Contextualized Point-of-Interest Recommendation

Peng Han[2*] , **Zhongxiao Li**[2*] , **Yong Liu**[3] , **Peilin Zhao**[4] , **Jing Li**[5] , **Hao Wang**[5†] and **Shuo Shang**[1†]

[1]University of Electronic Science and Technology of China
[2]King Abdullah University of Science and Technology
[3]Nanyang Technological University
[4]Tencent AI Lab
[5]Inception Institute of Artificial Intelligence
{peng.han, zhongxiao.li}@kaust.edu.sa, stephenliu@ntu.edu.sg, {peilinzhao, jingli.phd}@hotmail.com,
{haowang.paper,jedi.shang}@gmail.com

Abstract

Point-of-interest (POI) recommendation has become an increasingly important sub-field of recommendation system research. Previous methods employ various assumptions to exploit the contextual information for improving the recommendation accuracy. The common property among them is that similar users are more likely to visit similar POIs and similar POIs would like to be visited by the same user. However, none of existing methods utilize similarity explicitly to make recommendations. In this paper, we propose a new framework for POI recommendation, which explicitly utilizes similarity with contextual information. Specifically, we categorize the context information into two groups, *i.e.*, global and local context, and develop different regularization terms to incorporate them for recommendation. A graph Laplacian regularization term is utilized to exploit the global context information. Moreover, we cluster users into different groups, and let the objective function constrain the users in the same group to have similar predicted POI ratings. An alternating optimization method is developed to optimize our model and get the final rating matrix. The results in our experiments show that our algorithm outperforms all the state-of-the-art methods.

1 Introduction

Point of interest (POI) recommendation has become an increasingly important sub-field of recommendation system [Yang *et al.*, 2018; Liu *et al.*, 2018; Wu *et al.*, 2019] and aims to find new places for users that they might be interested in. It can help users find interesting spots that will make them enjoy their vacations when they are in unfamiliar regions. And it can also increase the shopkeepers' income by attracting more customers who would like to spend time and money at the store. Therefore, POI recommendation has become a hot research topic in recent years [Zheng, 2012; Zheng and Xie, 2011]. However, there are many challenges in this problem and one of the most challenging one is the data sparsity problem.

To tackle this problem, many methods incorporate the contextual information into the recommendation method with different assumptions. For example, IRenMF [Liu *et al.*, 2014b] assumes that the user will visit new POIs that are close to the POIs they visited before. And they construct an auxiliary label matrix by adding the weighted sum of neighboring POIs' ratings to every POI. LRT [Gao *et al.*, 2013] assumes that users will have different preference patterns in different time slots, so they construct different models for different time intervals. Although the assumptions are various, the common property behind them is that similar users should visit similar POIs and similar POIs should be visited by similar users. And the only difference between these assumptions is the way they construct the similarity, for example the similarity used in IRenMF is geographic distance and the similarity used in LRT is the time difference in one day.

However, there are two main drawbacks in their ways of utilizing contextual information. One is that they usually consider only one type of contextual information in one entity, *e.g.*, geographic distance between POIs or relationship between users. And they design the model specially for a specific type of context, making the models lack extensibility. The other problem is that they are not utilizing the contextual information explicitly, as most of the models focuse on check-in history, making contextual information utilization only to be an accessory component in the objective function. In this way, contextual information, which is a key point for the performance of POI recommendation, cannot be fully utilized.

To make the model extensible and the utilization of contextual information completely, we propose a new framework for POI recommendation. In our method, we construct one user matrix and one POI similarity matrix according to the corresponding user and POI contextual information. A lot of types of similarity can be computed by cosine similarity between feature vectors between two entities. Furthermore, different types of similarity can be combined as a weighted sum. In

[*] Equal Contribution.

[†] Corresponding Author.

this way, our framework is extensible for a large class of contextual information. Once the similarity matrices of users and POIs are constructed, we will use two global Laplacian regularization terms to constrain the predicted preference matrix. These can directly make sure that, in the final prediction, similar users should visit similar POIs and similar POIs should be visited by similar users. In addition, to exploit the contextual information hierarchically, we also impose a local regularizer to make the predicted preference matrix have local patterns. Based on user similarity, we use spectral clustering [Von Luxburg, 2007] to sort users into different groups. Then we impose an l_2-norm as a regularization term for the predicted preference matrix of every group, which can make the preference of users in the same group to be sparse and have similar patterns.

We form the objective function by putting the global and local regularizers together. To solve this optimization problem efficiently, we propose an alternating optimization method which decompose the objective function into two parts with an auxiliary variable. Accelerated proximal gradient (APG) algorithm is used to optimize the l_2-regularized part of the problem.

The contributions made in this work are as follows: (1) We propose a new framework for POI recommendation, which focuses on the explicit utilization of contextual information; (2) We can utilize different types of contextual information of both user and POI in our method; (3) We categorize contextual information into global and local types and utilize them by different regularization terms respectively; (4) We design an alternating optimization method to optimize the model; (5) The results of our method outperform the state-of-the-art methods on two large datasets.

