Point-of-Interest Recommendation With Global and Local Context

Peng Han, Shuo Shang, Aixin Sun, Peilin Zhao, Kai Zheng, Member, IEEE, and Xiangliang Zhang

Abstract—The task of point of interest (POI) recommendation aims to recommend unvisited places to users based on their check-in history. A major challenge in POI recommendation is data sparsity, because a user typically visits only a very small number of POIs among all available POIs. In this paper, we propose AUC-MF to address the POI recommendation problem by maximizing Area Under the ROC curve (AUC). AUC has been widely used for measuring classification performance with imbalanced data distributions. To optimize AUC, we transform the recommendation task to a classification problem, where the visited locations are positive examples and the unvisited are negative ones. We define a new lambda for AUC to utilize the LambdaMF model, which combines the lambda-based method and matrix factorization model in collaborative filtering. Many studies have shown that geographic information plays an important role in POI recommendation. In this study, we focus on two levels geographic information: local similarity and global similarity. We further show that AUC-MF can be easily extended to incorporate geographical contextual information for POI recommendation. Specifically, we propose two novel methods to incorporate geographical information in AUC-MF. Different from most existing models where the contextual information are incorporated into the objective function, the incorporation of contextual information in AUC-MF is a refinement of the model and a sampling strategy. The sampling strategy could speedup convergence and the refining of recommendations is independent of training of the model. This mechanism also enables AUC-MF to be able produce recommendations refined towards different contextual information, with minimum computational cost. Experiments on two datasets show that the proposed AUC-MF outperforms state-of-the-art methods significantly in terms of recommendation accuracy.

Index Terms—POI recommendation, AUC, matrix factorization, context

1 INTRODUCTION

LOCATION-BASED social networks (LSBNs) allow users to check in and share their experiences when they visit a point of interest (POI), such as a museum or a restaurant. With the development and popularity of various LSBN (Fig. 1) platforms e.g., BrightKite, Foursquare, and Gowalla, user check-in data is growing at an unprecedented pace. For instance, Foursquare had more than 50 million active users and more than 8 billion check-ins made by 2016.

The availability of abundant amount of user check-in data, enables many studies on recommender systems to further enhance user experiences. Examples include location recommendation [1], [2], [3], friend recommendation [4] and activity recommendation [5]. Among these applications, POI recommendations [6], [7], [8], [9], [10] gained extensive research attention in the past few years, and many algorithms have been proposed.

POI recommendation aims at finding unvisited locations that a user may be interested in, by learning from users check-in history and other related factors. POI recommendation is challenging for many reasons. One of the most important reasons is that user check-in data is extremely sparse. That is, the number of POIs visited by a user is typically a very small portion of all available POIs. To be reported shortly, in one of the datasets used in our experiments, on average a user visits only 15.9 POIs among all 46,617 available POIs in a city. This makes POI recommendation suffering from much worse data scarcity problem compared with many other recommendation tasks. For instance, the density of Netflix data for movie recommendation is 1.2 percent [11], which is much higher than that of POI check-in data. To make the problem even more challenging, check-in is implicit feedback [12], [13]. In other words, check-ins only provide the positive examples and the unlabeled POIs could be either negative or undiscovered positive.

In most existing studies, the main component of the objective function is Frobenius norm of the difference between check-in matrix and the recommendations made by a model [12], [14]. However, Frobenius norm is not specifically designed to handle the sparse problem, and cannot perform well when the data is extremely sparse, without constructing a complex model.

In this paper, we propose to maximize AUC to handle data sparsity in POI recommendation. AUC is a metric that is widely used for measuring classification accuracy for imbalanced data distributions. To the best of our knowledge, we are the first to use AUC as the objective function
for POI recommendation. To optimize AUC, we transform the recommendation task to a classification problem, where the visited locations are positive examples and the unvisited are negative ones. The sparsity of the check-in data now becomes data distribution imbalance which could be handled by AUC optimization. However, AUC is a non-smooth function, and how to optimize AUC itself is a difficult problem. Previous work [15], [16] replaces the indicator function in AUC with a convex surrogate, e.g., hinge loss function. But this approach cannot be easily extended to other non-smooth functions. In this work, we utilize the framework of LambdaMF [17], which combines lambda-based method [18] and the highly-successful matrix factorization model in collaborative filtering. More specifically, we use AUC as the objective function and define a new lambda which could utilize the property of AUC in our new framework named AUC-MF. Experimental results show that AUC-MF achieves state-of-the-art accuracy.

