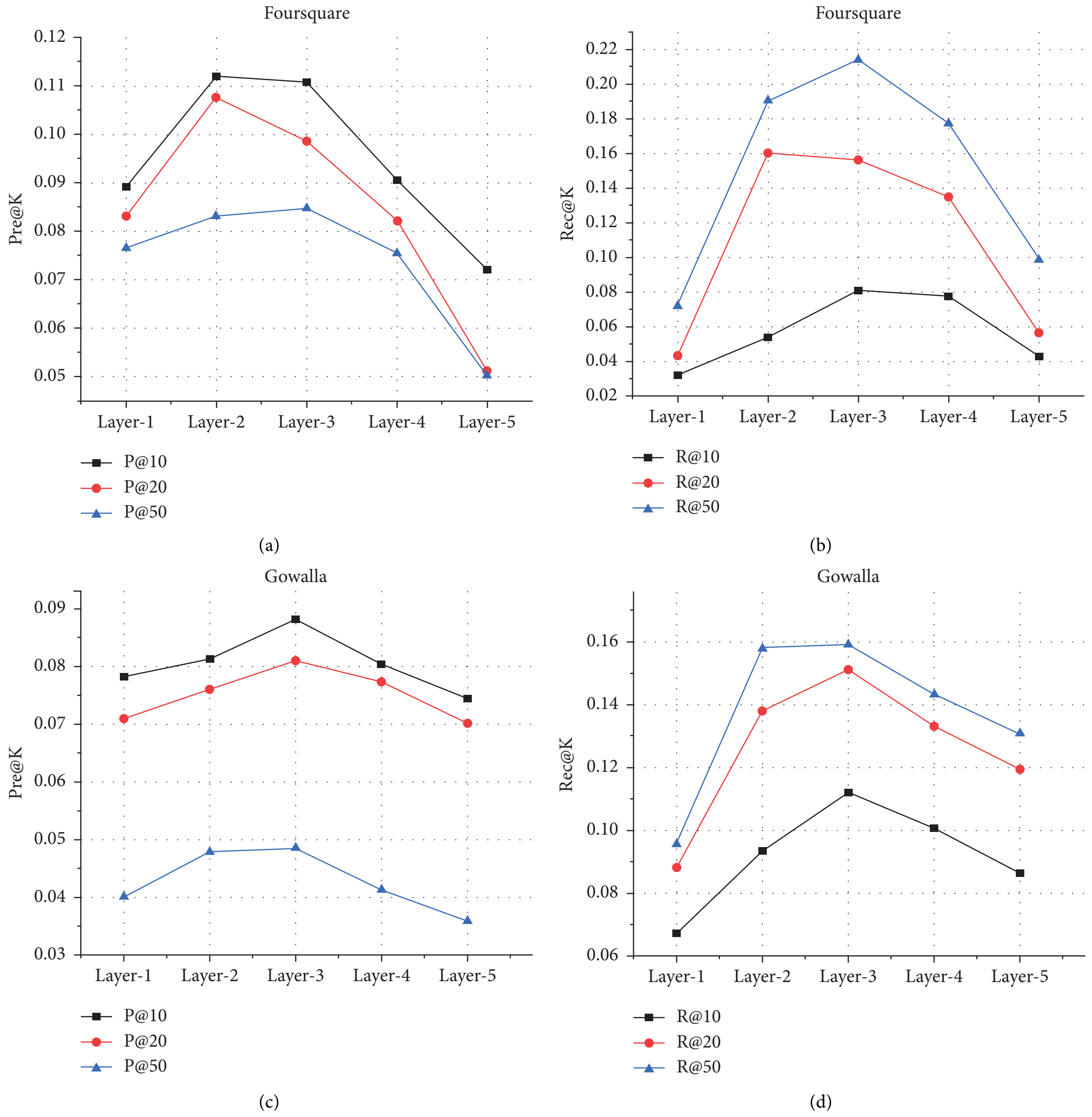


Figure 4: Effect of convolution layers. (a) Pre@K on Foursquare. (b) Rec@K on Foursquare. (c) Pre@K on Gowalla. (d) Rec@K on Gowalla.

