HRec: Heterogeneous Graph Embedding-Based Personalized Point-of-Interest Recommendation

Yijun Su[1,3], Xiang Li[1,2,3], Daren Zha[3(✉)], Wei Tang[1,2,3], Yiwen Jiang[1,2,3], Ji Xiang[3], and Neng Gao[2,3]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{suyijun,lixiang9015,tangwei,jiangyiwen}@iie.ac.cn

[2] State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing, China
gaoneng@iie.ac.cn

[3] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zhadaren,xiangji}@iie.ac.cn

Abstract. POI (point-of-interest) recommendation as an important location-based service has been widely utilized in helping people discover attractive locations. A variety of available check-in data provide a good opportunity for developing personalized POI recommender systems. However, the extreme sparsity of check-in data and inefficiency of exploiting unobserved feedback pose severe challenges for POI recommendation. To cope with these challenges, we develop a heterogeneous graph embedding-based personalized POI recommendation framework called HRec. It consists of two modules: the learning module and the ranking module. Specifically, we first propose the learning module to produce a series of intermediate feedback from unobserved feedback by learning the embeddings of users and POIs in the heterogeneous graph. Then we devise the ranking module to recommend each user the ultimate ranked list of relevant POIs by utilizing two pairwise feedback comparisons. Experimental results on two real-world datasets demonstrate the effectiveness and superiority of the proposed method.

Keywords: POI recommendation · Graph embedding · Personalized ranking

1 Introduction

Location-based social networks (LBSNs) have become popular recently because of the increasing proliferation of smart mobile devices with location-acquisition that make people easy to post their real location and location-related contents. These LBSNs like Foursquare, Facebook Places, and Yelp allow users to make friends and share their check-in experiences on Points-of-Interests (POIs), e.g.,

T. Gedeon et al. (Eds.): ICONIP 2019, LNCS 11955, pp. 37–49, 2019.
https://doi.org/10.1007/978-3-030-36718-3_4

restaurants, stores, and museums. Driven by a vast amount of check-in data collected, POI recommendation arises to improve the user experience. It has become an important location-based service to help people explore interesting and attractive places [3].

The task of POI recommendation is to model users' preferences and suggest novel POIs to users. It is a very challenging problem due to two major reasons. First, the check-in data are **extremely sparse**, which significantly increases the difficulty of POI recommender systems. In fact, a single user usually chooses only a small portion from millions of POIs to check in. This will make the user-POI matrix very sparse. In the literature, some researchers have sought to utilize **social information** and **geographical information** to supplement the highly sparse user-POI matrix. Most existing approaches have been proposed to incorporate social relations between users into collaborative filtering (CF) techniques, e.g., friend-based CF [16], matrix factorization with social regularization [1], and friend-based matrix factorization [2]. However, these methods provide considerably limited improvements on POI recommendation because social links of users are also sparse. On the other hand, most related works [1,5,10,12] attempt to establish independent geographical models to recommend POIs. Nonetheless, such modeling approaches only mean that the check-in activity is limited to the distance constraint and do not effectively represent users' preferences. Second, unobserved feedback is **implicit** and its number is **very large**, which will lead to the inefficiency of computation and the inaccuracy of prediction. Some researchers [2,6,8,13] have proposed to ranking-based models to alleviate this situation. Bayesian Personalised Ranking (BPR) [8] is a famous ranking-based model, which learns the ranking based on pairwise preference comparison over observed and unobserved feedback. However, due to the imbalance between users' visited POIs and non-visited POIs, the BPR model cannot successfully enhance prediction accuracy.