In addition to the binary check-in data, various types of contextual information have shown their effectiveness in improving POI recommendation accuracy [19], [20], [21], e.g., geographical coordinates of POIs, timestamps of check-ins, relationships among users, etc. A POI recommender framework like AUC-MF shall be extendible, to enable effective exploitation of such contextual information. In this work, we focus on geographic context, which relies on the similarity between POIs. Different from existing studies [22] that define the similarity only by the longitude and latitude of POIs, we also consider temporal information of user activity on them. We illustrate this point using an example 1.

Example 1. If John always goes to a drinking bar in late night, a recommender without temporal context may suggest him the following POIs: a shopping mall or a bank that are close to the bar. However, shopping mall and bank may only open during the daytime. Although these places are very near the bar, John would not visit them because of the time mismatch. But he may check in to a nearby hotel after he drinks too much, and that is often visited by other people in the night. That is, considering temporal information could give a better similarity between POIs.

We utilize geographic information on two levels: local similarity and global similarity. The motivation of local similarity is the same as most existing works [22], [23]: user would like to visit the nearest POIs to the locations they visited before. However, the similarity between two POIs in our work is measured by multiple types of contextual information, instead of only considering the geographic location. Moreover, we do not incorporate local similarity into the objective function, and this could reduce the time during the multiple iterative optimization process. We use a linear combination to refine the initial recommendation list by adding a weighted sum of all nearby visited POIs to the unvisited ones. The weight is computed by the contextual similarity and user check-in history.

At global level, our motivation for POI recommendation is different from others. Past works [22], [24] make the latent vectors of items in the same region be similar. The motivation behind is that items in the same region should have similar visiting property. For POI recommendation, the region is usually obtained by clustering POIs based on spatial distance [22]. Thus, the POIs in the same region could vary significantly, as the number of POIs in a region could be large. Moreover, constraining latent vectors may not lead to the expected results, even if this motivation is sensible. In our work, the motivation for utilizing global context is to focus on the POIs that users dislike. That is, a user will infrequently go to POIs in a region she never visits before. To achieve this, we propose a region-based sampling method to select negative samples during our optimizing process. In this way, we directly lower the predicted scores of the POIs that are not in a user’s interest region, and this also avoids lowering the ranks of true samples. Further, we used the contextual similarity to cluster POIs into different regions, and this could make the POIs in the same region more similar in terms of visiting patterns. In our implementation, Laplician Eigenmaps [25], [26] is used to generate low dimension vectors of the similarity for clustering.

Experimental results show incorporating contextual information further improves the recommendation accuracy. Other contextual information could also be incorporated into our model in a similar manner by constructing the combined similarity. Note that, most of existing models utilize the contextual information in the objective function, which would make the training process time-consuming. AUC-MF is flexible in the sense that the recommendations can be further enhanced as a refinement and sampling process. Depending on the data availability, users may choose recommendations refined by different contextual factors. For instance, users with more social relationships with others may prefer recommendations refined based on social relationships while users with limited social relationships may get refined recommendation by other contextual factors.

We summarize our contribution in this paper as follows:

- We propose a new framework named AUC-MF for POI recommendation, which effectively addresses the data scarcity problem.
- We define a contextual similarity for POIs which could be used for clustering and finding the nearest samples.
- We utilize the geographic contextual information in two levels with a sampling method and refinement of final results.
- Extensive experiments on two datasets from Gowalla demonstrate that AUC-MF outperforms state-of-the-art methods significantly for POI recommendation.
The rest of the paper is organized as follows. Section 2 introduces related works. Section 3 gives the preliminary knowledge about POI recommendation. Section 4 presents our recommendation framework. Section 5 describes how to utilize contextual information. After reporting experiments in Section 6, we conclude this paper in Section 7.

2 RELATED WORK

2.1 Matrix Factorization
Matrix Factorization (MF) [27], the most popular algorithm in the field of recommendation, decomposes check-in matrix into user matrix and POI matrix. Many models for POI recommendation are based on this framework. We will review three MF-based models in this section. Two of them utilize geographical context and achieve the top two performance in the latest experimental evaluation of POI recommendation models [28]. The third one exploits temporal information by using different latent vectors for different time slots.