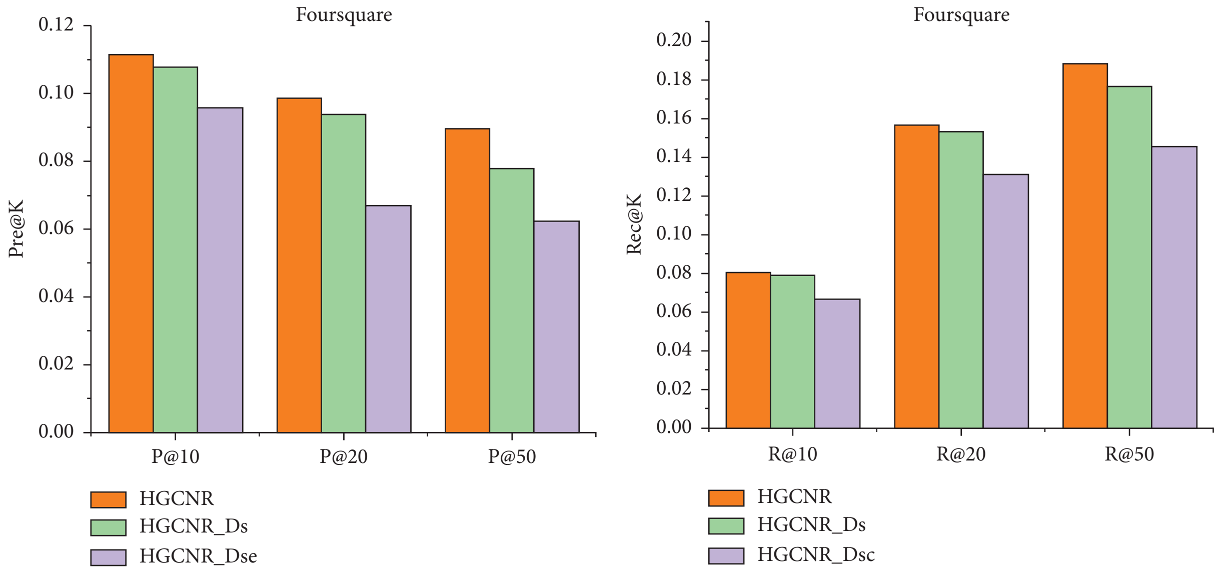
check into locations that they visited using their mobile devices. The data sets from the functioning period of Gowalla were available via the Gowalla API, and currently, there are no official distributors for the data sets. Like Foursquare, the Gowalla data set also contains the list of friends of every user in the data set. Besides, a detailed description of each POI and user profiles are also available in this data set.

In order to verify the recommended performance of the model proposed in this paper and to observe the optimal values of various parameter settings in the model of this paper, the precision rate Pre@K and the recall rate Rec@K are used as performance evaluation indexes in this paper.

4.2.1. Pre@K. The precision metric indicates the proportion of correct recommended POIs where users would visit within the top-K recommendations. The formula is as follows:

$$\text{Pre@}K = \frac{T_{\text{pu}}}{T_{\text{pu}} + F_{\text{pu}}}, \qquad (11)$$

where T_{pu} denotes the number of top-K recommendation lists generated by the algorithm that users really like and F_{pu} denotes the number of false recommendations generated by the top-K recommendation lists that users actually do not like.



(a)

(b)

Figure 5: Model comparison after layer removal. (a) Pre@K on Foursquare after layer removal. (b) Rec@K on Foursquare after layer removal.

4.2.2. Rec@K. The recall metric measures in what proportion the recommendation of K POIs covers ground truths of the POIs that users would visit. The formula is as follows:

$$\text{Rec@}K = \frac{T_{\text{pu}}}{T_{\text{pu}} + T_{\text{nu}}}, \tag{12}$$

where T_{nu} denotes the number of users who really liked but did not appear in the top-K recommendation list. We use the mean value of the precision rate and the mean value of the recall rate for all users to measure, and the formula is calculated as follows:

$$\begin{aligned} \text{Pre@}K &= \frac{1}{N}\sum_{u=1}^{N} \frac{T_{\text{pu}}}{T_{\text{pu}} + F_{\text{pu}}}, \\ \text{Rec@}K &= \frac{1}{N}\sum_{u=1}^{N} \frac{T_{\text{pu}}}{T_{\text{pu}} + T_{\text{nu}}}. \end{aligned} \tag{13}$$

4.3. Experimental Content

4.3.1. Effect of Convolution Layers. Existing studies have shown that one of the drawbacks of graph neural networks is that the models can be oversmoothed as the number of convolutional layers deepens [39, 40]. Because graph neural networks filter out high-frequency signals with large differences between nodes and retain low-frequency signals, deepening of the number of convolutional layers will make information differences between nodes become smaller and smaller and converge more and more and reduce the effectiveness of the model as a result [41]. Therefore, when designing recommendation models based on graph neural networks, it is necessary to choose an appropriate number of graph convolution layers. In this paper, the proposed model HGCNR on Foursquare and Gowalla data sets, the number of layers of graph convolution starts from 1 layer of graph convolution and increases layer by layer, and the observed performance metrics are shown in Figure 4.