More recently, graph embedding methods which embed information networks into low-dimensional vector spaces have been widely adopted for a variety of tasks such as link prediction, text mining, and sentiment analysis [9]. Such low-dimensional representation is denser than the user-POI check-in matrix, so graph embedding is a potential and powerful solution to alleviate the problem of data sparsity. In this paper, we extend these efforts and propose a **H**eterogeneous graph embedding-based personalized POI **Rec**ommendation framework (HRec) to effectively address the aforementioned challenges. The overall architecture of HRec is shown in Fig. 1. Our recommendation framework consists of two modules, one of which is the **learning module** and the other is the **ranking module**. (1) The learning module is to generate a series of intermediate feedback from unobserved feedback by exploiting social and geographical information networks, which is treated as weak preference relative to positive feedback while as strong preference in comparison to other unobserved feedback. The module learns vector representations for the nodes (i.e., users and POIs embeddings) in the heterogeneous graphs and then uses the learned representations for generating intermediate feedback. (2) The ranking module is to recommend each

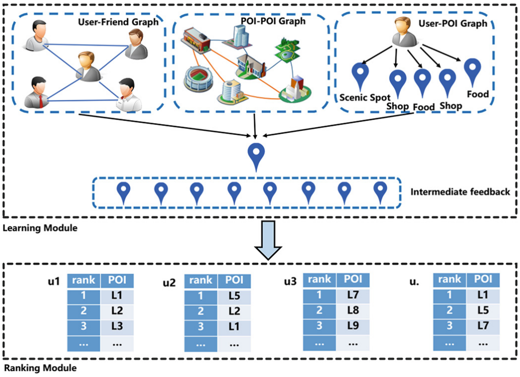

Fig. 1. The architecture framework of HRec.

user a ranked list of relevant POIs that the user might be interested in but has not visited before. In this module, we augment the ranking function of BPR by introducing the intermediate feedback generated by the learning module. Furthermore, we design a mini-batch gradient descent (MBGD) with the bootstrap sampling algorithm to optimize its objective function. Finally, we evaluate the proposed framework on two large-scale real-world datasets and prove its superiority to several state-of-the-art baselines.

To summarize, our work makes the following contributions:

1 We develop a Heterogeneous graph embedding-based personalized POI Recommendation framework (HRec) to overcome the data sparsity issue and inefficiency of exploiting unobserved feedback. The HRec consists of two modules: the learning module and the ranking module.
2 The learning module in HRec is devised for generating a series of intermediate feedback from unobserved feedback by learning the embeddings of users and POIs in the heterogeneous graph.
3 The ranking module in HRec is designed for recommending each user the ultimate ranked list of relevant POIs by utilizing two pairwise feedback comparisons.
4 We conduct extensive experiments on real-world datasets. Experimental results prove the effectiveness and efficiency of the proposed HRec framework.

2 Related Work

In this section, we discuss some existing works related to our research, particularly those employing social and geographical information for POI recommendation. As the main learning and ranking modules fall within the realm of graph embedding and personalized ranking, we also review these related techniques.

Based on the fact that friends are more likely to share common interests, social information is widely used in POI recommender systems [2]. In particular, friend-based collaborative filtering [12] and matrix factorization with social regularization [1] are two effective algorithms in LBSNs, which both integrate social relationship information into the collaborative filtering techniques to improve the quality of POI recommendation. Besides, Zhang et al. [15] designed a model to estimate the social check-in frequency by using a power-law distribution learned from historical check-ins of all users. Since the geographical characteristics of locations can affect users' check-in behavior, geographical information plays an important role in POI recommendation [1,5,10,12,13,15]. On the one hand, geographical distance between users and POIs limits users' check-in choice. On the other hand, as Tobler's First Law of Geography shown, geographical clustering phenomenon is very common in users' check-in activities. In particular, several representative models, such as power law distribution (PD) model [12], Multi-center Gaussian distribution model (MGM) [1], and Kernel Density Estimation (KDE) [14], are proposed to capture the geographical influence in POI recommendation.

Graph embedding techniques that embed information networks into low-dimensional vector spaces have attracted considerable attention and made great progress in recent years. For example, Xie et al. [11] proposed a graph embedding model for POI recommendations to systematically model the POI, user, and time relations and learned the representations. Zhao et al. [17] proposed a temporal POI embedding based on Skip-Gram model to capture users' temporal preference. However, few works based on graph embedding attempt to exploit social relations between users and geographical neighborhood characteristics between POIs for POI recommendations. From the perspective of ranking tasks, these collaborative filtering-based methods mentioned above can be viewed as pointwise methods. Indeed, empirical studies [6,13] have demonstrated that pointwise methods are generally less effective than pairwise ranking methods. Yuan et al. [13] proposed a GeoBPR model that injects users' geo-spatial preference. Manotumruksa et al. [6] developed a novel personalized ranking framework with multiple sampling criteria to enhance the performance of POI recommendation.