IRenMF [22] is based on Weighted Matrix Factorization (WMF) [29], [30]. This model utilizes geographic information under two motivations. With the intuition that users would visit the nearest POIs to the locations they visited before, IRenMF incorporates geographical neighborhood to a location. By considering that POIs in a region share user preference, they cluster POIs into different regions and make latent vectors of POIs in a similar region.

Rank-GeoFM [23] proposes a pair-wise non-smooth loss to reduce the influence of data sparsity problem. To handle the non-smooth problem, they use a continuous approximation to replace the original loss. Geographic information is utilized by constructing two latent vectors for every user. One is used to compute the user preference of the appointed POI, and the other is to compute the score of nearby neighborhoods. A sampling method is proposed to accelerate the updating process.

2.2 Non-Smooth Function
Non-smooth quality measures are particularly difficult to optimize directly, because the model scores are computed with the ranks of items. As the rank is only non-continuously changed in specific single point, we cannot get non-zero derivatives for the model parameters. In [18], they propose a general framework to optimize non-smooth function with stochastic gradient descent method. They construct a implicit convex function with virtual gradients under two conditions. To make the implicit function exist, the Jacobian matrix should be symmetric. To ensure the cost function to be convex, the Jacobian matrix should be positive semidefinite everywhere. This method could be applied in optimizing any non-smooth function.

LambdaMF [17] combines the Lambda method [18] with matrix factorization to solve the movie recommendation problem. Their objective function is to optimize the Normalized Discount Cumulative Gain (NDCG) metric. However, the objective measure in [17] does not satisfy the application of POI recommendation for two reasons. One reason is that the density of the check-in data in POI recommendation is much lower than the density of the data for movie recommendation. The other reason is that check-in is a type of implicit feedback, which offers only positive examples. Check-in does give explicit ratings (e.g., 1 to 5) as in movie recommendation. The unlabeled POIs are either negative or undiscovered positive. In our research, we use AUC as the objective function, which could handle the special properties of POI recommendation task.

2.3 Geographic Information Utilization
To measure the similarity between different trajectories, [31] extends the CTP query to include practical scenarios. Number of travelers that travel from each meeting point to the destination is considered to compute the connection travel cost, as well as the capacity of shuttle bus. In [32], [33], [34], they propose a new contextual similarity by combining the spatial and temporal distance.

Clustering POIs into different regions has been applied in many works. For example, [35] proposes a method named Collective Cluster Search to find the POI cluster that is closest to the query route. Spatial distance and density are considered to compute the distance between cluster and query route.

In our work, we compute the similarity between POIs and cluster them with new strategies.

2.4 Graph-Based Method
Graph-based embedding could be used for recommendation. In [36], they define the embeddings of POIs based on a graph-based metric which projects the Point-of-Interest into a low dimensional representation. The way they construct the embedding could catch user’s dynamic interests efficiently. Another way of utilizing graph to do recommendation is applying Graph Neural Networks (GNN). In [37], they use the visiting relationship as the graph for GNN to do recommendation.

2.5 Next-POI Recommendation
The task of next-POI recommendation [38], [39] is also studied in recent years, but the setting of it is different from that of POI recommendation. Instead of learning the general interests of every user, the target of next-POI recommendation is to learn the transition probabilities between POIs of every user. Given one user and the current POI or recently visited POIs, next-POI recommendation will provide the list of POIs that this user may visit next. The applied scenario of next-POI recommendation is also different from that of POI recommendation, where next-POI recommendation needs user’s location or recent check-ins as input.

3 PRELIMINARIES

3.1 Problem Definition
In this section, we first give concepts and their notations, after which we define the problem of POI recommendation. Table 1 illustrates primary notations frequently used in this paper.