As we can see from the experimental data, the Pre@k metric can get good results when performing one layer of graph convolution on two different data sets, probably because the subgraph layering design in this model can extract richer data information and thus improve the recommendation effect. As the number of layers of graph convolution deepens, the Pre@k indicator first shows an increasing trend and then decreases, and the curve trend shows that the recommendation effect does not improve with the increase in the number of layers of graph convolution but rather hurts the performance of the model. Similarly, the two data sets show the same results for the Rec@k metric; that is, the model performance increases and then decreases as the number of layers of graph convolution deepens. The difference is that, under the influence of different number of recommendations, the two metrics, Pre@k and Rec@k, either reach the optimal performance at 2-layer graph convolution or at 3-layer graph convolution. From the comprehensive experimental results, we believe that the overall model design is more reasonable for 3-layer graph convolution.

4.3.2. Effect of Layer Removal. In the user-centered graph neural network, the user social layer is removed to form the HGCNR_Ds model with a reduced layer, then, the user evaluation layer is removed, and only the user check-in layer is retained to form the HGCNR_Ds model, which is compared with the model with the full layer on the Foursquare data set. The results are shown in Figure 5.

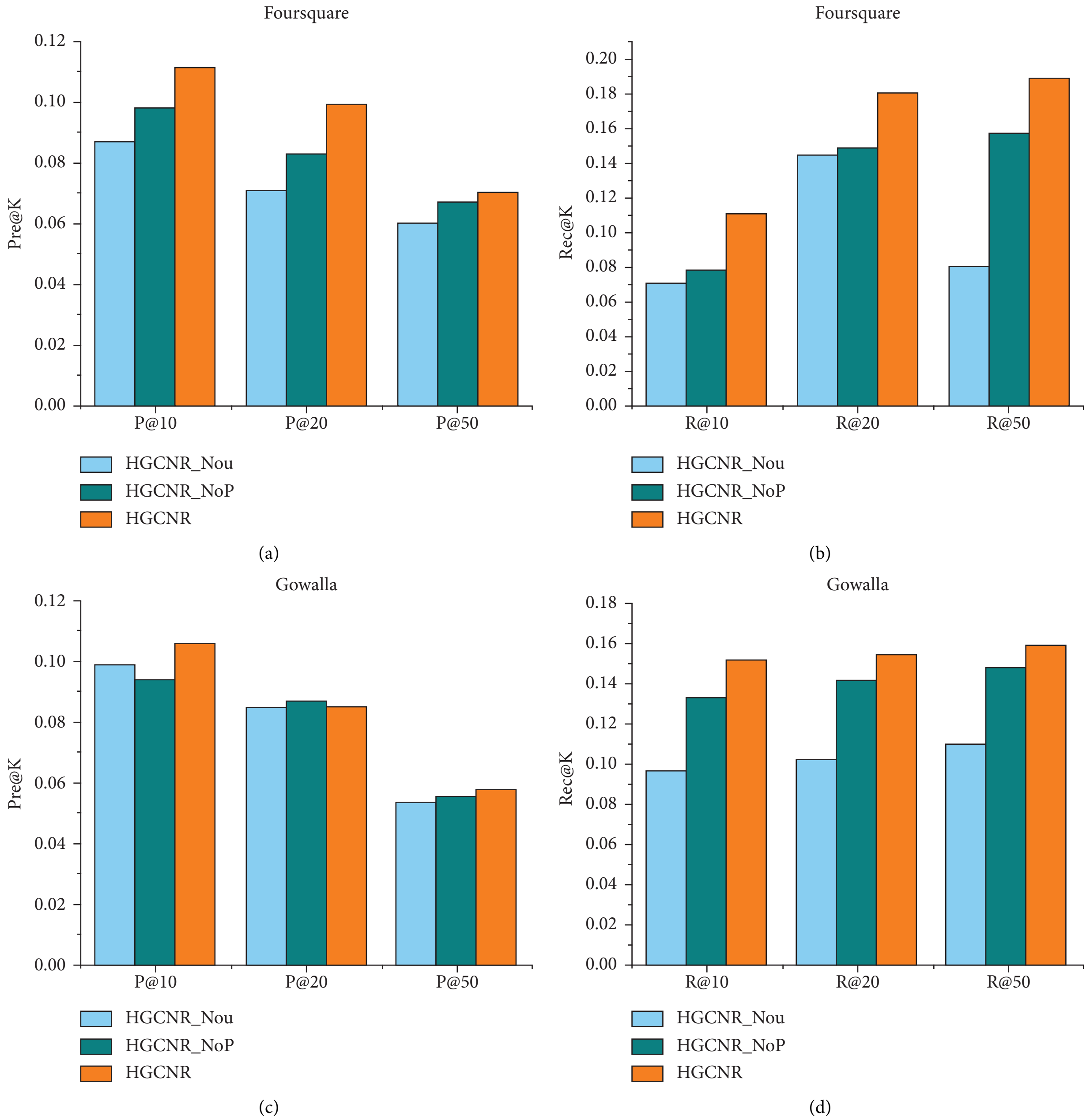


Figure 6: Comparison of the complete model and the simplified model. (a) Comparison on Foursquare with Pre@k. (b) Comparison on Foursquare with Rec@k. (c) Comparison on Gowalla with Pre@k. (d) Comparison on Gowalla with Rec@k.

From the comparison of the results, it can be seen that the performance of the reduced layer model with simply removing user social layers is slightly degraded in general compared to the performance of the full model, but the model with two sublayers removed undergoes larger degradation in performance compared to the full model. It shows that the sublayer decomposition approach can indeed extract useful information. Removing sublayers also removes a lot of useful node information and structural information, which is detrimental to the performance of the model.

4.3.3. Comparison of the Complete Model and the Simplified Model. To examine the effectiveness of the model proposed in this paper and to explore the effectiveness of the user subgraph layer and interest point subgraph layer design, the author simplifies the model in this paper into a user-centered graph structure without layered convolution called HGCNR_Nou, a simplified model with an interest point-centered graph structure without layered convolution called HGCNR_Nop, and the complete model proposed in this paper called HGCNR. The HGCNR_Nou model performs heterogeneous graph convolution directly with the user as the central node, while the HGCNR_Nop model performs heterogeneous graph convolution directly with the interest point as the central node. The number of recommendations in the experiment is set to 10, 20, and 50, respectively. The results obtained on the experimental data set are shown in Figure 6.