In this paper, our work distinguishes itself from previous researches in several aspects. First, to the best of our knowledge, it is the first effort that exploits social relations between users and geographical neighborhood characteristics between POIs to address the challenges of data sparsity and inefficiency of unobserved feedback in a unified way. Second, we generate a series of intermediate feedback from unobserved feedback in the learning module to augment the ranking function of Bayesian Personalised Ranking (BPR) [8]. Moreover, we integrate the embeddings of users and POIs and BPR in a systematic way for POI recommendations.

3 Problem Statement

Let users and POIs denoted by $\mathcal{U} = \{u_1, u_2, ...\}$ and $\mathcal{L} = \{l_1, l_2, ...\}$. Each user u checked in some POIs $\mathcal{L}_u$. Each POI has a location $l_j = \{lon_j, lat_j\}$ in terms of longitude and latitude. We use $\mathcal{F}_u = \{f_1, f_2, ...\}$ to represent the set of the user's friends. In this paper, we consider three different types of feedback, namely positive, intermediate, negative feedback. The positive feedback is defined as a set of POIs previously checked in by user u: $P_u = \mathcal{L}_u$. The intermediate feedback $I_u = \{l_1, ..., l_c\}$ is learned from unvisited POIs in the learning module. The remaining unvisited POIs are viewed as the negative feedback $N_u = \{l_1, ..., l_h\}$. Here *negative* only means no explicit feedback can be observed from the user and does not denote users' dislike of the POIs.

Definition 1. ***User-POI Graph****, denoted as $G_{ul} = (\mathcal{U} \cup \mathcal{L}, \mathcal{E}_{ul})$, is a bipartite graph where $\mathcal{E}_{ul}$ is the set of edges between users and POIs. The weight w_{ul} between user u and POI l is simply defined as the frequency of user u checked in POI l.*

Definition 2. ***User-Friend Graph****, denoted as $G_{uf} = (\mathcal{U} \cup \mathcal{F}, \mathcal{E}_{uf})$, is a social relation graph where $\mathcal{F}$ is a set of users' friends and $\mathcal{E}_{uf}$ is the set of edges between users and friends. The weight w_{uf} between user u and friend f is defined as common check-in ratio between user u and his friend f, which is measured by $\frac{|\mathcal{L}_u \cap \mathcal{L}_f|}{|\mathcal{L}_u \cup \mathcal{L}_f|}$.*

Definition 3. ***POI-POI Graph****, denoted as $G_{ll} = (\mathcal{L} \cup \mathcal{L}, \mathcal{E}_{ll})$, captures the geographical neighborhood characteristics between POIs. In general, if POI l_i is a geographical neighbor of POI l_j, there will be an edge between l_i and l_j. The weight w_{ij} of the edge between l_i and l_j is set to 1 when POI l_i is the neighbors in geographical space to POI l_j.*

Problem 1 **(POI Recommendation).** Given a user check-in record $\mathcal{L}_u$, the geographical coordinates of POIs and the user's social friends $\mathcal{F}_u$, the task of POI recommendation is to generate a ranked list of POIs that the user might be interested in but has not visited before in LBSNs.

4 POI Recommendation Framework

4.1 Learning Module

In this module, the aim is to generate a series of intermediate feedback from unobserved feedback by learning the user and POI embeddings of heterogeneous information networks. We adopt the bipartite graph embedding approach from Tang et al. [9], which is a representation learning method for heterogeneous text networks.