Given the historical interactions between $m$ users and $n$ POIs, the task of POI recommendation is to recommend a target user $u$ a list of POIs. The POIs in the recommended list should have not been visited by the target user. In many real-world scenarios, POI recommendation tasks are based on user implicit preference feedback, i.e., whether a user has visited a POI. This kind of feedback is usually represented by using a set of binary variables $y_{ui} \in \{0,1\}$. If a user $u$ has visited a POI $i$, $y_{ui}$ is set to 1, and 0 otherwise. Note that
TABLE 1
Frequently Used Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m, n</td>
<td>Number of users m and number of POIs n</td>
</tr>
<tr>
<td>( U, L )</td>
<td>User set ( U ) and item set ( L )</td>
</tr>
<tr>
<td>( L^+_u, L^-_u, D )</td>
<td>Positive and negative POI sets ( L^+_u, L^-_u ) for user ( u ), and all positive pairs ( D )</td>
</tr>
<tr>
<td>( y_{ui}, s^+<em>i, \pi</em>{ui} )</td>
<td>Groundtruth ( y_{ui} ), predicted score ( s^+<em>i ) and predicted rank ( \pi</em>{ui} ) for POI ( i ) with respect to user ( u )</td>
</tr>
<tr>
<td>( C, \lambda )</td>
<td>Implicit function ( C ) and virtual gradient function ( \lambda )</td>
</tr>
<tr>
<td>( P, \pi )</td>
<td>User latent matrix ( P ) and POIs' i-th row vector ( \pi_i )</td>
</tr>
<tr>
<td>( Q, q )</td>
<td>POI latent matrix ( Q ) and POIs' s-th row vector ( q_s )</td>
</tr>
<tr>
<td>( w, \alpha )</td>
<td>Implicit parameter ( w ) to represent ( p ) and ( q ), and regularization parameter ( \alpha ) to constrain ( p ) and ( q )</td>
</tr>
<tr>
<td>( x_i )</td>
<td>POI ( i )'s coordinate ( x_i )</td>
</tr>
<tr>
<td>( F, T, A )</td>
<td>Visited hour histogram ( T^\text{vis} ) and auxiliary matrix ( A ) to compute the temporal feature ( F ) for POIs</td>
</tr>
<tr>
<td>( G, T, S^{TG} )</td>
<td>Spatial similarity matrix ( G ), temporal similarity matrix ( T ) and temporal-spatial similarity matrix ( S^{TG} )</td>
</tr>
<tr>
<td>( W, \ell, \Lambda )</td>
<td>( \ell ) near neighbour weight matrix ( W ) of ( S^{TG} ), Laplacian matrix ( \Lambda ), eigenvector ( \ell ) and eigenvalue ( \Lambda ) of ( W )</td>
</tr>
<tr>
<td>( P^{(j)}, n_y )</td>
<td>One POI region ( P^{(j)} ) and number of regions ( n_y )</td>
</tr>
<tr>
<td>( B_u )</td>
<td>Set of user ( u )'s unvisited regions</td>
</tr>
<tr>
<td>( \eta )</td>
<td>Learning rate</td>
</tr>
<tr>
<td>( N )</td>
<td>Local ( k ) near neighbour weight matrix</td>
</tr>
<tr>
<td>(</td>
<td>\cdot</td>
</tr>
<tr>
<td>( \otimes )</td>
<td>Hadamard product of two matrices</td>
</tr>
<tr>
<td>( \mathcal{I}(\cdot) )</td>
<td>Indicator function</td>
</tr>
<tr>
<td>( R )</td>
<td>Predicted user preference matrix</td>
</tr>
<tr>
<td>( \gamma, \mu )</td>
<td>Weights to control the influence of local context</td>
</tr>
</tbody>
</table>

\( y_{ui} = 0 \) does not explicitly indicate that \( u \) is not interested in \( i \). It may be the result that user \( u \) does not know the existence of \( i \). In this paper, we denote the set of users and POIs by \( U \) and \( L \), respectively. For a user \( u \), we denote her visited POIs by \( L^+_u = \{ i | y_{ui} = 1, i \in L \} \) and her unvisited POIs as \( L^-_u = L \setminus L^+_u \). Then, the set of all user-POI interactions is defined as \( D = \{ (u, i) | u \in U, i \in L \} \). For POI recommendation, we aim to recommend every user \( u \) a list of POIs from their unvisited POIs \( L^-_u \).

### 3.2 Overview of the Proposed Solution

To address the data sparsity of POI recommendation, we reformulate recommendation into an imbalanced classification problem. AUC optimization is then applied to solve this problem. To handle the non-smooth component in the AUC formula, Lambda-based method is used to optimize the objective function. To make the optimization be consistent with AUC, we define a new lambda with AUC definition. Combining with matrix factorization, our method for POI recommendation is named AUC-MF.