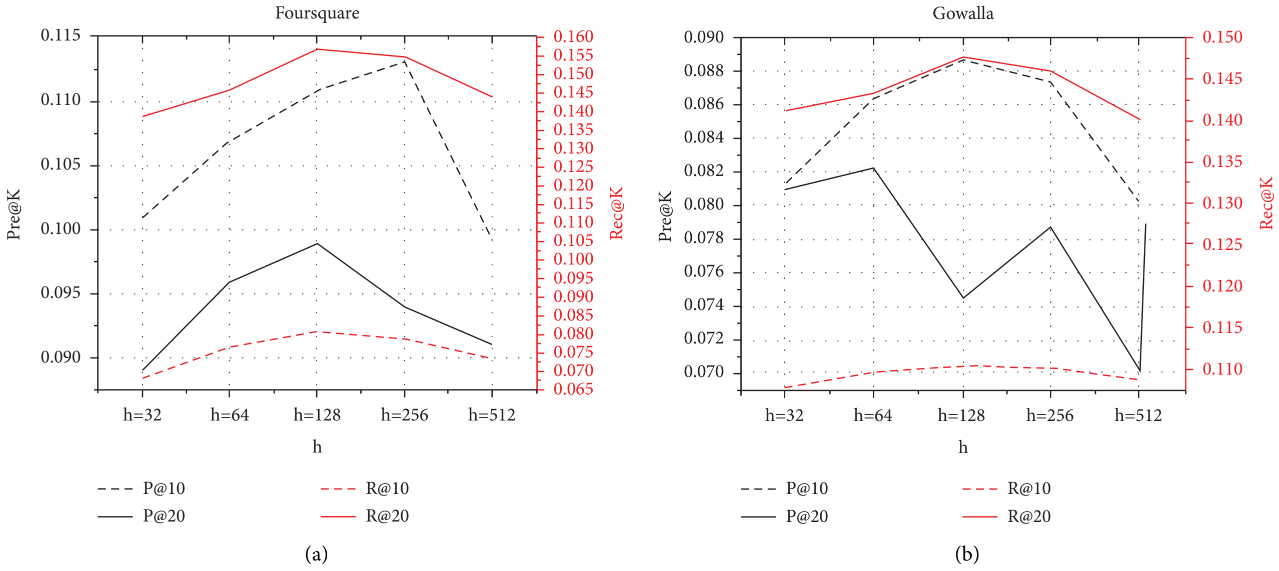


Figure 7: Influence of embedding dimension.

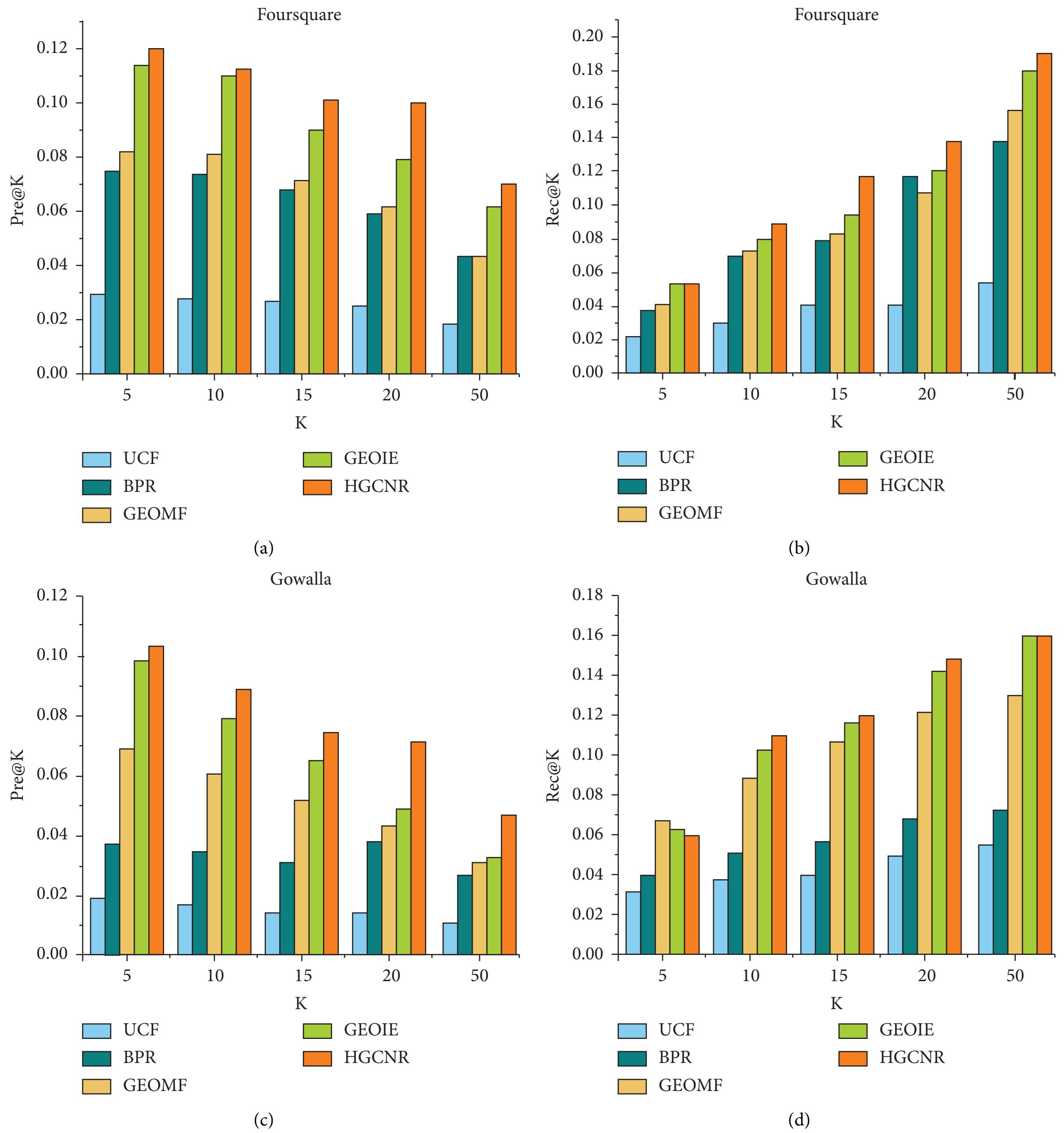
Through the experiments, we can see that the performance observations of the simplified models HGCNR_Nou and HGCNR_Nop on both data sets are significantly lower than those of the full model HGCNR proposed in this paper, indicating that the layered convolution of the user-centered graph structure and the layered convolution of the interest point-centered graph structure proposed in this paper are effective, and this layered graph convolution approach can obtain richer user information and interest point information content, thus improving the recommendation effect. Also, we should note that although the performance of both simplified models is lower than that of the full model, a comparison between the two simplified models shows that the recommendation performance of the HGCNR_Nop model is slightly higher than that of HGCNR_Nou. Analyzing the reason, it may be because the HGCNR_Nop model performs hierarchical graph convolution with three layers of subgraphs on the user-centered graph structure to obtain richer user information data, while the HGCNR_Nou model does not decompose the user graph structure but only performs subgraph decomposition on the interest point-centered graph structure. From this, we can speculate that, in interest point recommendation, the richness of user information affects the performance of recommendation results to a greater extent; that is, it is the characteristic of the user that determines the selection of interest points rather than the characteristic of interest points that determines the selection of the user, and this conclusion is also in line with common sense.

