

Research Article

Point-of-Interest Recommendation Model Based on Graph Convolutional Neural Network

Ziyang Wu  and Ning Xu

School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China

Correspondence should be addressed to Ziyang Wu; wu1292300145@whut.edu.cn

Received 31 May 2022; Revised 18 June 2022; Accepted 23 June 2022; Published 6 July 2022

Academic Editor: Kalidoss Rajakani

Copyright © 2022 Ziyang Wu and Ning Xu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of location-based social networks, the point-of-interest recommendation has become one of the research hotspots in the field of recommendation. However, traditional technologies like collaborative filtering are limited by the influence of data sparsity and cannot accurately capture the users' preferences from the complex context. In order to address this problem, a recommendation model based on graph convolutional neural network is proposed, named RMGCN. RMGCN is composed of three parts: graph structure features extraction module, geographical factor evaluation module, and score calculation module. The graph structure feature extraction module is used to extract node features from the graph structure data composed of user check-in records. The geographical factor evaluation module is used to calculate the influence coefficient of geographical factors on user's decision-making behaviors. The score calculation module is used to combine the output results of the above two modules and calculate the user's preference scores of point-of-interests combined with temporal context and spatial context. Experimental results on two real-world datasets show that RMGCN has better recommendation performance than baselines.

1. Introduction

In location-based social networks, users communicate with other users on the platform by sharing check-in information. The core content of check-in information is the location, which is called point-of-interest (POI) in location-based social networks. With the rapid development of location-based social networks, the scale of social platforms has been expanding. On the one hand, the large-scale platform attracts a large number of merchants, which increases the variety and scope of choices available to users, resulting in the problem of information overload—users cannot make efficient choice decisions in a short time; on the other hand, a large-scale platform can accommodate more active users, who generate a large amount of user behavior data during interaction with social platforms. Obviously, fully mining user behavior data can create great economic value for the platform. In order to improve the user experience and increase the economic benefits of the platform, POI [1] recommendation emerged at the right time and gradually became a hot research topic.

Compared with traditional recommendation tasks, such as music recommendation and film recommendation, the challenges faced by POI recommendation are more severe. This challenge mainly consists of two aspects: data sparsity and complex context. First, data sparsity: in the scenario of POI recommendation, the cost of generating an activity record is relatively high, resulting in fewer check-in records for most users, i.e., serious data sparsity problems will be faced during modeling; second, complex context: a check-in decision of a user is affected not only by the temporal context but also by the geographical context, which also means that the decision of a user is affected by multiple contexts, that is, a user's preference [2, 3] will vary with the context. How to accurately capture a user's preference in a complex context is another challenge in the field of POI recommendation.

To address the above-mentioned two challenges, recent works have been carried out from the perspective of integrated multifactor joint modeling. Zhang and Chow [4] integrated geographical factors, social relationships, and POI

classification into the same framework and combined various information to assess users' preferences for POIs. Xie et al. [5] reconstructed users' check-in records in the form of a bipartite graph and used the network embedding model to learn the characteristic representation of users and POIs and finally generated a recommendation list for users through the scoring calculation formula integrating time, geography, and semantics. Li et al. [6] proposed a high-order tensor decomposition algorithm based on time perception to capture the influence of time information, geographical location, and POI classification on user decision-making.

As can be seen from the above researches, integrating various context information is conducive to modeling users' preferences, and the graph-based approach can naturally and intuitively reconstruct user check-in behaviors in complex contexts [5]; however, most recent research works are shallow models, lacking the ability to deeply mine user characteristics from graph structure data.

For this purpose, a POI recommendation method based on Graph Convolutional Network (GCN) and multiple contexts is proposed to mine the characteristic representation of users and POIs from graph structure data. Firstly, two kinds of bipartite graphs are constructed according to the check-in information of users to capture the correlation between POIs and users and that between POIs and time factors, respectively; then, features of users, POIs, and various contexts are extracted by the improved GCN model; finally, a unified scoring formula is used to calculate the preference score of each POI for the target user and generate a recommendation list.

The contributions of this paper are summarized as follows:

- (1) We construct two novel graphs for capture user preferences in complex context
- (2) We propose an enhanced neighborhood aggregation function for precisely representing the user preferences
- (3) Extensive experiments show that our proposed model is superior, compared with representative baselines

2. Related Work

This chapter will review recent research works from two aspects: POI recommendation and GCN.