2 Related Works

Various methods have been proposed to utilize the context information for POI recommendation. One group of methods exploit the user-based context. In RankGeoFM [Li *et al.*, 2015], the authors proposed two different latent factors for each user, one for the preference to the target POI and the others for neighboring POIs. Another group of methods focus on exploiting the POI-based context. For example, IRenMF [Liu *et al.*, 2014b] utilizes similarity between different POIs based on the geographic distance [Liu *et al.*, 2014b]. For every POI and a specific user, it uses a weighted sum of the user's preference of the nearby POIs to estimate the rating of this POI. And it constrains the latent factors of the POIs in the same region with similar patterns. Although it only utilizes the geographic distances between POIs, the performance of this work is ranked top according to the latest survey [Liu *et al.*, 2017]. Moreover, there also exist some methods exploit hybrid context for recommendation. For instance, GeoMF [Lian *et al.*, 2014] splits the whole space into different regions, and it incorporates the users' interest and POIs' influence of these regions into the their model. Then the final preference score is combined by individual and regional components.

3 Preliminaries

This section introduces some background about the POI recommendation problem and graph Laplacian regularization.

3.1 Problem Definition

Suppose in the recommendation task the total number of users is m and the total number of POIs is n. Let $U = \{u_1, u_2, \ldots, u_m\}$ be the set of users and let $V = \{v_1, v_2, \ldots, v_n\}$ be the set of POIs. Let P_u be the set of indices of POIs that is visited by user u ($u \in U$). Given the past check-in transaction history D, the task of POI recommendation is to recommend for each user u, a new set of POI indices $\hat{P}_u$ ($P_u \bigcap \hat{P}_u = \emptyset$) that match their preference. The transaction history D is a set of tuples of the users and their visited POIs, *i.e.*, $D = \{(u, v) | u \in U, v \in V\}$. The contextual information of the users (*e.g.*, social relations) and the POIs (*e.g.*, geographical coordinates) are also available to the recommendation system while making recommendations. As in a typical supervised learning setting, D serves as the training set. An additional set of transactions D^{Te}, which contains check-ins made by each user unobserved by the recommendation system, serves as the test set. Let P_u^{Te} denote the set of indices of POIs that is visited by user u in the test set. The quality of recommendation is evaluated by the size of overlap between $\hat{P}_u$ and P_u^{Te}. For convenience, the transaction history set, D, is also represented as a rating matrix, $\mathbf{Y}$, where $Y_{ij} = 1$ if $(u_i, v_j) \in D$ and $(\mathbf{Y})_{ij} = 0$ if $(u_i, v_j) \notin D$.

3.2 Graph Laplacian Regularization

Let G be a weighted undirected graph, where $\mathcal{V} = \{\nu_1, \nu_2, \ldots, \nu_{|\mathcal{V}|}\}$ is its vertex set, and its weight matrix be $\mathbf{W} = [W_{ij}]_{i,j=1\ldots|\mathcal{V}|}$, where W_{ij} denotes the weight of edge between ν_i and ν_j. $\mathbf{W}$ is a symmetric matrix, *i.e.*, $W_{ij} = W_{ji}$ as G is undirected. The degree matrix $\mathbf{D}$ of G is a diagonal matrix, $\mathrm{diag}\left(d_1, d_2, \ldots, d_{|\mathcal{V}|}\right)$, where $d_i = \sum_{j=1}^{|\mathcal{V}|} W_{ij}$. Then, the normalized Laplacian matrix of G is defined as, $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$. Let $f : \mathcal{V} \rightarrow \mathbb{R}$ be a real-valued function defined on the vertex space $\mathcal{V}$. The normalized Laplacian regularization of f on graph G is defined as the quadratic form,

$$\mathcal{L}_f = f(\mathcal{V})^\top \mathbf{L} f(\mathcal{V}) \tag{1}$$

where $f(\mathcal{V}) = \left[f(\nu_1), f(\nu_2), \ldots, f(\nu_{|\mathcal{V}|})\right]^\top$ is the vector formed by applying function f on the vertex space $\mathcal{V}$.

4 The Proposed Recommendation Model

This section presents the details of the proposed recommendation framework. The objective of our model is to predict a rating matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ by optimizing an objective function. Each element $R_{i,j}$ in $\mathbf{R}$ represents the inferred preference of user u_i over POI v_j. The new POIs for user u_i are then recommended based on the values of $R_{i,1}, R_{i,2}, \ldots, R_{i,n}$. The objective function includes three regularization terms on $\mathbf{R}$ and each of them will explicitly utilize user-based global contextual information, POI-based global contextual information, and local contextual information, respectively. Finally, we describe how the different regularization terms are put together and how the optimization is carried out.

4.1 Exploiting Global Context Information

User-based Global Context: To incorporate the user-based global contextual information in our framework, we assume that similar users should have similar ratings on a particular POI. The user similarity is computed from the user-based global contextual information (*e.g.*, social relationship information between users). Let G_{user} be a weighted undirected graph, where the vertex set is the user set U. The edge weights are given by a symmetric weight matrix $\mathbf{W}_{user} = \left[W_{ij}^{user}\right]_{i,j=1\ldots m}$, where W_{ij}^{user} is the similarity between user i and user j. Here, we assume that $\mathbf{W}_{user}$ is given and the description of how it is constructed from specific types of user-based global contextual information is postponed to Section 4.3. As in Section 3.2, the degree matrix of G_{user} is $\mathbf{D}_{user} = \mathrm{diag}(d_1^{user}, d_2^{user}, \ldots, d_m^{user})$ and the normalized Laplacian matrix of G_{user} is $\mathbf{L}_{user} = \mathbf{I} - \mathbf{D}_{user}^{-1/2}\mathbf{W}_{user}\mathbf{D}_{user}^{-1/2}$. Suppose there is a rating matrix $\mathbf{R} \in \mathbb{R}^{m\times n}$, where R_{ij} represents the rating of user u_i on POI v_j. We define the normalized graph Laplacian regularization of $\mathbf{R}$ on G_{user} *for a particular POI* v_j as $\mathcal{L}_{user}(\mathbf{R}_{:,j}) = \mathbf{R}_{:,j}^{\top}\mathbf{L}_{user}\mathbf{R}_{:,j}$, where ":" indicates taking all items of a row/column. Then, we can have

$$\mathcal{L}_{user}(\mathbf{R}_{:,j}) = \sum_{i=1}^{m}\sum_{k=1}^{m} W_{ik}^{user}\Big[\frac{R_{ij}}{\sqrt{d_i^{user}}} - \frac{R_{kj}}{\sqrt{d_k^{user}}}\Big]^2. \quad (2)$$