To utilize the geographic context, we define a temporal-spatial distance to measure the similarity between POIs. Then we incorporate the contextual information into our model from both global and local perspectives. For the global level context, we cluster POIs into different regions based on the proposed similarity. A sampling method is proposed to make sure that negative samples could only be gotten from inactive regions for every user. Local level context is used with the motivation that users prefer to visit nearby POIs of those visited before. Instead of incorporating the local contextual information into the objective function, we propose to refine the recommendation list with a linear combination.

### 4 AUC-MF for POI Recommendation

This section details the proposed AUC-MF, illustrated in Fig. 2. Given the check-in data, we generate positive and negative user-POI pairs from the check-in data. Then we use them as the input to AUC-MF. By optimizing AUC-MF with stochastic gradient descent (SGD) method, we get the user preference matrix.

#### 4.1 AUC Metric for POI Recommendation

AUC [40] is a decision threshold metric that the probability for a randomly drawn positive instance has a higher decision value than a randomly sampled negative instance. With the definition of AUC measure for binary classification [41], we define the AUC measure for POI recommendation as follows:

\[
\text{AUC} = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|L^+_u|} \sum_{i \in L^+_u} \frac{1}{|L^-_u|} \sum_{j \in L^-_u} \mathcal{I}(\pi_{ui} > \pi_{uj})
\]

\[
= \frac{1}{|U|} \sum_{u \in U} \frac{1}{|L^+_u|} \sum_{i \in L^+_u} \frac{1}{|L^-_u|} \sum_{j \in L^-_u} \mathcal{I}(\pi_{ui} \leq \pi_{uj})
\]

In the above equation, \(| \cdot |\) denotes the cardinality of the set. \( \pi_{ui} \) is the rank position of \( i \) in the ranked POI list, generated based on the user \( u \)'s preferences. The rank is based on the predicted scores \( (s^+_i, s^+_2, \ldots, s^+_n) \), in descending order. \( \mathcal{I}(\cdot) \) is an indicator function that outputs 1 if the condition holds and 0 otherwise. Thus maximizing AUC is equivalent to minimizing

\[
C = \frac{1}{|U|} \sum_{u \in U} \sum_{i \in L^+_u} \sum_{j \in L^-_u} \mathcal{I}(\pi_{ui} \leq \pi_{uj}) \frac{|L^+_u|}{|L^-_u|}.
\]

By this definition, we see that positive samples are computed the same number of times as negative ones. This decreases the impact of data sparsity in recommendation task. Moreover, the weight for a user depends on the
number of her positive samples, as the number of negative samples is much larger than the number of positive ones. The more her positive samples are, the less the weight is. This setting naturally handle the imbalance between different users. For example, some users may check in a large number of places. If we give the same weight to their check-in as others’, the recommender would be inclined to their preference.

For any \((u, i) \in \mathcal{D}\) and \(j \in L_u^i\), we define the cost function \(C_{ij}^u\) as follows:
\[
C_{ij}^u = \frac{\mathbb{I}(u_i \leq u_j)}{|L_u^i||L_u^j|}
\]

Thus,
\[
C = \sum_{u \in U} \sum_{i \in L_u^i} \sum_{j \in L_u^j} C_{ij}^u
\]

However, the indicator function \(\mathbb{I}(\cdot)\) is a non-smooth function, which means that the derivatives of the cost with respect to the model parameters are either zero or undefined. To apply SGD for optimization, we need to generate gradients for the cost function. Next, we will focus on the virtual derivatives of the cost \(C_{ij}^u\) with respect to the model parameter \(w\).