2.1. Researches on POI Recommendation. In recent years, the research works deeply mine the influence of various factors such as time factor, geographical factor, and POI classification on user behaviors. Zhang and Chow [7] divided users' check-in time into workdays and weekends to explore the temporal correlation of users' check-in behaviors and used the kernel density estimation method to model the influence of geographical factors on user preferences. Finally, a recommendation model TICRec that integrated time correlation and regional correlation was proposed. Liu et al. [8] deeply mined the influence of time factors on users' decision-making and proposed a double-weighted low-rank graph

construction model, which combines users' interest and their changing sequential preferences with time interval assessment to provide POI recommendations for specific time periods. Lian et al. [9] performed visualized analysis on the locations of users' check-in behaviors and found a spatial aggregation phenomenon, that is, individual visit locations tend to aggregate; in addition, they defined the user activity region and POI influence region to capture this phenomenon. Li et al. [10] introduced the idea of learning-to-rank into the field of POI recommendation, constructed a partial order relationship for POIs by using the check-in frequency of users, and proposed a ranking-based geographic factorization model Rank-GeoFM to mitigate the negative impact of data sparsity on the recommendation performance of the model. Feng et al. [11] proposed a recommendation model for joint modeling of user preferences and POI sequence transformation effects by integrating the influence of geographical factors based on the representation learning method. Yang et al. [12] integrated time factors and space factors and proposed a model UPOST for user preference of space-time topics. This model infers user preferences for different types of locations in different periods by learning space-time topics from users' historical semantic locations.

It can be seen from the above studies that time factors and geographical factors play an important role in modeling user preferences in the POI recommendation field. Therefore, this paper will comprehensively consider the influence of time factors and geographical factors and introduce them into the joint modeling of user preferences in the unified model to improve the recommendation performance of the model.

2.2. Researches on GCN. Graph Convolutional Network (GCN) was first proposed by Kipf and Welling [13] and used for semisupervised learning tasks. It aggregates features from neighbor nodes of target nodes through a message transfer mechanism to enhance the feature representation of target nodes. Although GCN has a powerful ability to extract node features from graph structures, it is still subject to graph size and aggregation mode. In order to solve the problem that GCN cannot be applied to large-scale graph data, Hamilton et al. [14] proposed the GraphSAGE model, changing the traditional aggregation mode in GCN to sampling aggregation and randomly sampling a certain number of nodes from its two-step neighbor nodes for feature aggregation according to the topological structure of the target node in the network. This reduced the computational size of the aggregation operation. Veličković et al. [15] introduced the attention mechanism from the perspective of aggregation mode and constructed the graph attention network (GAT). Different from the average aggregation method in traditional GCN, GAT learns the attention coefficient between nodes through a layer of feedforward neural network and applies it to aggregation operations to realize weighted feature aggregation. In addition, He et al. [16] pointed out that there is redundant feature mapping operation in traditional GCN from the perspective of efficiency of GCN and proposed a simplified version of GCN called LightGCN. In LightGCN, only the node aggregation operation is retained. The results show that LightGCN has higher training efficiency without losing model accuracy.

In order to consider the training efficiency and recommendation performance of the model, this paper uses LightGCN as a basic model to construct a graph neural network to extract node features.

3. Background Knowledge and Definition of Concepts

This chapter introduces the traditional GCN and LightGCN models and defines the core concepts relevant to the research work in this paper.

3.1. Basic GCN and LightGCN. GCN was inspired by convolutional neural networks (CNN). The core modules of CNN are the convolution layer and pooling layer. The convolution layer is used for feature extraction, and the pooling layer is used for feature compression. The combination of the two can demonstrate a powerful feature extraction capability in data based on Euclidean space, but cannot be used in non-Euclidean-space data such as graph data. Therefore, Kipf and Welling [13] redefined CNN's core operations, convolution, and pooling on the data of the graph structure.

For a given static undirected graph $G = (V, E)$, V represents the set of nodes in the graph; E represents the set of edges in the graph. The calculation method of the $h + 1$ feature representation vector $\vec{z}_{v_i}^{h+1}$ of the node $v_i \in V$ in GCN in the figure is shown in

$$\vec{z}_{v_i}^{h+1} = \sigma \left(W_{h+1} \sum_{v_j \in N(v_i) \cup v_i} \frac{\vec{z}_{v_j}^h}{\sqrt{|N(v_j)| |N(v_i)|}} \right), \quad (1)$$

where $N(v_i)$ represents the first-order neighbor node of the node v_i in the figure, and the denominator of the fraction in Formula (1) represents the degree normalization of the two nodes, which corresponds to the convolution operation in CNN, namely, the feature extraction operation. W_{h+1} represents the weighting matrix of GCN in $h + 1$, used for feature mapping of feature vectors. $\sigma(\cdot)$ represents a nonlinear activation function.

He et al. [16] analyzed the convolution operation of GCN from the perspective of efficiency and proposed a light graph convolution operation (LightGC) through theoretical and experimental verification. The calculation method of this operation is shown in

$$\vec{z}_{v_i}^{h+1} = \sum_{v_j \in N(v_i)} \frac{\vec{z}_{v_j}^h}{\sqrt{|N(v_j)| |N(v_i)|}}. \quad (2)$$

It can be seen from Formula (2) that the step of feature mapping in traditional GCN is omitted in LightGC, and only the operation of aggregating features from node neighbor sets is retained. It can be seen from the theoretical analysis of LightGCN [16] and experimental data that the omission of feature mapping does not affect the performance of GCN on recommendation tasks, and the overhead of the

model is reduced due to the omission of this step. Therefore, this paper will take LightGCN based on LightGC as the basic model to construct the POI recommendation method.

3.2. Definition of Concepts. This section defines the concepts of the graph structure involved in this article and gives a description of the POI recommendation problem.