From Eq. (2), it is clear to see that if we incorporate $\mathcal{L}(\mathbf{R}_{:,j})$ in the loss function, it will encourage more similar users to have more similar ratings on POI v_j. From this observation, we sum all the normalized graph Laplacian regularization of $\mathbf{R}$ on graph G_{user} *for all POIs*, and have

$$\mathcal{L}_{user}(\mathbf{R}) = \sum_{j=1}^{n}\mathcal{L}_{user}(\mathbf{R}_{:,j}) = \mathrm{trace}(\mathbf{R}^{\top}\mathbf{L}_{user}\mathbf{R}), \quad (3)$$

which can be used as the user-based Laplacian regularization term for the rating matrix $\mathbf{R}$. Eq. (3) explicitly uses contextual information to encourage similar users to have similar ratings on all POIs.

POI-based Global Context: The POI-based global contextual information is utilized similarly as its user-based counterpart. With POI-based global contextual information, we can build a similarity graph in the POI space as we have done for users. We can define G_{poi}, $\mathbf{W}_{poi}$, $\mathbf{D}_{poi}$ and $\mathbf{L}_{poi}$ in the same way. For a given rating matrix $\mathbf{R}$, its normalized Laplacian regularization on G_{poi} *for a particular user* u_i is $\mathcal{L}_{poi}(\mathbf{R}_{i,:}) = \mathbf{R}_{i,:}\mathbf{L}_{poi}\mathbf{R}_{i,:}^{\top}$. Same as in Eq. (3), its POI-based Laplacian regularization term *for all users* is

$$\mathcal{L}_{poi}(\mathbf{R}) = \sum_{i=1}^{m}\mathcal{L}_{poi}(\mathbf{R}_{i,:}) = \mathrm{trace}(\mathbf{R}\mathbf{L}_{poi}\mathbf{R}^{\top}), \quad (4)$$

which is used as the POI-based Laplacian regularization term for the rating matrix $\mathbf{R}$. This Laplacian regularization explicitly encourage similar POIs to be rated similarly by all users.

4.2 Exploiting Local Context Information

The regularization terms introduced in Section 4.1 can effectively utilize *global* context information between users or POIs. We can understand "global" through Eq. (2) where every pairs of users are enumerated and their squared differences are summed together and adjusted by the global similarity score W_{ik}^{user} existing between any pairs of users. In this section, we introduce a regularizer of another type that takes into account *local* contextual information between users and POIs. The local regularizer is also based on the assumption that similar users should have similar POI ratings. But the local regularizer divide users into groups, therefore making the similarity "local". To distribute similar users into groups, we apply spectral clustering [Von Luxburg, 2007] on the laplacian matrix ($\mathbf{L}_{user}$) of user similarity graph (G_{user}). Assume the total number of clusters is G. We denote the rating of POI v_j by the users in g-th cluster as $\mathbf{R}_{(g),j}$. The definition of local regularizer $\mathcal{J}(\mathbf{R})$ is as follows,

$$\mathcal{J}(\mathbf{R}) = \sum_{g=1}^{G}\sum_{j=1}^{n}\omega_g\|\mathbf{R}_{(g),j}\|_2, \quad (5)$$

where $\omega_g = \sqrt{n_g}$ and n_g is the number of users in cluster g. This regularizer is a *group lasso regularizer*, which was proposed in [Yuan and Lin, 2006] and has very wide applications [Jenatton *et al.*, 2010; Kim and Xing, 2010; Kolar *et al.*, 2009]. Following the same argument in [Liu *et al.*, 2014b], the local regularizer in Eq. (5) encourages the ratings of users belonging to a same group have similar sparsity patterns on ratings if it is used in the objective function.

Then, we can formulate the final objective function as,

$$\mathcal{L}(\mathbf{R}) = \frac{1}{2}\Big[\lambda_1\mathcal{L}_{user}(\mathbf{R}) + \lambda_2\mathcal{L}_{poi}(\mathbf{R}) + \lambda_3\|\mathbf{R} - \mathbf{Y}\|_F^2 + \lambda_4\mathcal{J}(\mathbf{R})\Big], \quad (6)$$