Bipartite Graph Embedding. Given a bipartite graph $G_{AB} = (\mathcal{V}_A \cup \mathcal{V}_B, \mathcal{E})$, where $\mathcal{V}_A$ and $\mathcal{V}_B$ are two disjoint sets of vertices of different types, and $\mathcal{E}$ is the set of edges between them. The conditional probability of vertex v_i in set $\mathcal{V}_A$ generated by vertex v_j in set $\mathcal{V}_B$ can be defined as:

$$p(v_i|v_j) = \frac{exp(\boldsymbol{z}_i^T \cdot \boldsymbol{z}_j)}{\sum_{v_k \in \mathcal{V}_A} exp(\boldsymbol{z}_k^T \cdot \boldsymbol{z}_j)} \tag{1}$$

where $\boldsymbol{z}_i$ denotes the embedding vector for vertex v_i, and $\boldsymbol{z}_j$ is the embedding vector of vertex v_j. For each vertex v_j in $\mathcal{V}_B$, Eq. (1) defines a conditional distribution $p(\cdot|v_j)$ over all the vertices in the set $\mathcal{V}_A$. For each edge e_{ij}, its empirical distribution is given by $\hat{p}(v_i|v_j) = \frac{w_{ij}}{deg_j}$, where w_{ij} is the edge weight between v_i and v_j and $deg_j = \sum_{i \in \mathcal{V}_A} w_{ij}$.

To learn embeddings, we make the conditional distribution $p(\cdot|v_j)$ closely approximates the empirical distribution $\hat{p}(\cdot|v_j)$. Hence, we minimize the following objective function over the graph G_{AB}:

$$O_{AB} = \sum_{j \in \mathcal{V}_B} \lambda_j d(\hat{p}(\cdot|v_j), p(\cdot|v_j)) \tag{2}$$

where $d(\cdot,\cdot)$ is the KL-divergence between two distributions, and λ_j is the importance of vertex v_j in the graph, which can be set as the degree deg_j. Omiting some constants, the objective function can be writen as:

$$O_{AB} = -\sum_{(i,j) \in \mathcal{E}} w_{ij} \log p(v_i|v_j) \tag{3}$$

Optimizing the objective function Eq. (3) is computationally expensive, which requires the summation over the entire set of vertices when calculating the conditional probability $p(\cdot|v_j)$. To overcome this problem, we use the techniques of edge sampling [9] and negative sampling [7]. For each edge e_{ij}, its final objective function is:

$$O_{AB} = -\sum_{(i,j) \in \mathcal{E}} \left[\log \sigma(\boldsymbol{z}_i^T \cdot \boldsymbol{z}_j) + \sum_{n=1}^{K} \mathcal{E}_{v_n \sim P_n(v)} \log \sigma(-\boldsymbol{z}_n^T \cdot \boldsymbol{z}_j) \right] \tag{4}$$

where $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function, K is the number of negative edges. In our implementation, we set $K = 5$, $P_n(v) \propto d_v^{3/4}$ from the empirical setting of [7], where d_v is the out-degree of node v.

Joint Training Learning. The heterogeneous information network is composed of three bipartite graphs: User-POI, User-Friend and POI-POI. To collectively embed the three bipartite graphs, minimizing the sum of all objective functions as following:

$$O = O_{ul} + O_{uf} + O_{ll} \tag{5}$$

where

$$O_{ul} = -\sum_{(i,j) \in \mathcal{E}_{ul}} w_{ij} \log p(u_i|l_j) \tag{6}$$

$$O_{uf} = -\sum_{(i,j) \in \mathcal{E}_{uf}} w_{ij} \log p(u_i|f_j) \tag{7}$$

$$O_{ll} = -\sum_{(i,j) \in \mathcal{E}_{ll}} w_{ij} \log p(l_i|l_j) \tag{8}$$

We learn user and POI embeddings by joint training the three bipartite graphs. In each step, we adopt the asynchronous stochastic gradient algorithm (ASGD) to update the model parameters. See Algorithm 1 for more details. Finally, we sort all unobserved POIs in accordance with their scores $s = \boldsymbol{z}_u^T \boldsymbol{z}_l$

to acquire the Top-t as intermediate feedback for each user, where $\boldsymbol{z}_u, \boldsymbol{z}_l$ are embeddings for user u, POI l and t is the number of intermediate feedback we defined.