### 4.2 Rank With Non-Smooth Function

LambdaRank [18] provides a method that can be extended to any non-smooth and multivariate cost functions. It is based on the idea of RankNet, which formulates pair-wise ranking problem into gradient descent for a pair of documents. LambdaRank formulates gradient of a list-wise ranking measure, which is called \(\Lambda\). Taking the POI recommendation as an example, for user \(u\), the gradient of an implicit cost function \(C^u\) with respect to the score \(s_j^u\) of POI \(j\) is written as
\[
\frac{dC^u}{ds_j^u} = -\lambda_j^u(s^u_i, y_{u1}, \ldots, s^u_n, y_{un}).
\]

To make the implicit cost function exist and be convex, two conditions should be satisfied. First, the Jacobian matrix must be symmetric, which means
\[
\frac{\partial \lambda_k^u}{\partial s_j^u} = \frac{\partial \lambda_j^u}{\partial s_k^u} \quad \forall j, k \in \{1, \ldots, n\}.
\]

Second, the Jacobian matrix should be positive semidefinite everywhere. Note that for constant \(\lambda_s\), the above two conditions are trivially satisfied.

The authors reported that \(|\Delta NDCG|\) is the best \(\lambda\) used for optimizing NDCG from several possible \(\lambda\)s tested in [18]. It is defined as the absolute NDCG gain from swapping documents \(i\) and \(j\). Instead of showing the details of their formulation, we give the general form of the lambda method for POI recommendation with \((u, i) \in \mathcal{D}\) and \(j \in L_u^i\).
\[
\frac{dC_{ij}^u}{ds_i^u} = -\lambda_{i,j}^u = -\frac{dC_{ij}^u}{ds_j^u}.
\]

When \(\lambda_{i,j}^u\) is set as a positive constant, POI \(i\) must move up and POI \(j\) must move down in the ranked list to reduce the cost. Then the derivatives of the cost \(C_{ij}^u\) with respect to the model parameter \(w\) is
\[
\frac{dC_{ij}^u}{dw} = \frac{dC_{ij}^u}{ds_i^u} \frac{d{s_i^u}}{dw} + \frac{dC_{ij}^u}{ds_j^u} \frac{d{s_j^u}}{dw}
= -\lambda_{i,j}^u \frac{d{s_i^u}}{dw} + \lambda_{i,j}^u \frac{d{s_j^u}}{dw}.
\]

After we get the derivatives, we can apply them in the matrix factorization based methods to tackle the task of POI recommendation.

### 4.3 Lambda in Matrix Factorization

Matrix factorization based algorithms are the most popular and important recommendation algorithms [27]. Given \(m\) users and \(n\) POIs, there is a sparse matrix \(Y \in \mathbb{R}^{m \times n}\) to describe the training data, where \(i\)-th row represents visited POIs of user \(i\) and \(j\)-th column denotes who have visited POI \(j\). If user \(i\) has visited POI \(j\), then we set \(y_{ij} = 1\), otherwise, \(y_{ij} = 0\). The idea of MF is to factorize \(Y\) into two latent matrix \(P \in \mathbb{R}^{m \times d}\) and \(Q \in \mathbb{R}^{n \\times d}\) with a self-defined factor dimension \(d\). The \(i\)-th row latent vector \(p_i \in \mathbb{R}^{1 \times d}\) in \(P\) represents the user \(i\) while the \(j\)-th row latent vector \(q_j \in \mathbb{R}^{1 \times d}\) in \(Q\) represents the POI \(j\). The preference score of \(u\) to \(i\) is approximated by:
\[
s_i^u = p_u^T q_i^T.
\]

LambdaMF [17] presents a way to apply SGD to learn model parameters. We utilize its theory and redefine the gradients to optimize \(P\) and \(Q\). First, given any \((u, i) \in \mathcal{D}\) and \(j \in L_u^i\), in POI recommendation task, we should update the POI latent vector \(q_j\) and \(q_i\) and user latent vector \(p_u\). As the score \(s_i^u\) of the pair \((u, i)\) is generated from the inner-product of \(p_u\) and \(q_i\), we have
\[
\frac{ds_i^u}{dp_u} = q_i, \quad \frac{ds_i^u}{dq_i} = p_u.
\]

Thus the gradient can be computed as:
\[
\begin{align*}
\frac{dC_{ij}^u}{dp_u} &= \frac{dC_{ij}^u}{ds_i^u} \frac{d{s_i^u}}{dp_u} = -\lambda_{i,j}^u p_u, \\
\frac{dC_{ij}^u}{dq_i} &= \frac{dC_{ij}^u}{ds_i^u} \frac{d{s_i^u}}{dq_i} = \lambda_{i,j}^u p_u, \\
\frac{dC_{ij}^u}{dp_u} &= \frac{dC_{ij}^u}{ds_i^u} \frac{d{s_i^u}}{dp_u} + \frac{dC_{ij}^u}{ds_j^u} \frac{d{s_j^u}}{dp_u} = \lambda_{i,j}^u (q_j - q_i).
\end{align*}
\]

The definition of specific \(\lambda\) is the key in our algorithm. To make the algorithm effective and efficient, and to ensure that it satisfies the theory of LambdaRank [18], we define a simple and generic lambda for our framework. Before giving the explicit definition of \(\lambda\), we would first solve the over-flow problem of latent vector.