In order to analyze the users' check-in behaviors, this paper converts users' check-in records into graph-structure data where nodes represent users, POIs, and timestamps. Further, we construct two undirected and weighted graphs: user-POI graph and time-POI graph. The two graphs are defined as follows:

Definition 1 (user-POI graph). If $U = \{u_1, \dots, u_m\}$ is used to represent user set and $P = \{p_1, \dots, p_n\}$ is used to represent POI set, m, n represents the number of users and the number of POIs, respectively. Then, the bipartite graph of user-POI can be expressed as $G_{UP} = (U, P, E_{up})$. The composition rule of graph G_{UP} is: if a user u signs in to POI p , there will be an edge e_{up} in graph G_{UP} , and the weight of this edge is the frequency of user u signing in to POI p , which is recorded as w_{up} .

The user-POI graph is used to describe the user's check-in behaviors and visually express the user's preference for the POI by taking the check-in frequency as the edge weight. Without loss of generality, the weights are normalized.

Definition 2 (time-POI graph). If $T = \{t_1, \dots, t_b\}$ is used to represent the time set and $P = \{p_1, \dots, p_n\}$ is used to represent the POI set, b, n represents the time and the number of POIs, respectively. Then, the bipartite graph of time-POI can be expressed as $G_{TP} = (T, P, E_{tp})$. The composition rule of graph G_{TP} is: if POI p is within a time t , there will be an edge e_{tp} in graph G_{TP} , and the weight of this edge is the frequency of POI p being signed in within the time t , which is recorded as w_{tp} . The weights of time-POI graph are also normalized.

The time-POI graph is used to describe the correlation between POIs and check-in time. According to the conclusions of the research works [5, 6], the check-in frequencies of the same POI are different at different times; the check-in frequencies of different POIs are different at the same time. This indicates that users' check-in behaviors are largely influenced by time factors. In this paper, the time is divided according to a model of 12 hours a day to capture the check-in rules of users at different times of the day.

Definition 3 (POI recommendation). According to the description of POI recommendation in researches [5, 10], POI recommendation is defined as follows: for a recommendation request $q = (u, l, t)$, u indicates the target user; l indicates the location of the user, and t indicates the time when the request is initiated. POI recommendation is to generate a list of recommended POIs $List = \{p_1, \dots, p_k\}$ (length: k) that the user has not visited.

Input User check-in set D , regularization coefficient λ , user-POI graph G_{UP} , time-POI graph G_{TP} , geographical threshold d_0

Output Model parameters Θ

1. Initialize model parameters Θ ;
2. Calculate the feature vector $\vec{u}, \vec{pu}, \vec{t}, \vec{pt}$ according to Equation (3);
3. Calculate the geographical factor coefficient gc according to Equation (4);
4. Calculate the user preference scores for the POI according to Equation (6);
5. Minimize Equation (7) and optimize model parameters with a stochastic gradient descent algorithm, repeating steps 2, 3, 4, and 5 before model convergence.

ALGORITHM 1: Model Training.

Input Model parameters Θ , Recommendation request $q = (u, l, t)$

Output Recommended list $List = \{p_1, \dots, p_k\}$

1. Calculate the user preference scores for the POI according to Equation (6);
2. Sort the POIs in descending order of preference score;
3. Take the first k POIs to form a recommendation list for the user.

ALGORITHM 2: Recommendation List Generation.

4. Recommendation Model Based on Graph Convolution Neural Network (RMGCN)

This chapter will detail the proposed model, including the overall framework, core modules, and model optimization.

4.1. Model Framework. In view of the above-mentioned statements, this paper proposes a Recommendation Model Based on Graph Convolution Neural Network (RMGCN), which consists of three core modules: a graph structure feature extraction module, geographical factor evaluation module, and score calculation module.

The graph structure feature extraction module is used to extract the feature representation of nodes from two types of bipartite graphs, a user-POI graph, and a time-POI graph. The geographical factor evaluation module is used to model the influence of geographical factors on user decision-making behaviors. Finally, the score calculation module is used to calculate user preference scores for each POI and generate a final recommendation list.

4.2. Graph Structure Feature Extraction Module. The graph structure feature extraction module is used to extract the features of nodes from two bipartite graphs. Specifically, the feature vector \vec{u}, \vec{pu} of a user-POI graph is extracted from the user-POI graph to describe the correlation between the user and the point-of-interest; the feature vector \vec{t}, \vec{pt} of time and POI is extracted from the time-POI graph to describe the direct correlation between time and POI.

It can be understood from the description in Section 3.1 that the LightGC operation is capable of efficiently completing node feature aggregation and extraction, though the valuable ability to combine edge weight in the graph is absent from LightGC. Edge weight attributes are extremely important in graphs for both a user-POI graph and a time-POI graph: for a user-POI graph, the weight of the edge indi-

cates the frequency at which the user accesses the POI. The greater the weight, the higher the frequency, which reflects user preference for the POI; likewise, for a time-POI graph, the weight of the edge indicates how often the POI is accessed during that time. The higher the weight, the more frequently it is accessed, which also reflects a significant probability of the POI's being accessed during the same time in the next cycle.