Algorithm 1. Joint training

Input: Bipartite graphs (User-POI graph G_{ul}, User-Friend graph G_{uf}, POI-POI graph G_{ll}), number of samples T, number of negative samples K, vector dimension d.
Output: users embeddings: $Z_u \in \mathbb{R}^{|\mathcal{U}| \times d}$ and POI embeddings $Z_l \in \mathbb{R}^{|\mathcal{L}| \times d}$
1: **while** $iter \leq T$ **do**
2: sample an edge from $\mathcal{E}_{ul}$ and draw K negative edges, and update the user and POI embeddings;
3: sample an edge from $\mathcal{E}_{uf}$ and draw K negative edges, and update the user embeddings;
4: sample an edge from $\mathcal{E}_{ll}$ and draw K negative edges, and update the POI embeddings;
5: **end while**

4.2 Ranking Module

In this module, we augment the ranking function of BPR by introducing the intermediate feedback. Specifically, we treat the intermediate feedback as weak preference relative to positive feedback while as strong preference in comparison to other unobserved feedback. Compared with the basic assumption of BPR, our assumption can mine more contribution information from unobserved POIs. Thus, for user u, the ranking order of her preference over positive feedback $i \in P_u$, intermediate feedback $c \in I_u$, and negative feedback $j \in N_u$ is given as the following:

$$\begin{cases} \hat{r}_{ui} > \hat{r}_{uc} \\ \hat{r}_{uc} > \hat{r}_{uj} \end{cases} \Rightarrow \begin{cases} W_u H_i^T + b_i > W_u H_c^T + b_c \\ W_u H_c^T + b_c > W_u H_j^T + b_j \end{cases} \tag{9}$$

where $\hat{r}_{ui}$ is the predicted users' preference score, which is modelled by matrix factorization, i.e., $\hat{r}_{ui} = W_u H_i^T + b_i$. The W_u and H_i^T denotes latent feature vectors of user u and POI i, respectively. The b_i is the bias term of POI i. Thus, model parameters $\Theta = \left\{W \in \mathbb{R}^{|\mathcal{U}| \times k}, H \in \mathbb{R}^{|\mathcal{L}| \times k}, b \in \mathbb{R}^{|\mathcal{L}|}\right\}$.

Due to the BPR method gives equal weight to each POI pair, it does not distinguish between their different contributions in learning the objective function. To address this limitation, we assign a higher weight to highlight its contribution. To this end, we propose the augmented bayesian personalized ranking function based on matrix factorization to compute the ranking loss function, given by:

$$\begin{aligned} J(\Theta) = \min_{W,H} - \sum_{u \in \mathcal{U}} \Bigg[& \sum_{i \in \mathcal{P}_u} \sum_{c \in \mathcal{I}_u} \ln \sigma \left(c_{uic}(\hat{r}_{ui} - \hat{r}_{uc})\right) \\ & + \sum_{c \in \mathcal{I}_u} \sum_{j \in \mathcal{N}_u} \ln \sigma (\hat{r}_{uc} - \hat{r}_{uj}) \Bigg] \\ & + \lambda_\Theta ||\Theta||^2 \end{aligned} \tag{10}$$

where c_{uic} denotes the weight of the difference between positive and intermediate feedback, and its value is determined by the difference of two visit frequencies $c_{uic} = 1 + \alpha f_{ui}$, where α is a tuning parameter and f_{ui} represents the check-in frequency of user u on POI i. λ_Θ are model specific regularization parameters and $\sigma(x)$ is the sigmoid function.

Algorithm 2. Ranking Algorithm

Input: feedback data: user $u \in \mathcal{U}$, positive feedback $\mathcal{P}_u$, intermediate feedback $\mathcal{I}_u$, and negative feedback $\mathcal{N}_u$
hyperparameters: sampling times st, batch size bs, learning rate η, and regularization parameters $\lambda_u, \lambda_i, \lambda_c, \lambda_j, \beta_i, \beta_c, \beta_j$
Output: model parameters $\Theta = \{W, H, b\}$