where $\mathcal{L}_{user}(\mathbf{R})$, $\mathcal{L}_{poi}(\mathbf{R})$ and $\mathcal{J}(\mathbf{R})$ are defined in Eq. (3), Eq. (4) and Eq. (5) respectively. We have added a new term $\|\mathbf{R} - \mathbf{Y}\|_F^2$ to encourage the rating matrix $\mathbf{R}$ to be as close as possible to the ground truth rating matrix $\mathbf{Y}$ of the training set (defined in Eq. (3)). The different terms are weighted by tunable parameters λ_1, λ_2, λ_3 and λ_4. We can then get the optimal rating matrix as $\mathbf{R}^* = \arg\min_{\mathbf{R}}\mathcal{L}(\mathbf{R})$. Let $V_u = \{j \in \mathbb{N}^+ | 1 \le j \le n\}\backslash P_u$ (P_u is defined in Section 3.1 and "$\backslash$" is the set difference operator) and $\mathcal{I}_k^{(u)} \in V_u$ be such that $\mathcal{I}_k^{(u)}$ achieves kth greatest rating in terms of rating score $(\mathbf{R}^*)_{u,\mathcal{I}_k^{(u)}}$. If we need to recommend K POIs to user u, the set of top K recommendations of POI indices will be $\hat{P}_u^{(K)} = \{\mathcal{I}_1^{(u)}, \mathcal{I}_2^{(u)}, \ldots, \mathcal{I}_K^{(u)}\}$.

4.3 Similarity Graph Construction

We now introduce how G_{user} and G_{poi} defined in Section 4.1 are constructed. In the experiment datasets, *i.e.*, Gowalla and Yelp (see Section 5.1 for detailed description), the following information are available: (1) the check-in transactions of each user, (2) the check-in time of each user at each POI, (3) the social relations between users, (4) the geographic coordinates of the POIs. Then, we use the above contextual information to construct the user similarity graph as well as the POI similarity graph.

User Similarity Graph Construction

The user similarity graph G_{user} is built based on the following assumptions: (1) Users that visit similar set of POIs at similar periods of a day are more similar (spatial-temporal similarity); (2) The users having social relationships are more similar (social relation similarity). We define a feature vector $\mathbf{f}_i^{\text{chkin-u}} \in \mathbb{R}^{24\times n}$ whose entries records the number of check-ins of user u_i at each POI in each hour of a day (because the number of POIs is n and there are 24 hours in a day, the dimension of $\mathbf{f}_i^{\text{chkin-u}}$ is $24 \times n$). The spatial-temporal similarity between two users, u_i and u_j (S_{ij}^{ST}), is defined as follows,

$$S_{ij}^{ST} = \frac{\mathbf{f}_i^{\text{chkin-u}} \cdot \mathbf{f}_j^{\text{chkin-u}}}{\|\mathbf{f}_i^{\text{chkin-u}}\|_2 \|\mathbf{f}_j^{\text{chkin-u}}\|_2}. \tag{7}$$

We denote the spatial-temporal similarity matrix as $\mathbf{S}^{ST} = \left[S_{ij}^{ST}\right]_{i,j=1\ldots m}$. To incorporate the social relation similarity, we compute the user similarity matrix as,

$$\mathbf{S}_{user} = \frac{(\alpha_{user} + \mathbf{S}^{SR}) \odot \mathbf{S}^{ST}}{1 + \alpha_{user}} \tag{8}$$

where $\odot$ denotes the Hadamard product, α_{user} is a smoothing factor and $\mathbf{S}^{SR}$ is a symmetric matrix that records the social relations between users. $S_{ij}^{SR} = 1$ if u_i and u_j have a social relationship, and $S_{ij}^{SR} = 0$ if they do not. Finally, we obtain G_{user} as the k-Nearest Neighbor (k-NN) graph of $\mathbf{S}_{user}$.

POI Similarity Graph Construction

The POI similarity graph G_{poi} is based on the following assumptions: (1) Geographically proximal POIs are more similar to each other (geographic similarity); (2) Similar POIs are visited in similar periods of a day (temporal similarity). We define a feature vector $\mathbf{f}_i^{\text{chkin-p}} \in \mathbb{R}^{24}$ that records the number of times a POI v_i is visited by *all users* in each hour of a day. The temporal similarity between two POIs, v_i and v_j (S_{ij}^{TE}), is defined as follows,

$$S_{ij}^{TE} = \frac{\mathbf{f}_i^{\text{chkin-p}} \cdot \mathbf{f}_j^{\text{chkin-p}}}{\|\mathbf{f}_i^{\text{chkin-p}}\|_2 \|\mathbf{f}_j^{\text{chkin-p}}\|_2}. \tag{9}$$

Then, we denote the temporal similarity matrix for POIs by $\mathbf{S}^{TE} = \left[S_{ij}^{TE}\right]_{i,j=1\ldots n}$. To take into account geographical similarity by proximity, we define the geographical similarity between two POIs v_i and v_j through the Gaussian kernel, $S_{ij}^{GE} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are geographic coordinates of POI v_i and POI v_j respectively. σ is the standard deviation parameter. We subsequently define the geographical similarity matrix $\mathbf{S}^{GE} = \left[S_{ij}^{GE}\right]_{i,j=1\ldots n}$. To put together the temporal and geographical similarity matrices, we define the POI similarity matrix as,

$$\mathbf{S}_{poi} = \frac{(\alpha_{poi} + \mathbf{S}^{TE}) \odot \mathbf{S}^{GE}}{1 + \alpha_{poi}} \tag{10}$$

where α_{poi} is a smoothing factor. Finally, we obtain G_{poi} as the k-NN graph of $\mathbf{S}_{poi}$.