Therefore, in order to incorporate the important edge weight attribute into the aggregation operation, Equation (2) has been improved upon in this paper. Retention normalization is employed and a convolution operation of WD-LightGC with fusion degree and weight normalization is proposed for the purpose of integrating edge weight:

$$\vec{z}_{v_i}^{h+1} = \sum_{v_j \in N(v_i)} \frac{w_{v_i, v_j}}{\sum_{v_x \in N(v_i)} w_{v_i, v_x}} \frac{\vec{z}_{v_j}^h}{\sqrt{|N(v_j)| |N(v_i)|}}. \quad (3)$$

Equation (3) involves the calculation formula of WD-LightGC, which optimizes the process of node feature extraction by introducing edge weight coefficients and then improves the recommendation performance of the model. The corresponding node feature vectors $\vec{u}, \vec{pu}, \vec{t}, \vec{pt}$ can be extracted from the two bipartite graphs with the help of Equation (3).

4.3. Geographical Factor Evaluation Module. Researches [5, 6, 9] have shown geographical distance to be an important factor in user decision-making. An analysis of real-life situations has indicated that the geographical factor may exert less influence on users within an acceptable distance range. Beyond that range, however, users will abstain from visiting a location even if they would like to because of the effort required to reach it. For example, if the desired restaurant

TABLE 1: Statistics in the datasets.

Dataset name	Number of users	Number of POI	Density
Yelp_LV	6359	16435	0.13%
Foursqaure_TKY	2987	5235	0.26%

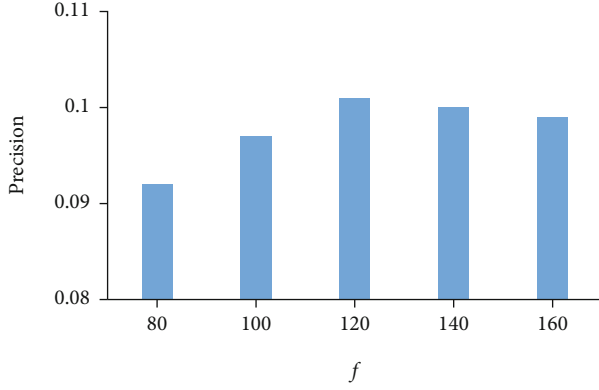


FIGURE 1: Precision on TKY dataset.

is 20 km away from the current location at lunchtime, then the user will most likely forego undertaking the journey.

In order to grasp the impact of geographical factors on user check-in behaviors, the geographical factor evaluation module has been incorporated in RMGCN to calculate the geographical factor coefficient of the current POI p , as shown in

$$gc(l, l_p) = \frac{1}{\text{dis}(l, l_p)}, \quad (4)$$

where $\text{dis}(l, l_p)$ represents the geographical distance coefficient between the POI p and the current location l , which is calculated as shown in

$$\text{dis}(l, l_p) = \frac{\max(\text{geo}(l, l_p), d_o)}{d_o}, \quad (5)$$

where $\text{geo}(l, l_p)$ represents a function that calculates the geographical distance between two points through latitude and longitude, both of which are expressed in latitude and longitude. d_o denotes the geographic distance threshold, which is the hyperparameter of the model, and $\max(\cdot)$ denotes taking the maximum value. As shown in Equation (4) and Equation (5), within the range d_o , the geographical factor exerts less influence on user choice, and gc is a constant; outside the range d_o , gc will decrease as the value of $\text{dis}(l, l_p)$ increases, which means that users are less likely to visit places that are too far away.

4.4. Score Calculation Module. Based on the node feature and geographical factor coefficients calculated in Sections 4.2 and 4.3, this paper has designed a score calculation module that incorporates temporal and spatial contexts. To be

more specific, the calculation method of the user u_i and POI p_j preference scores at time t_x and location l_o is shown in

$$s_{u_i, p_j} = \left(\vec{u}_i^T \cdot \vec{p}u_j + \vec{t}_x^T \cdot \vec{p}t_j \right) \cdot gc(l_o, l_p). \quad (6)$$

Through Equation (6), the preference scores of the target user for each POI at the current time and geographical location can be obtained to generate the recommendation list under the current time and location for the user according to the preference scores in descending order.

4.5. Model Optimization and Recommendation Process. For model optimization, this paper has adopted a framework based on Bayesian personalized ranking (BPR) [17] for model parameter learning.

The loss function based on BPR is defined as shown in

$$L = - \sum_{u=1}^m \sum_{p_i \in D_u} \sum_{p_j \notin D_u} \text{In} \tau(s_{u, p_i} - s_{u, p_j}) + \lambda \|\Theta\|^2, \quad (7)$$

where D_u represents the set of user u check-ins, $\tau(\cdot)$ represents the sigmoid function, λ represents the regularization coefficient, and Θ represents the model parameters. The training of the model can be accomplished by employing the stochastic gradient descent algorithm and minimizing Equation (7). The optimization process is shown in Algorithm 1.