4.3.4. Influence of Embedding Dimension. The dimension h of the spatial vector also affects the ability of the embedding vector to represent the data features to a certain extent. Intuitively, the larger the spatial dimension is, the more data features can be represented and the greater the positive impact on recommendations [42, 43]. In our experiment, h is set to be 32, 64, 128, 256, and 512. Figure 7 shows the performance of HGCNR for the different values of h on the two data sets.

The results demonstrate that the performance in all evaluation metrics has similar behavior with the varying value of h. For the two data sets, the performance increases with the increase of h at the beginning and then, the best performance is achieved when $h = 128$ or $h = 256$ in Foursquare. For the Gowalla data set, the best performance is achieved at $h = 64$ or $h = 128$. However, we need to note that the recommendation performance of the model does not keep on improving with the increase of the spatial vector dimension. On both data sets, performance decreases when the embedding dimension increases to 512.

Such experimental results show that the representation capability of spatial vectors will increase with the increase of dimensionality, but the increase of spatial vector dimensionality does not improve recommendation performance infinitely. The spatial vector dimension being too large or too small will have a negative impact on the representation capability of spatial vectors. Too small spatial vector dimension will not be able to express more and richer data information, but too large spatial vector dimension will make the convergence of model training slower, and the training time of the model will increase. When the training method is set to a fixed number of iterations, if the vector dimension is too large, the structure of the graph nodes may not reach a stable state during the iteration process, and the trained spatial vector representation will be weaker. Overall, the parameter is fixed $h = 128$.

4.3.5. Comparison with Baseline. Based on the above experimental study results, the author obtained the final experimental comparison setup. For the comparison experiments with the comparison baseline, the following experimental environments and experimental setups are uniformly used:



(a)

(b)

(c)

(d)

Figure 8: Comparison results with the baseline.

GPU: Tesla V100 and video memory: 32 GB

CPU: 4 Cores, RAM: 32 GB, and disk: 100 GB

The data set is divided into a training set and a test set in the ratio of 8 : 2 for the experimental evaluation. The Adam algorithm was used to randomly optimize the parameters in the model. The learning rate is set to 0.001, and the dropout is set to 0.2. The maximum epoch is 200, and the early stop design is used.

The performance of the proposed model and the baseline model is compared in the experiments when the generated top-K is 5, 10, 15, 20, and 50, respectively. The comparison results with the baseline are shown in Figure 8. The experimental comparison shows that the model proposed in this paper achieves better results on both experimental data sets, indicating that the model proposed in this paper is effective in improving recommendation performance.