Algorithm 1 Alternating Optimization

```
 1: Input
 2:     $\mathbf{Y}$: Initial rating matrix
 3:     $\mathbf{L}_{user}$: Normalized user Laplacian matrix
 4:     $\mathbf{L}_{poi}$: Normalized POI Laplacian matrix
 5:     $\lambda_1, \lambda_2, \lambda_3, \lambda_4$: Weighting parameters
 6: Output
 7:     $\mathbf{R}^*$: An optimized rating matrix
 8: Initialize $\mathbf{T}^{[2]} \leftarrow \mathbf{Y}$
 9: do
10:     (User Step) $\mathbf{T}^{[1]} \leftarrow \arg\min_{\mathbf{R}} \mathcal{L}_1(\mathbf{R}, \mathbf{T}^{[2]})$
            by APG with Eq. (15), Eq. (16) and Eq. (17)
11:     (POI Step) $\mathbf{T}^{[2]} \leftarrow \arg\min_{\mathbf{R}} \mathcal{L}_2(\mathbf{R}, \mathbf{T}^{[1]})$
            by solving Eq. (19)
12: until convergence
13: $\mathbf{R}^* \leftarrow \mathbf{T}^{[1]}$
```

4.4 Optimization

The optimization problem in Eq. (6) can be solved by an alternating algorithm, which includes the following two steps in each iteration. The first step is the *user step*, where user-based terms ($\mathcal{L}_{user}(\mathbf{R})$ and $\mathcal{J}(\mathbf{R})$) in Eq. (6) are optimized. The second step is the *POI step*, where POI-based terms ($\mathcal{L}_{poi}(\mathbf{R})$) in Eq. (6) is optimized. Concretely, we define two functions $\mathcal{L}_1(\mathbf{R}, \mathbf{T})$ and $\mathcal{L}_2(\mathbf{R}, \mathbf{T})$, each consists of the user-based terms and POI-based terms in Eq. (6),

$$\mathcal{L}_1(\mathbf{R}, \mathbf{T}) = \frac{1}{2}\left[\lambda_1 \mathcal{L}_{user}(\mathbf{R}) + \lambda_3 \|\mathbf{R} - \mathbf{T}\|_F^2 + \lambda_4 \mathcal{J}(\mathbf{R})\right], \tag{11}$$

$$\mathcal{L}_2(\mathbf{R}, \mathbf{T}) = \frac{1}{2}\left[\lambda_2 \mathcal{L}_{poi}(\mathbf{R}) + \lambda_3 \|\mathbf{R} - \mathbf{T}\|_F^2\right], \tag{12}$$

where $\mathbf{T}$ is a temporary placeholder matrix variable that represents some intermediate estimation of $\mathbf{R}^*$. The optimization procedure is summarized in Algorithm 1. We use two temporary variables $\mathbf{T}^{[1]}$ and $\mathbf{T}^{[2]}$ that keep track of the intermediate values of $\mathbf{R}^*$ after the user step and the POI step on each iteration. The temporary variable $\mathbf{T}^{[2]}$ is initialized to $\mathbf{Y}$ in the beginning (line 8). The term $\lambda_3\|\mathbf{R} - \mathbf{T}\|_F^2$ in Eq. (11) and Eq. (12) constrains the optimization result at current step to be as close as possible to the previous step. At convergence, the two matrices $\mathbf{T}^{[1]}$ and $\mathbf{T}^{[2]}$ are in general different, although each of them has a stable value. $\mathbf{T}^{[1]}$ can be regarded as the rating matrix after the user step (line 10) and $\mathbf{T}^{[2]}$ as the one after the POI step (line 11). We also observe in practice that $\mathbf{T}^{[1]}$ generally has better performance than $\mathbf{T}^{[2]}$. Therefore, when the iterations complete, we set the optimized rating matrix $\mathbf{R}^*$ to $\mathbf{T}^{[1]}$ in line 13.

We now describe how the user step and POI step optimizations are carried out. We look at the user step first. Note that the optimization objective of the user step can be separated into a smooth part ($\mathcal{L}_{1,sm}$) and a non-smooth part ($\mathcal{L}_{1,non}$), with $\mathcal{L}_1(\mathbf{R}, \mathbf{T}) = \mathcal{L}_{1,sm}(\mathbf{R}, \mathbf{T}) + \mathcal{L}_{1,non}(\mathbf{R})$ where,

$$\mathcal{L}_{1,sm}(\mathbf{R}, \mathbf{T}) = \frac{1}{2}\left[\lambda_1 \operatorname{trace}(\mathbf{R}^\top \mathbf{L}_{user} \mathbf{R}) + \lambda_3 \|\mathbf{R} - \mathbf{T}\|_F^2\right], \tag{13}$$

and

$$\mathcal{L}_{1,non}(\mathbf{R}) = \frac{\lambda_4}{2} \sum_{g=1}^{G} \sum_{j=1}^{n} \omega_g \|\mathbf{R}_{(g),j}\|_2. \tag{14}$$

Both $\mathcal{L}_{1,sm}(\mathbf{R}, \mathbf{T})$ and $\mathcal{L}_{1,non}(\mathbf{R})$ are convex, the optimization problem in the user step can be solved by the Accelerated Proximal Gradient (APG) algorithm [Toh and Yun, 2010].