Initialization Θ with Normal distribution $\mathcal{N}$(0,0.1)
for $t = 1$ to st **do**
  Uniformly sample a user u from $\mathcal{U}$
  Uniformly sample a positive feedback i from $\mathcal{P}_u$
  Uniformly sample a intermediate feedback c from $\mathcal{I}_u$
  Uniformly sample a negative feedback j from $\mathcal{N}_u$
end for
$s = 0$
while $(s+1) * bs \leq st$ **do**
  for $j = 1$ to bs **do**
    $\hat{r}_{uic} = (1 - \sigma(c_{uic}(\hat{r}_{ui} - \hat{r}_{uc}))) \cdot c_{uic}, \hat{r}_{ucj} = 1 - \sigma((\hat{r}_{uc} - \hat{r}_{uj}))$
    $W_u \leftarrow W_u + \eta\,([\hat{r}_{uic}(H_i - H_c) + \hat{r}_{ucj}(H_c - Hj)] - \lambda_u W_u)$
    $H_i \leftarrow H_i + \eta\,(\hat{r}_{uic} W_u - \lambda_i H_i)$
    $H_c \leftarrow H_c + \eta\,(-\hat{r}_{uic} W_u + \hat{r}_{ucj} W_u - \lambda_c H_c)$
    $H_j \leftarrow H_i + \eta\,(-\hat{r}_{ucj} W_u - \lambda_i H_i)$
    $b_i \leftarrow b_i + \eta\,(\hat{r}_{uic} - \beta_i b_i)$
    $b_c \leftarrow b_c + \eta\,(-\hat{r}_{uic} + \hat{r}_{ucj} - \beta_c b_c)$
    $b_j \leftarrow b_j + \eta\,(-\hat{r}_{ucj} - \beta_j b_j)$
  end for
  $s = s + 1$
end while
return Θ

We propose a Mini-batch Gradient Descent (MBGD) with the bootstrap sampling to optimize the objective function. See Algorithm 2 for more details.

5 Experimental Evaluation

5.1 Datasets

We make use of two publicly available real-world datasets, Gowalla [4] and Foursquare [2], to evaluate the performance of the proposed framework. Each check-in record contains a user ID, a location ID, a timestamp and geo-coordinates of the location. Also, data sets have social links information. The data statistics are shown in Table 1. In our experiments, we divide each dataset into training set, tuning set and test set in terms of the user's check-in time instead of choosing a random partition method. For each user, the earliest 70% check-ins are selected for training, the most recent 20% check-ins as testing, and the next 10% as tuning.

Table 1. Statistical information of the two datasets

Statistical item	Gowalla	Foursquare
Number of users	5,628	2,551
Number of POIs	31,803	13,474
Number of check-ins	620,683	124,933
Number of social links	46,001	32,512
User-POI matrix density	0.22%	0.291%

5.2 Evaluation Metrics

We use two widely-used metrics [4] to evaluate the performance of the model we proposed: precision (Pre@N) and recall (Rec@N), where N is the number of recommended POIs. Pre@N refers to the ratio of recovered POIs to the top-N recommended POIs and Rec@N measures the ratio of recovered POIs to the set of visited POIs in the testing data.

5.3 Baseline Methods

To illustrate the effectiveness of our recommendation framework, we compare it with the following state-of-the-art methods.

- **Random**: Random method is to recommend users with random POIs.
- **BPR-KNN**: This is a ranking-based adaptive model, which employs item-based k-nearest-neighbor to recommend POIs [8].
- **BPR-MF**: This is a classical pairwise ranking model based on matrix factorization [8].
- **GeoBPR**: This is a state-of-the-art method for POI recommendation, which incorporates the geographic feedback into the BPR model [13].

5.4 Parameter Settings

For all the compared baselines, we adopt the optimal parameter configuration reported in their works. In our experiments, all critical parameters are tuned through cross-validation. Empirically, for the learning module, the vector dimension d is set to 100, the tuning parameter α is set to 0.5 and the number of intermediate feedback $t = 2000$. In Foursquare dataset, the learning rate η is set to 0.001, the latent factor dimension $k = 40$, and regularization parameters $\lambda_u = 0.005$, $\lambda_i = \lambda_c = \lambda_j = 0.005$, $\beta_i = \beta_c = \beta_j = 0.006$. In Gowalla dataset, the learning rate η is set to 0.005, the latent factor dimension $k = 30$, and regularization parameters $\lambda_u = 0.005$, $\lambda_i = \lambda_c = \lambda_j = 0.005$, $\beta_i = \beta_c = \beta_j = 0.003$. The effect of the latent factor dimension k will be detailed later.