After the trained model parameters Θ have been obtained, the recommendation list is generated for the user as shown in Algorithm 2.

5. Experimental Results and Analysis

This chapter focuses on the experimental setup and an analysis of the results. The experimental setup includes the datasets, baselines, and experimental settings; the result analysis contains reporting on the results of various implemented projects and provides a corresponding experimental analysis.

5.1. Datasets. The datasets used in this experiment are open-source datasets published on real social platforms: Yelp (<https://www.yelp.com/dataset>) and Foursquare (<https://sites.google.com/site/yangdingqi/home/>). The content of the datasets is the check-in records of users in two cities. Each check-in record consists of four parts: user ID, POI ID, check-in time, and check-in location. Specific statistics are shown in Table 1.

Four datasets represent the activity records of users in four cities, with Yelp_LV (hereinafter referred to as LV) representing Las Vegas and Foursqaure_TKY (hereinafter referred to as TKY) representing Tokyo. Density is employed to measure the sparsity of the datasets. The datasets are divided by sorting each user's check-in records by check-in time and taking the top 80% of check-in records as the training set and the remaining part as the test set.

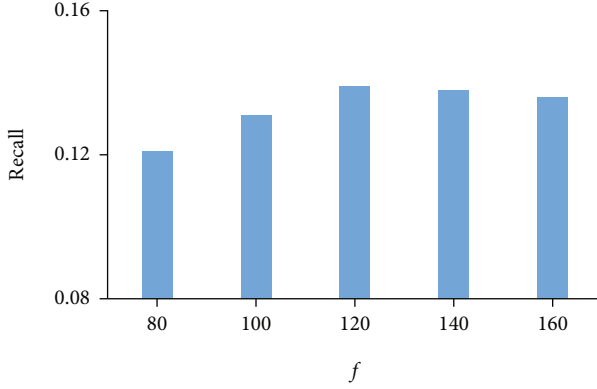


FIGURE 2: Recall on TKY dataset.

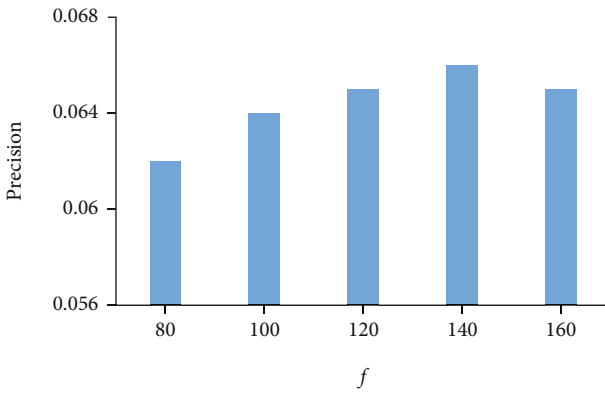


FIGURE 3: Precision on LV dataset.

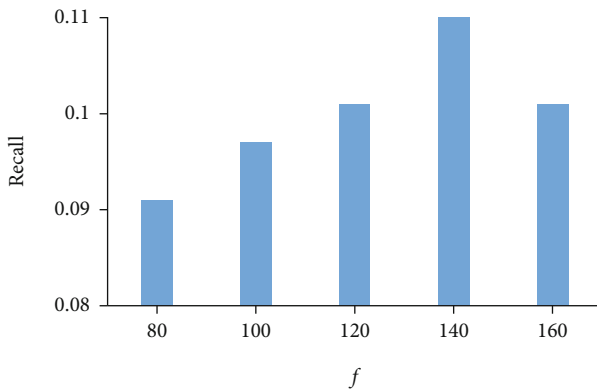


FIGURE 4: Recall on LV dataset.

5.2. Comparison Algorithm and Evaluation Metrics. For the experiment in this paper, the following POI recommendation model has been selected to compare the performance with the model RMGCN proposed in the text:

GE [5]: based on the POI recommendation model embedded in the graph, the sequence factor, time factor, and geographical factor have been integrated into a unified model in the form of a bipartite graph.

TAD-FPMC [6]: based on the POI recommendation model of high-order tensor decomposition, this model has divided the POI recommendation into two steps: the first step generates recommendations of POI classification according to user behavior; the second step recommends specific POIs for users according to POI classification.

PRME-G [18]: based on the POI recommendation model of embedded representation, the user preference features are learned by mapping user check-in behaviors to two potential spaces.

To evaluate the recommendation performance of the model, we adopt precision and recall as evaluation metrics. The evaluation indexes are calculated as shown in

$$\text{Precision} = \frac{|D_{\text{test}} \cap \text{Top}_k|}{|\text{Top}_k|}, \quad (8)$$

$$\text{Recall} = \frac{|D_{\text{test}} \cap \text{Top}_k|}{|D_{\text{test}}|}, \quad (9)$$

where D_{test} represents the test set and Top_k represents the recommendation list with a length of k .

For baselines, we adopt the recommendation parameter settings in their papers. For our proposed model, we set learning rate 0.05 and set λ 0.0005.

5.3. Parameter Sensitivity Experiment. There are two hyperparameters in the RMGCN model: the dimension f of the feature vector and the threshold d_o of geographical factor.