To fit the objective function into the APG framework, we will need to derive the following auxiliary functions. First, the gradient of $\mathcal{L}_{1,sm}(\mathbf{R}, \mathbf{T})$ with respect to variable $\mathbf{R}$ is

$$\nabla_R \mathcal{L}_{1,sm}(\mathbf{R}, \mathbf{T}) = \lambda_1 L_{user} \mathbf{R} + \lambda_3 (\mathbf{R} - \mathbf{T}). \tag{15}$$

Then, the quadratic approximation of $\mathcal{L}_1(\mathbf{M}, \mathbf{T})$ at a specific point $\mathbf{R}$ (center of Taylor expansion) is

$$\begin{aligned} \mathcal{L}_1(\mathbf{M}, \mathbf{T}) &\approx \frac{\tau}{2} \|\mathbf{M} - \mathbf{Z}\|_F^2 + \mathcal{L}_{1,non}(\mathbf{M}) + \mathcal{L}_{1,sm}(\mathbf{R}, \mathbf{T}) \\ &- \frac{1}{2\tau} \|\nabla_R \mathcal{L}_{1,sm}(\mathbf{R}, \mathbf{T})\|_F^2 \overset{\text{def}}{=} Q_\tau(\mathbf{M}, \mathbf{R}), \end{aligned} \tag{16}$$

where $\mathbf{M}$ is in the neighborhood of $\mathbf{R}$, τ is the inverse of gradient descent step size and $\mathbf{Z} = \mathbf{R} - \tau^{-1} \nabla_{\mathbf{R}} \mathcal{L}_{1,sm}(\mathbf{R}, \mathbf{T})$. The gradient projection function $S_\tau(\mathbf{Z})$ is defined as $S_\tau(\mathbf{Z}) = \arg\min_{\mathbf{M}} Q_\tau(\mathbf{M}, \mathbf{R}) = \arg\min_{\mathbf{M}} \frac{\tau}{2} \|\mathbf{M} - \mathbf{Z}\|_F^2 + \mathcal{L}_{1,non}(\mathbf{M})$. A closed form solution of $S_\tau(\mathbf{Z})$ is obtainable, following the derivation in [Liu *et al.*, 2014b]. Let $\mathbf{M}^* = S_\tau(\mathbf{Z})$, then

$$\mathbf{M}^*_{(g),j} = \begin{cases} \frac{\mathbf{Z}_{(g),j} \left(\|\mathbf{Z}_{(g),j}\|_2 - \frac{\lambda_4 \omega_g}{\tau} \right)}{\|\mathbf{Z}_{(g),j}\|_2} & \text{if } \|\mathbf{Z}_{(g),j}\| > \frac{\lambda_4 \omega_g}{\tau}, \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{17}$$

In this way, the objective function in the user step can be optimized by Algorithm 1 in [Toh and Yun, 2010].

Next, we discuss the optimization in the POI step. In contrast to $\mathcal{L}_1(\mathbf{R}, \mathbf{T})$, $\mathcal{L}_2(\mathbf{R}, \mathbf{T})$ is completely smooth and convex, which can be easily optimized after setting the gradient of Eq. (12) to zero:

$$\nabla \mathcal{L}_2(\mathbf{R}, \mathbf{T}) = \lambda_2 \mathbf{R} L_{poi} + \lambda_3 (\mathbf{R} - \mathbf{T}) = \mathbf{0}, \tag{18}$$

which is equivalent to solving the linear system of $\mathbf{R}$,

$$\mathbf{R}(\lambda_2 L_{poi} + \lambda_3 \mathbf{I}) = \lambda_3 \mathbf{T}. \tag{19}$$

5 Experiments

5.1 Experimental Settings

Datasets

We use two datasets (*Gowalla* and *Yelp*) to evaluate the POI recommendation methods. The *Gowalla* dataset[1] contains user check-in data from February 2009 till October 2010 on the *Gowalla* social network. The dataset contains 18,737 users and 32,510 POIs. The total number of check-ins is 1,278,274. The *Yelp* dataset[2] contains 860,888 check-ins of 30,887 users and 18,995 POIs. They are from round 7 of the *Yelp dataset challenge*. For both datasets, additional contextual information, including social relationships between users and geographic coordinates of POIs, are available.

Method	P@5	P@10	P@20	P@50	R@5	R@10	R@20	R@50
USG	0.0605	0.0490	0.0384	0.0274	0.0431	0.0685	0.1059	0.1828
iGSLR	0.0288	0.0250	0.0197	0.0139	0.0190	0.0317	0.0488	0.0784
LORE	0.0409	0.0323	0.0242	0.0152	0.0262	0.0397	0.0579	0.0844
LFBCA	0.0602	0.0494	0.0393	0.0277	0.0426	0.0688	0.1075	0.1835
IRenMF	0.0662	0.0538	0.0429	0.0303	0.0462	0.0742	0.1163	0.1991
GeoMF	0.0644	0.0520	0.0415	0.0292	0.0454	0.0719	0.1121	0.1906
MGMPFM	0.0249	0.0203	0.0170	0.0132	0.0187	0.0293	0.0475	0.0901
GeoPFM	0.0386	0.0314	0.0256	0.0186	0.0262	0.0426	0.0659	0.1163
Caser	0.0378	0.0294	0.0224	0.0159	0.0254	0.0392	0.0592	0.1022
Ours	**0.0722**	**0.0586**	**0.0462**	**0.0322**	**0.0491**	**0.0789**	**0.1224**	**0.2073**

Table 1: Performances of different methods on Gowalla dataset.

Setup and Metrics

The two experimental datasets are partitioned into training set, tuning set and test set. The check-ins of each user is splitted with ratio 70%, 20% and 10% for training, tuning and testing, respectively. The performances of POI recommendation methods are evaluated by two metrics: Precision at K ($P@K$) and Recall at K ($R@K$). Empirically, we set K to 5, 10, 20, and 50.