5.5 Experimental Results

Performance Comparisons. Results of all POI recommendation models in terms of Pre@N and Rec@N on Foursquare and Gowalla are presented in Figs. 2 and 3, respectively. One can observe that HRec framework always outperforms all the compared POI recommendation methods on the two datasets. On the one hand, compared with non-ranking algorithm Random, our recommendation framework presents an absolute advantage. In fact, Random model outputs the lowest performance. For example, in terms of Pre@5 and Rec@5, HRec attains 0.044, 0.0251 and 0.0298, 0.0134 on Foursquare and Gowalla datasets, respectively. On the other hand, our framework significantly outperforms other three ranking algorithms BPR-KNN, BPR-MF and GeoBPR. For instance, HRec improves the second best recommendation algorithm GeoBPR by 33.3%, 39% and 2.5%, 1.4% in terms of Pre@5, Rec@5 on Foursquare and Gowalla, respectively. Based on the performance comparison of non-ranking and ranking algorithms, the effectiveness and superiority of the proposed method HRec are proved. The reasons are two fold: (1) HRec makes full of social and geographical information by learning the embeddings of users and POIs in the heterogeneous

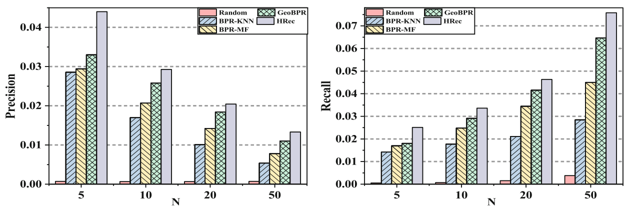

Fig. 2. Varying N on Foursquare

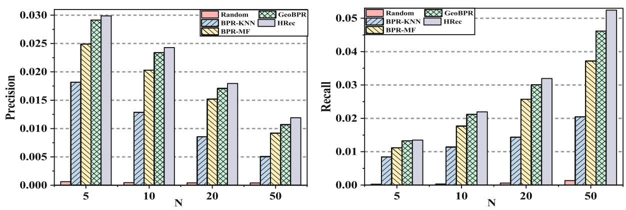

Fig. 3. Varying N on Gowalla

graph. (2) HRec effectively exploits a series of intermediate POIs learned from unvisited POIs and utilizes two pairwise feedback comparisons to greatly assist ranking.

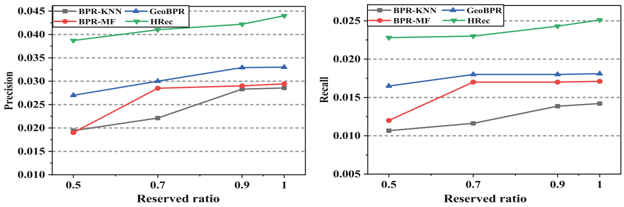

Fig. 4. Impact of data sparsity

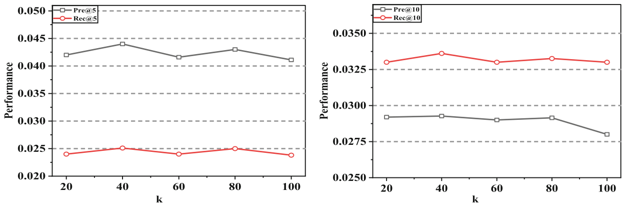

Fig. 5. Influence of latent factor dimensions k

Impact of Data Sparsity. Here, we study how HRec deals with the data sparsity problem. In order to produce user-POI check-in matrix with different sparsity, we randomly reserve $x\%$ ($x = 50{,}70{,}90{,}100$) of check-ins from each user's visited records. The smaller the reserved ratio x is, the sparser the user-POI check-in matrix is. Figure 4 reports Pre@5 and Rec@5 of all recommendation algorithms on Foursquare under different sparsity. Due to Random outputs poor performance, it is not added here for comparison. Based on the results, we can observe that the Pre@5 and Rec@5 of all algorithms increase with the increase of the reserved ratio x. One possible explanation is that, with the increase of the proportion of the training set, the number of positive examples increases, and then contributes to the improvement. We can further observe that our framework HRec consistently outperforms all ranking and non-ranking baselines under various data sparsity scenarios, which shows great strengths.