Since the feature mapping step is omitted in GCN, the dimension of the extracted node feature vector is the same as that of the node vector at initialization time, so the dimension value should be the hyperparameter of the model. In order to set an optimal group of parameter values, the performance of RMGCN is evaluated by setting different values on two datasets, the best of which is selected as the optimal values in the end.

For the experiment of the dimension f of feature vector, the value d_o of is fixed at 20 km, the recommendation list length is fixed at 15, and a series of values f are assigned to $f = (80, 100, 120, 140, 160)$. Finally, the experiment is carried out on two datasets, with the experiment results shown in Figures 1–4.

From Figures 1–4, it can be found that the performance of the model shows a tendency to increase initially and then decrease as the dimension of the feature vector increases. This is due to the fact that when the dimension is relatively small, the features that can be expressed have less meaning and cannot carry more information. Therefore, when the dimension increases, the performance of the model also tends to increase. When the dimension increases by a certain value, the model is set to a fixed number of iterations, which results in the high-dimensional feature vector's not being trained sufficiently with the limited number of iterations, resulting in performance degradation of the model. This also means that when larger dimensions are set, more iterations are needed during model training to ensure adequate training. For efficiency and performance, the dimension is set to 120 on TKY dataset and 140 on LV dataset.

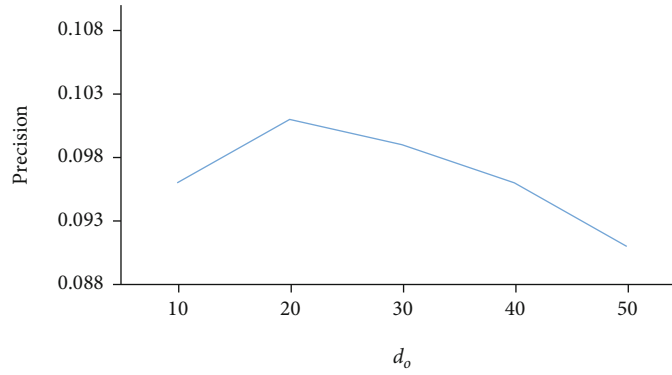


FIGURE 5: Precision on TKY dataset.

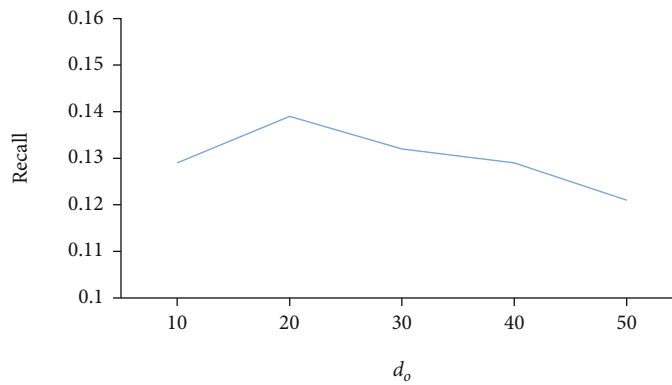


FIGURE 6: Recall on TKY dataset.

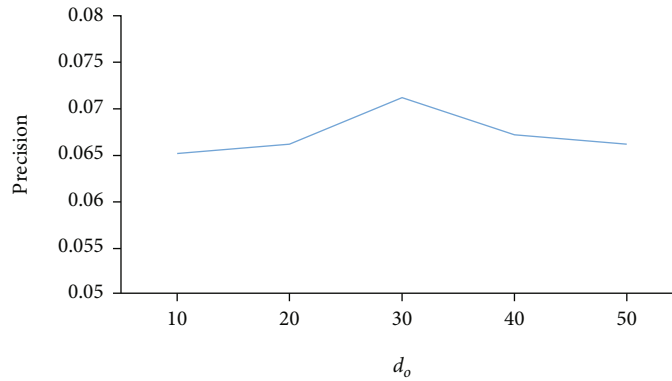


FIGURE 7: Precision on LV dataset.

The value of the geographical threshold is determined after the value of the feature vector dimension. The experimental method also entails fixing the feature dimension, adjusting the value of the geographical threshold, and observing the performance of the model. The geographical threshold takes a value in the range of $d_o = (10, 20, 30, 40, 50)$, in kilometers. The value range for setting the geographical threshold here is estimated on the basis of experience, assuming a vehicle speed of 50 km/h in an urban area, and an hourly distance as the maximum range that can be considered by the user. The experimental results are shown in Figures 5–8.

Figures 5–8 show the same trend, which is related to the equation for calculating the geographical threshold. The geographical threshold represents a tendency on the part of the user when making choices to disregard the distance from the current location to the target location within a certain geographical distance. Beyond this distance, the probability that the location will be selected by the user decreases as the distance increases. The experimental results in Figures 5–8 also confirm this hypothesis. It is worth noting that the two cities manifest different geographical thresholds, which may be related to the construction of the city.