### 4.4 Regularization Term

The popular POIs that are visited by many users can cause a serious problem when using lambda-based method, similar to the problem caused by popular movies for movie recommendation [17]. Suppose that there exists a popular POI \(i\) that has been visited by all users, the score for item \(i\) would...
keep becoming higher during the training process. Because of \(-\lambda_i^u\), the derivatives of \(C_i^u\) with respect to the score of \(s_{ij}^u\), for any user \(u\) is negative: it moves the score \(s_{ij}^u\) up. Moreover, if the latent factors of all users are similar, the increasing of the predicted score of POI \(i\) for every user \(u\) would not cause the decrease of predicted score of POI \(i\) for other users who have visited POI \(i\). Hence, the latent factor of POI \(i\) would keep growing and soon cause overflow.

### 4.5 AUC Lambda

In the previous sections we have provided a general definition of \(\lambda\), and now we discuss how to choose \(\lambda\) explicitly. There are two conditions \(\lambda\) should satisfy, that would make the implicit cost function exist and be convex. First, the Jacobian matrix of the implicit cost function with respect to the rating score must be symmetric. This represents that there exists a cost function for which \(\lambda\) is derivative. Once satisfying the condition of existence, we should make sure that the implicit cost function is convex. That is, the Jacobian matrix should be positive semidefinite everywhere. As discussed before, the constant \(\lambda\) could satisfy both conditions. Considering minimizing the cost function, \(\lambda\) should be positive. Therefore, given any \((u, i) \in \mathcal{D}\) and \(j \in L_u\), we set

\[
\lambda_{ij}^u = |\Delta \text{AUC}| = \frac{|r_{ij}^u - r_{ij}^p| + 1}{|L_u^+||L_u^-|},
\]

where \(r_{ij}^u\) denotes the predicted rank of POI \(i\) for user \(u\) and \(\Delta \text{AUC}\) means the absolute AUC difference by swapping the two POIs. The whole process of AUC-MF is given in Algorithm 1.

### 5 Extend AUC-MF With Contextual Information

In this section, we first define temporal-spatial similarity. Then we will show how to utilize the global and local level geographic context under our framework.

#### 5.1 Contextual Similarity Matrix

Geographic information plays an important role in the task of POI recommendation. The key of utilizing geographic information is how to measure the similarity between POIs. Many works have shown that utilizing the similarity between POIs benefits recommendation accuracy. However most of them [22], [23] compute the similarity solely by the spatial information of POIs. Without considering temporal characteristics of POIs, the similarity may not truly reflect the visiting correlation between POIs. Because the activity hours between them may be different, even if these POIs are very close. In our work, we propose a temporal similarity based on the activity hours of POIs. The activity hours are computed by the user’s visited hours of the POIs. Moreover, we define the category similarity of POIs with their categories. Then we combine the spatial similarity, temporal similarity and category similarity to compute the contextual similarity.

The spatial similarity \(G \in \mathbb{R}^{n \times n}\) is defined by the geographic distance. To measure the spatial similarity \(G_{i,j}\) between two POIs \(i\) and \(j\), we use Gaussian distance:

\[
G_{i,j} = e^{-\frac{||x_i - x_j||^2}{\sigma^2}},
\]

where \(x_i\) is the vector of POI \(i\)’s coordinate, \(x_j\), latitude and longitude, and \(\sigma\) is a constant, empirically set to 0.1 in our experiments.