5. Conclusion

Point-of-interest recommendation is a kind of recommendation task, and by predicting locations that users may be interested in and making recommendations to them, it can not only improve the efficiency of users using the system but also enhance the user experience. To obtain better recommendation results, the author proposes a recommendation algorithm based on a heterogeneous graph convolution network, which uses the original graph structure information of users and points of interest directly and uses subgraph hierarchical graph convolution to obtain better user node information and point of interest node information from the user-centered user subgraph and point of interest-centered point of interest subgraph, respectively, and generates point of interest recommendation results by this method. In this paper, the author uses a simplified model and a complete model to generate the recommendation results. In this paper, the author demonstrates the effectiveness of this subgraph hierarchy on recommendation results by comparing the simplified model with the full model and the effectiveness of this model on recommendation performance improvement by comparing the baseline. In the next work, the author will further investigate the effect of different subgraph layer ablation on recommendation performance and whether adding the attention mechanism is effective for model performance improvement.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This study was supported by the Education Department of Guangdong Province (Education Teaching Reform Research and Practice Project, GDJG2021111).

References

[1] W. Zhou, D. Cao, Y. Xu, and B. Liu, “A survey of recommendation systems,” *Journal of Hebei University of Science and Technology*, vol. 41, no. 1, pp. 76–87, 2020.

[2] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, “A review on deep learning for recommender systems: challenges and remedies,” *Artificial Intelligence Review*, vol. 52, pp. 1–37, 2019.

[3] C. Zhou, J. Shen, Y. Li, and X. Guo, “Review of classical recommendation algorithms,” *Computer Science and Application*, vol. 09, no. 09, pp. 1803–1813, 2019.

[4] H. P. Ambulgekar, M. K. Pathak, and M. B. Kokare, “A survey on collaborative filtering: tasks, approaches and applications,” in *Proceedings of the International Ethical Hacking Conference*, Singapore, Asia, October 2018.

[5] M. Altulyan, L. Yao, X. Wang, C. Huang, S. S. Kanhere, and Q. Z. Sheng, “Recommender Systems for the Internet of Things: A Survey,” 2020, https://arxiv.org/abs/2007.06758.

[6] L. Chen, Y. Liu, X. He, L. Gao, and Z. Zheng, “Matching user with item set: collaborative bundle recommendation with deep attention network,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 2095–2101, Macao, China, August 2019.

[7] K. Yang, J. Zhu, and X. Guo, “POI neural-rec model via graph embedding representation,” *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 208–218, 2021.

[8] W. Heitor, S. Nícollas, C. V. Matheus, M. Fernando, C. M. P. Adriano, and R. Leonardo, “A survey on point-of-interest recommendation in location-based social networks,” in *Proceedings of the Brazilian Symposium on Multimedia and the Web*, pp. 185–192, SãoLuís, Brazil, November 2020.

[9] M. A. Islam, M. M. Mohammad, S. S. S. Das, and M. E. Ali, “A Survey on Deep Learning Based Point-Of-Interest (POI) Recommendations,” 2020, https://arxiv.org/abs/2011.10187.

[10] D. Lian, Y. Wu, Y. Ge, X. Xie, and E. Chen, “Geography-Aware sequential location recommendation,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Virtual Event, CA, USA, 2019 August.

[11] W. R. Tobler, “A computer movie simulating urban growth in the Detroit region,” *Economic Geography*, vol. 46, no. 2, pp. 234–240, 1970.

[12] G. Zhao, P. Lou, X. Qian, and X. Hou, “Personalized location recommendation by fusing sentimental and spatial context,” *Knowledge-Based Systems*, vol. 196, Article ID 105849, 2020.

[13] L. Hu, A. Sun, and Y. Liu, “Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction,” in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information*, Gold Coast Queensland, Australia, July 2014.

[14] Y. Wu, K. Li, G. Zhao, and X. Qian, “Personalized long-and short-term preference learning for next POI recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1944–1957, 2022.

[15] X. Ren, M. Song, and J. Song, “Point-of-Interest recommendation based on the user check-in behavior,” *Chinese Journal of Computers*, vol. 40, no. 1, pp. 28–51, 2017.

[16] S. Wu, F. Sun, W. Zhang, and B. Cui, *Graph Neural Networks in Recommender Systems*, A Survey, 2020, https://arxiv.org/abs/2011.02260.

[17] S. Wang, L. Hu, and Y. Wang, *Graph Learning Approaches to Recommender Systems: A Review*, https://arxiv.org/abs/2004.11718, 2020.

[18] W. L. Hamilton, “Graph representation learning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159, 2020.

[19] Y. Ma, S. Wang, C. C. Aggarwal, and J. Tang, “Graph convolutional networks with EigenPooling,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage AK, USA, July, 2019.