Parameter Sensitivity. In this study, we employ matrix factorization to predict the difference between the two scores of preference for users. Hence, in this section, we study the influence of variable k, which is the number of latent feature dimension. Due to limited space, we only show the performance of the recommendation on Foursquare dataset. In our experiment, k is set to 20, 40, 60, 80 and 100, respectively. Figure 5 reports the recommended quality for different values of k. Based on the results, we can observe that the performance in all evaluation metrics has similar behaviour with the varying value of k. The performance increases with the increase of the k at the beginning, then hits the highest recommended quality when $k = 40$, and eventually tends to decline. The above trend indicates that the performance achieves best at $k = 40$, and so we finally choose the optimal parameter $k = 40$.

6 Conclusions

This paper presents a novel personalized POI recommendation framework called the HRec, which can address the data sparsity issue and inefficiency of exploiting unobserved feedback. The HRec consists of two modules: the learning module and the ranking module. The learning module is designed for producing a series of intermediate feedback from unobserved feedback by learning the embeddings of users and POIs in the heterogeneous graph. The ranking module is devised for recommending each user the ultimate ranked list of relevant POIs by effectively exploiting intermediate feedback generated by the learning module. Experimental results on two real-world datasets demonstrate that HRec performs better than other compared models for POI recommendations.

Acknowledgments. This work is supported by the National Key Research and Development Program of China, and National Natural Science Foundation of China (No. U163620068).

References

1. Cheng, C., Yang, H., King, I., Lyu, M.R.: A unified point-of-interest recommendation framework in location-based social networks. ACM TIST **8**(1), 10:1–10:21 (2016)
2. Li, H., Ge, Y., Hong, R., Zhu, H.: Point-of-interest recommendations: learning potential check-ins from friends. In: ACM KDD, pp. 975–984 (2016)
3. Lian, D., et al.: Scalable content-aware collaborative filtering for location recommendation. IEEE TKDE **30**(6), 1122–1135 (2018)
4. Liu, Y., Pham, T.A.N., Cong, G., Yuan, Q.: An experimental evaluation of point-of-interest recommendation in location-based social networks. In: VLDB, pp. 1010–1021 (2017)
5. Ma, C., Zhang, Y., Wang, Q., Liu, X.: Point-of-interest recommendation: exploiting self-attentive autoencoders with neighbor-aware influence. In: ACM CIKM, pp. 697–706 (2018)
6. Manotumruksa, J., Macdonald, C., Ounis, I.: A personalised ranking framework with multiple sampling criteria for venue recommendation. In: ACM CIKM, pp. 1469–1478 (2017)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS, pp. 3111–3119 (2013)
8. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461 (2009)

9. Tang, J., Qu, M., Mei, Q.: PTE: predictive text embedding through large-scale heterogeneous text networks. In: ACM KDD, pp. 1165–1174 (2015)
10. Wang, H., Shen, H., Ouyang, W., Cheng, X.: Exploiting poi-specific geographical influence for point-of-interest recommendation. In: IJCAI, pp. 3877–3883 (2018)
11. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. In: ACM CIKM, pp. 15–24 (2016)
12. Ye, M., Yin, P., Lee, W.C., Lee, D.L.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: ACM SIGIR, pp. 325–334 (2011)
13. Yuan, F., Guo, G., Jose, J., Chen, L., Yu, H.: Joint geo-spatial preference and pairwise ranking for point-of-interest recommendation. In: IEEE ICTAI, pp. 46–53 (2016)
14. Zhang, J.D., Chow, C.Y.: iGSLR: personalized geo-social location recommendation: a kernel density estimation approach. In: ACM SIGSPATIAL, pp. 334–343 (2013)
15. Zhang, J.D., Chow, C.Y.: GeoSoCa: exploiting geographical, social and categorical correlations for point-of-interest recommendations. In: ACM SIGIR, pp. 443–452 (2015)
16. Zhang, J.D., Chow, C.Y., Li, Y.: Lore: exploiting sequential influence for location recommendations. In: ACM SIGSPATIAL, pp. 103–112 (2014)
17. Zhao, S., Zhao, T., King, I., Lyu, M.R.: Geo-teaser: geo-temporal sequential embedding rank for point-of-interest recommendation. In: WWW, pp. 153–162 (2017)