Baseline Methods

We compare the proposed method with the following baseline methods: (1) **USG** [Ye *et al.*, 2011]: This method considers geographical influence by modeling the probability of a user visiting a new place given the user's historical check-ins; (2) **MGMPFM** [Cheng *et al.*, 2012]: This method uses PFM to model a user's preference for a POI, and a Multi-center Gaussian Model to estimate the probability of users to visit a POI given their living region; (3) **iGSLR** [Zhang and Chow, 2013]: This method computes similarity between users by measuring their friendship and distance between their residences. The probability of visiting a new place is estimated by Kernel Density Estimation (KDE); (4) **LFBCA** [Wang *et al.*, 2013]: This method builds a graph to model the users' relations. User preference, social influence, and user check-in similarity are represented by different edges; (5) **LORE** [Zhang *et al.*, 2014]: This method also models the probability of visiting a new POI by KDE. Moreover, it considers a sequential influence between the visited POIs and a given new POI; (6) **IRenMF** [Liu *et al.*, 2014b]: This method predicts the recommendation score based on both latent factors of a specific user-POI pair, and on neighboring POIs weighted by their similarity; (7) **GeoMF** [Lian *et al.*, 2014]: This method learns four kinds of latent factors for recommendation: user latent factors, POI latent factors, user activity areas latent factors and POI influence latent factors; (8) **GeoPFM** [Liu *et al.*, 2014a]: This approach models both geographical preference and user interest preference through PFM. It is able to handle skewed user check-in data; (9) **Caser** [Tang and Wang, 2018]: This method applies deep learning method on recommendation with convolutional networks.

Parameter Settings

The parameters in our model are sampled as follows: $\lambda_1 \in [0.05, 10]$, $\lambda_2 \in [0.05, 10]$, $\lambda_4 \in [10^{-6}, 10^{-1}]$, the user smoothing factor $\alpha_{user} \in [0, 1]$, the POI smoothing factor $\alpha_{poi} \in [0, 1]$, Gaussian kernel standard deviation $\sigma \in [0.1, 10]$ and the number of clusters for spectral clustering $G \in \{20, 50, 100\}$.

[1] http://snap.stanford.edu/data/loc-gowalla.html

[2] https://www.yelp.com/dataset/challenge

Method	P@5	P@10	P@20	P@50	R@5	R@10	R@20	R@50
USG	0.0254	0.0221	0.0186	0.0141	0.0262	0.0447	0.0734	0.1355
iGSLR	0.0127	0.0108	0.0089	0.0067	0.0100	0.0181	0.0285	0.0494
LORE	0.0240	0.0198	0.0158	0.0107	0.0189	0.0306	0.0458	0.0730
LFBCA	0.0232	0.0198	0.0173	0.0135	0.0236	0.0396	0.0677	0.1291
IRenMF	0.0286	0.0249	0.0209	0.0158	0.0291	0.0500	0.0819	0.1502
GeoMF	0.0309	0.0262	0.0215	0.0161	0.0315	0.0526	0.0851	0.1555
MGMPFM	0.0162	0.0134	0.0109	0.0086	0.0162	0.0275	0.0451	0.0868
GeoPFM	0.0222	0.0190	0.0158	0.0118	0.0219	0.0357	0.0606	0.1125
Caser	0.0178	0.0164	0.0145	0.0117	0.0171	0.0314	0.0553	0.1102
Ours	**0.0315**	**0.0271**	**0.0224**	**0.0167**	**0.0319**	**0.0542**	**0.0878**	**0.1597**

Table 2: Performances of different methods on Yelp dataset.

5.2 Performance Comparison

The performance on Gowalla and Yelp are summarized in Table 1 and Table 2. We have following observations:

- MF methods and our methods generally perform best in terms of precision and recall when $K = 5, 10, 20$ and 50.
- The PFM-based methods and two UCF methods, iGSLR and LORE, is inferior to other methods for most of the metrics.
- It seems that the high-perforance methods tend to integrate multiple levels of information. For example, the two MF-based methods, IRenMF and GeoMF, use latent factors for recommendation.
- For our method, without any probabilistic assumption (unlike PFM-based methods), we explicitly integrate multiple levels of information in the user-based and POI-based Laplacian regularization, which should be the reason that it achieves high performance among all methods.

5.3 Parameter Analysis

We study the effect of key parameters (λ_1,λ_2,λ_3 and λ_4) in the objective function which control the relative importance of each regularization term. Because if we change the four parameters proportionally, there will be no effect on the optimization result. Therefore, for simplicity, we set $\lambda_3 = 1$ and only change λ_1,λ_2 and λ_4.

We show the effect of changing one parameter at a time, while keeping the rest of the parameters at their optimal values. The intervals of the three parameters that we have tested are $[0.05, 10]$, $[0.05, 1.5]$ and $[10^{-6}, 10^{-1}]$, respectively. We show the two metrics P@10 and R@10 on both datasets.

The results are shown in Figure 1. We can see that the optimal values of the two parameters adjusting the user-based Laplacian regularization (λ_1) and POI-based Laplacian regularization (λ_2) tend to be of the same order with λ_3 (which is set to 1). A value that is too high or too low can both cause performance degradation. The optimal λ_1 is generally higher than λ_2 on both datasets, suggesting that user-based Laplacian regularization is more informative than its POI counterpart. This might suggest that user-based contextual information is more important than POI-based contextual information. The parameter for local regularizer (λ_4) affects the performance of the model only slightly when it is small, but will rapidly decrease performance when it is too large. Same as the observation in [Liu *et al.*, 2014b], a too-strong local regularization can introduce too much sparsity in the optimization result, seriously affecting recommendation. The local regularizer has a much smaller performance gain compared with the previous two regularizers, probably because a lot of contexual information has already been exploited in the two global regularizers.