[20] L. Chen, J. Li, and J. Peng, *A Survey of Adversarial Learning on Graph*, https://arxiv.org/abs/2003.05730, 2020.

[21] X. Wang, H. Jin, and A. Zhang, “Disentangled graph collaborative filtering,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information*, China, July, 2020.

[22] Y. Fu, J. Wan, H. Zhao, W. Jiang, and S. Pu, “Preference-aware heterogeneous graph neural networks for recommendation,” in *Proceedings of the IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, Baltimore, MD, USA, November, 2020.

[23] X. Wang, H. Ji, and C. Shi, “Heterogeneous graph attention network,” in *Proceedings of the World Wide Web Conference*, pp. 202–2032, San Francisco CA, USA, May, 2019.

[24] J. Zhao, X. Wang, C. Shi, Z. Liu, and Y. Ye, “Network schema preserving heterogeneous information network embedding,” in *Proceedings of the 29th International Joint Conference on*

Artificial Intelligence (IJCAI), pp. 1366–1372, Yokohama, Japan, January, 2021.

[25] C. Shi, Y. Li, J. Zhang, Y. Sun, and P. S. Yu, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2017.

[26] H. Tang, G. Zhao, X. Bu, and X. Qian, "Dynamic evolution of multi-graph based collaborative filtering for recommendation systems," *Knowledge-Based Systems*, vol. 228, Article ID 107251, 2021.

[27] S. Zhao, T. Zhao, I. King, and M. R. Lyu, "Geo-teaser: geo-temporal sequential embedding rank for point-of-interest recommendation," in *Proceedings of the 26th International Conference on World Wide Web Companion*, Perth, Australia, April, 2017.

[28] S. Liu, "User modeling for point-of-interest recommendations in location-based social networks: the state of the art," *Mobile Information Systems*, vol. 2018, Article ID 7807461, 13 pages, 2018.

[29] B. Jin, C. Gao, X. He, D. Jin, and Y. Li, "Multi-behavior recommendation with graph convolutional networks," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*, China, July, 2020.

[30] G. Christoforidis, P. Kefalas, A. N. Papadopoulos, and Y. Manolopoulos, "RELINE: point-of-interest recommendations using multiple network embeddings," *Knowledge and Information Systems*, vol. 63, no. 4, pp. 791–817, 2021.

[31] S. Xu, X. Fu, J. Cao, B. Liu, and Z. Wang, "Survey on user location prediction based on geo-social networking data," *World Wide Web*, vol. 23, no. 3, pp. 1621–1664, 2020.

[32] M. Xie, H. Yin, H. Wang, and L. Yunyao, "Learning graph-based POI embedding for location-based recommendation," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis, Indiana, USA, October, 2016.

[33] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, Montreal Quebec, Canada, June, 2009.

[34] M. Ye, P. Yin, W. C. Lee, and D. L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information*, Beijing, China, July, 2011.

[35] D. Lian, C. Zhao, X. Xie, and L. Dima, "GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, August, 2014.

[36] H. Wang, H. Shen, W. Ouyang, and X. Cheng, "Exploiting poi-specific geographical influence for point-of-interest recommendation," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, July, 2018.

[37] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "An empirical study of geographic user activity patterns in Foursquare," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, Barcelona, Catalonia, Spain, July, 2011.

[38] R. Baral and T. Li, "MAPS: a multi aspect personalized POI recommender system," in *Proceedings of the 10th ACM Conference on Recommender Systems*, Boston, USA, September, 2016.

[39] B. Xu, K. Cen, J. Huang, H. Shen, and X. Cheng, "A survey on graph convolutional neural network," *Chinese Journal of Computers*, vol. 43, no. 5, pp. 755–780, 2020.

[40] R. Ying, R. He, K. Chen, P. Eksombatchai, and W. L. Hamilton, "Graph convolutional neural networks for Web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, United Kingdom, July, 2018.

[41] S. Zhu, C. Zhou, S. Pan, X. Zhu, and B. Wang, "Relation structure-aware heterogeneous graph neural network," in *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1534–1539, Beijing, China, November, 2019.

[42] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: a survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.

[43] R. Hussein, D. Yang, and P. Cudré-Mauroux, "Are meta-paths necessary? Revisiting heterogeneous graph embeddings," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 437–446, Torino, Italy, October, 2018.