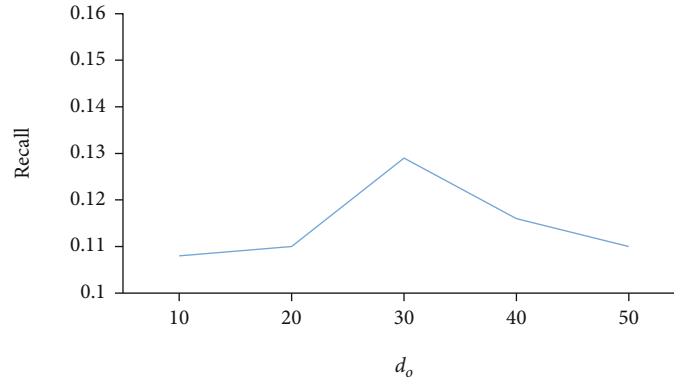


FIGURE 8: Recall on LV dataset.

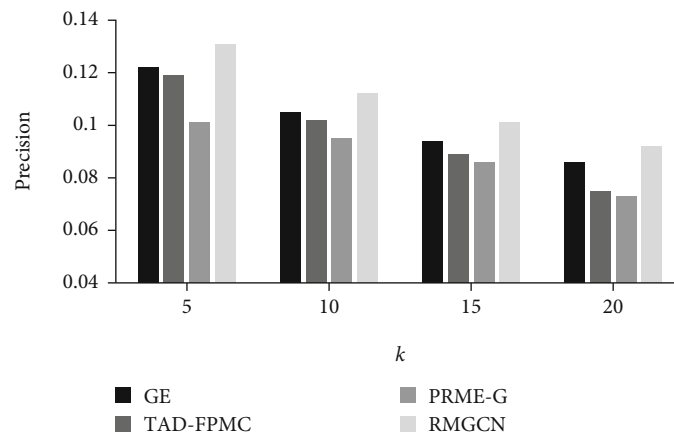


FIGURE 9: Precision on TKY dataset.

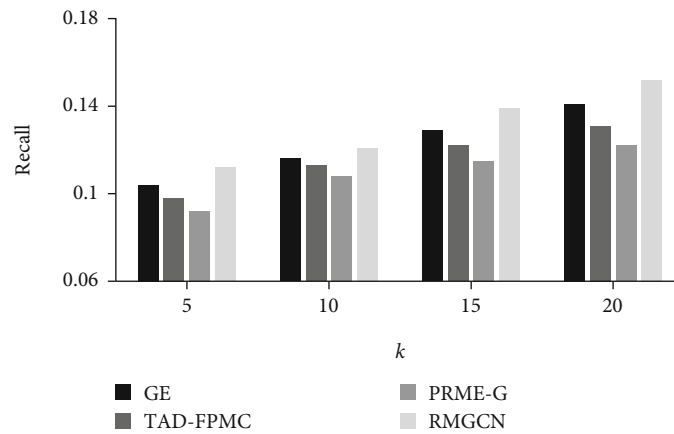


FIGURE 10: Recall on TKY dataset.

So far, the values of two important hyperparameters on two datasets have been determined through experiments: on the TKY dataset, the values of dimension and geographical threshold are 120 and 20km; on the LV dataset, the values of both are 140 and 30km. These two sets of values will be employed as optimal parameters to train RMGCN and for model comparison experiments.

5.4. Model Comparison Experiment. This section compares the recommendation performance of RMGCN with the other three baselines on two datasets. The recommendation list has a value range of $k = (5, 10, 15, 20)$. Experimental results are detailed in Figures 9–12.

Throughout the experimental results of the two datasets, the RMGCN model proposed in this paper has shown a

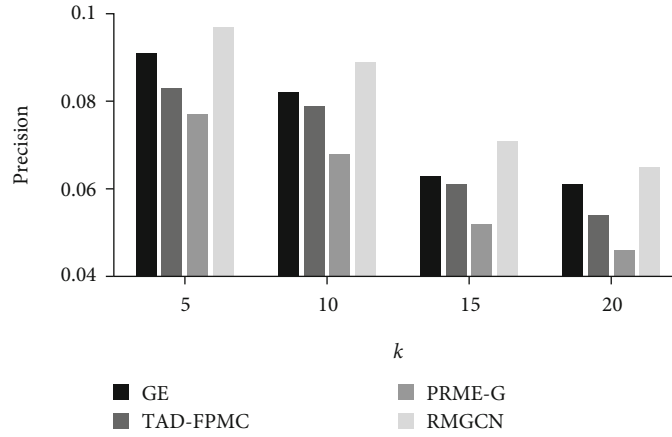


FIGURE 11: Precision on LV dataset.