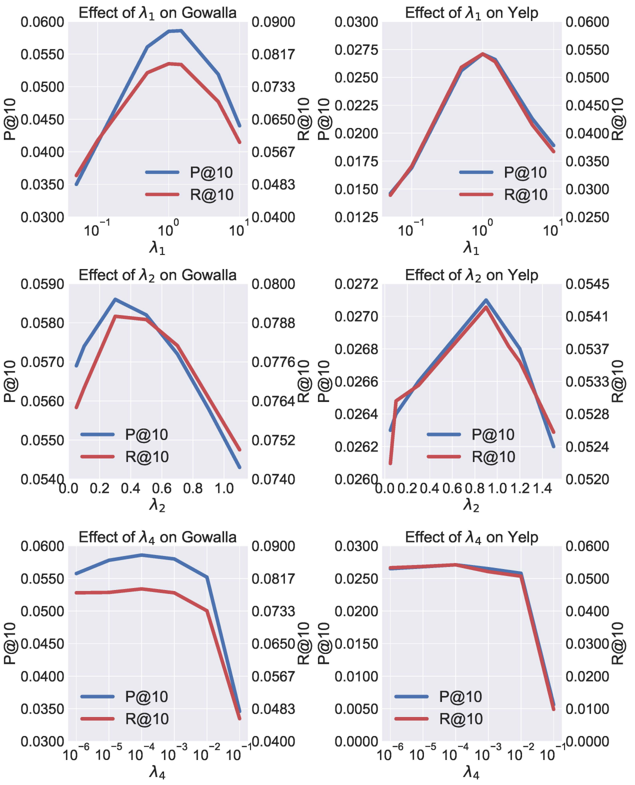

Figure 1: Performance trends of proposed method with different parameter settings.

6 Conclusion

In this paper, we designed a new framework for POI recommendation, which explicitly exploits global contextual information of users and POIs through Laplacian regularization and local contextual information through a local regularizer. We integrated various levels of similarity information into the similarity graph which the Laplacian regularizations and the local regularizer are based on. We described the formulations of objective functions of our model and how they are optimized. We showed by experiments that our method outperforms the existing methods on the two datasets, which shows the importance of constructing similarity by information integration in POI recommendation. The simplicity and extensibility of our framework can shed light on future research direction for POI recommendation and possibly methods for other recommendation tasks with contextual information.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61932004).

References

[Cheng *et al.*, 2012] Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[Gao *et al.*, 2013] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 93–100, 2013.

[Jenatton *et al.*, 2010] Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Structured sparse principal component analysis. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 366–373, 2010.

[Kim and Xing, 2010] Seyoung Kim and Eric P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning*, pages 543–550, 2010.

[Kolar *et al.*, 2009] Mladen Kolar, Le Song, and Eric P. Xing. Sparsistent learning of varying-coefficient models with structural changes. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009.*, pages 1006–1014, 2009.

[Li *et al.*, 2015] Xutao Li, Gao Cong, Xiao-Li Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *SIGIR*, pages 433–442, 2015.

[Lian *et al.*, 2014] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *KDD*, pages 831–840, 2014.

[Liu *et al.*, 2014a] Bin Liu, Hui Xiong, Spiros Papadimitriou, Yanjie Fu, and Zijun Yao. A general geographical probabilistic factor model for point of interest recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1167–1179, 2014.

[Liu *et al.*, 2014b] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 739–748, 2014.

[Liu *et al.*, 2017] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment*, 10(10):1010–1021, 2017.

[Liu *et al.*, 2018] Yong Liu, Lifan Zhao, Guimei Liu, Xinyan Lu, Peng Gao, Xiao-Li Li, and Zhihui Jin. Dynamic bayesian logistic matrix factorization for recommendation with implicit feedback. In *IJCAI*, pages 3463–3469, 2018.

[Tang and Wang, 2018] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018*, pages 565–573, 2018.

[Toh and Yun, 2010] Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of optimization*, 6(615-640):15, 2010.

[Von Luxburg, 2007] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[Wang *et al.*, 2013] Hao Wang, Manolis Terrovitis, and Nikos Mamoulis. Location recommendation in location-based social networks using user check-in data. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 374–383, 2013.

[Wu *et al.*, 2019] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. Pd-gan: adversarial learning for personalized diversity-promoting recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3870–3876. AAAI Press, 2019.

[Yang *et al.*, 2018] Peng Yang, Peilin Zhao, Yong Liu, and Xin Gao. Robust cost-sensitive learning for recommendation with implicit feedback. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 621–629. SIAM, 2018.

[Ye *et al.*, 2011] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun 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 Retrieval*, pages 325–334, 2011.

[Yuan and Lin, 2006] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[Zhang and Chow, 2013] Jia-Dong Zhang and Chi-Yin Chow. igslr: personalized geo-social location recommendation: a kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 334–343, 2013.

[Zhang *et al.*, 2014] Jia-Dong Zhang, Chi-Yin Chow, and Yanhua Li. Lore: Exploiting sequential influence for location recommendations. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 103–112, 2014.

[Zheng and Xie, 2011] Yu Zheng and Xing Xie. Location-based social networks: Locations. In *Computing with spatial trajectories*, pages 277–308. Springer, 2011.

[Zheng, 2012] Yu Zheng. Tutorial on location-based social networks. In *Proceedings of the 21st international conference on World wide web, WWW*, volume 12, 2012.