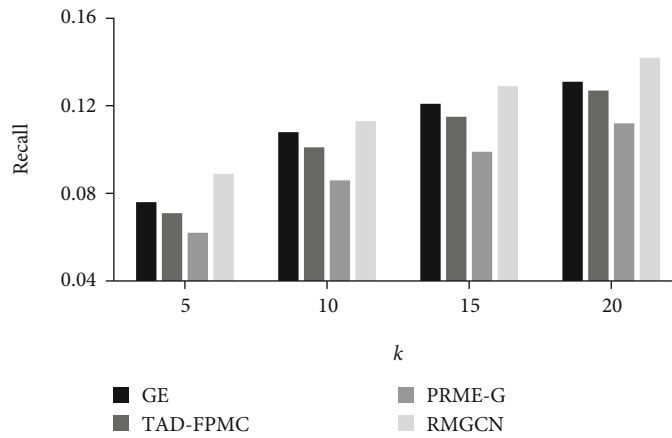


FIGURE 12: Recall on LV dataset.

stronger performance than the baseline model, which to a certain extent indicates that in-depth mining of graph structure information can boost the recommendation performance of the model. GE is not as effective as RMGCN. Though both are recommendation models based on a graph structure, GE is a shallow-layer model that is less capable of in-depth mining node features. Moreover, the user preference features generated by GE depend on the time delay function, which is difficult to apply to cases with sparse data. Both TAD-FPMC and PRME-G suffer from data sparsity because of the large weight of the user check-in sequence in the modeling processes of both. However, when the user check-in data is sparse, the user check-in sequence information fails to be effectively extracted from the data, resulting in a failure to accurately capture user preference features and the degraded performance of the model in the end. Moreover, the performance of various models on the TKY dataset is stronger than that of the LV dataset because the TKY dataset is much denser than the LV dataset (as can be seen from Section 4.1), which also shows that data sparsity has a greater impact on the performance of recommendation models.

6. Conclusion

In order to blunt the effect of data sparsity on recommendation models, a recommendation model of RMGCN based on convolutional neural networks has been presented in this paper from a deep exploration of graph structure data constructed by user check-in records. RMGCN extracts the node features through a convolutional neural network based on edge weight normalization and evaluates the user preference scores for POI from the perspective of temporal and spatial contexts. Experimental results on two real-world datasets make it clear that RMGCN can boast a better recommendation performance than the baseline model.

Subsequent research will investigate how to use graph structure data to deal with the cold start problem of the POI recommendations.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] X. Y. Ren, M. N. Song, and J. D. Song, "Context-aware point-of-interest recommendation in location-based social networks," *Chinese Journal of Computers*, vol. 40, no. 4, pp. 824–841, 2017.
- [2] L. Shi, J. Du, G. Cheng, X. Liu, Z. Xiong, and J. Luo, "Cross-media search method based on complementary attention and generative adversarial network for social networks," *International Journal of Intelligent Systems*, vol. 37, pp. 84393–84416, 2021.
- [3] L. Shi, G. Song, G. Cheng, and X. Liu, "A user-based aggregation topic model for understanding user's preference and intention in social network," *Neurocomputing*, vol. 413, pp. 1–13, 2020.
- [4] J. Zhang and C. Y. Chow, "Geo SoCa: exploiting geographical, social and categorical correlations for point-of-interest recommendations," in *the 38th International ACM SIGIR Conference*, pp. 443–452, New York, 2015.
- [5] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, "Learning graph-based POI embedding for location-based recommendation," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 15–24, New York, 2016.
- [6] X. Li, M. Jiang, H. Hong, and L. Liao, "A time-aware personalized point-of-interest recommendation via high-order tensor factorization," *ACM Transactions on Information Systems*, vol. 35, no. 4, pp. 1–23, 2017.
- [7] J. D. Zhang and C. Y. Chow, "TICRec: a probabilistic framework to utilize temporal influence correlations for time-aware location recommendations," *IEEE Transactions on Services Computing*, vol. 9, no. 4, pp. 633–646, 2017.
- [8] Y. Liu, C. Liu, B. Liu, M. Qu, and H. Xiong, "Unified point-of-interest recommendation with temporal interval assessment," in *the 22nd ACM SIGKDD International Conference*, pp. 1015–1024, ACM, New York, 2016.
- [9] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geo MF: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *the 20th ACM SIGKDD International Conference*, pp. 831–840, ACM, New York, 2014.
- [10] X. Li, C. Gao, X. L. Li, T.-A. N. Pham, and S. Krishnaswamy, "Rank-Geo FM: a ranking based geographical factorization method for point of interest recommendation," in *the 38th International ACM SIGIR Conference*, pp. 433–442, ACM, New York, 2015.
- [11] S. Feng, C. Gao, B. An, and Y. M. Chee, "POI2Vec: geographical latent representation for predicting future visitors," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17)*, pp. 102–108, AAAI Press, California, 2017.
- [12] S. Yang, G. Huang, Y. Xiang, X. Zhou, and C. H. Chi, "Modeling user preferences on spatiotemporal topics for point-of-interest recommendation," in *IEEE International Conference on Services Computing*, pp. 204–211, Honolulu, HI, USA, 2017.
- [13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," <https://arxiv.org/abs/1609.02907>.
- [14] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, <https://arxiv.org/abs/1710.10903>.
- [16] X. He, K. Deng, and X. Wang, "Lightgcn: simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 639–648, ACM, New York, 2020.
- [17] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2009, <https://arxiv.org/abs/1205.2618>.
- [18] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new POI," in *Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 2069–2075, New York ACM, 2015.