To calculate the temporal similarity between POIs, we first define the temporal feature for them. Specially, we partition a day into 24 time slots \((i.e., one hour per slot)\) and use a 24 dimension vector to represent each POI. The value for
each dimension is the relative frequency of the check-ins received by the POI. The check-in may be recorded at any time during a user’s stay period. Someone would check in a place when she just arrives to express the feeling of freshness. Another user may check in and comment on the visiting before leaving the place. Thus, to compute the similarity between two POIs, instead of only considering the check-ins received in the matching hours, we also consider the check-ins received in adjacent time slots.

More specifically, we use a matrix \( T_{h\mathsf{his}} \in \mathbb{R}^{n \times 24} \) to record the check-in frequencies of all the \( n \) POIs. An entry of \( T_{h\mathsf{his}} \) is set as follows:

\[
T_{ij}^{h\mathsf{his}} = \begin{cases} 
\frac{c_{ij}}{\sum_j c_{ij}} & \text{if } \sum_j c_{ij} > 0 \\
0 & \text{otherwise}
\end{cases}
\]

(16)

where \( c_{ij} \) denotes the number of check-ins POIs \( p_i \) received from all users within the \( j \)-th hour of a day, from the training set. To consider not only the check-ins made in matching hours, but also in the adjacent hours, we introduce an auxiliary matrix \( A \in \mathbb{R}^{24 \times 24} \):

\[
A_{ij} = \begin{cases} 
1 & \text{if } i = j \\
0.5 & \text{if } (j - i) \% 24 = 1 \text{ or } (j - i) \% 24 = 23 \\
0 & \text{otherwise}
\end{cases}
\]

(17)

In matrix \( A \), 0.5 is a weight assigned to the matching made with adjacent hours. Then the temporal features \( F \in \mathbb{R}^{n \times 24} \) for computing temporal similarity between POIs is calculated as

\[
F = \text{CNorm}(T_{h\mathsf{his}} A),
\]

(18)

where \( \text{CNorm}(\cdot) \) denotes a norm computation that every element in the matrix is divided by the \( L2 \) norm of its corresponding column vector. We now compute the temporal similarity matrix \( T \in \mathbb{R}^{n \times n} \) of all POIs as

\[
T = FF^T.
\]

(19)

The category is an important property of POI, which will indicate the similarity between POIs directly. And one POI may have multiple categories. Given the category matrix \( M \), where \( M_{i,c} = 1 \) means POI \( v_i \) has the \( c \)-th category, we could get the category similarity matrix \( Ca \) between all POI pairs as

\[
Ca = MM^T.
\]

(20)

Finally, we construct contextual similarity matrix \( S^C \in \mathbb{R}^{n \times n} \) by increasing the spatial similarity between POIs if they have temporal relationship or category similarity as follows:

\[
S^C = (1 + T) \otimes (1 + Ca) \otimes G,
\]

(21)

where \( \otimes \) denotes the Hadamard product of two matrices. Under this way, the geographic similarity could be refined by the temporal information and categories of POIs.

The contextual information of users could also improve the accuracy of POI recommendation, as the interests between friends may have connection. Given the social relationships between users, we could define the contextual similarity of users as \( S^U \), where \( S^U_{ij} = 1 \) if \( i \)-th user and \( j \)-th user have social connection.

5.2 Global Level Context

Past works [22], [24] make latent vectors of items in the same region be similar. The motivation behind this is that items in the same region should have similar properties. For POI recommendation, the POIs are usually clustered by the spatial distance. If the number of clusters is not large enough, POIs in the same region may vary significantly. On the other hand, a large number of clusters would lead to high computational cost. In our work, we utilize the global level geographic context from a different perspective. We consider that user would not visit the POIs that are in their inactive areas. Instead of constraining latent vectors of POIs to achieve our goal, we focus on controlling the predicted scores by a sampling method.

To make POIs in one region more similar, we cluster POIs based on our contextual similarity matrix. Instead of directly performing POI clustering by this similarity matrix, we utilize the laplacian eigenmaps [25] to generate low-dimension vectors of POIs for more efficient and effective clustering. To minimize noises in POI clustering, we filter the temporal-spatial similarity matrix \( S^P \) to obtain a weight matrix \( W \in \mathbb{R}^{n \times n} \) by considering only the \( k \) nearest temporal-geographical neighbours of every POI. More specifically, the weight matrix \( W \) is set as follows:

\[
W_{ij} = \begin{cases} 
S^P_{ij} & \text{if } p_i \in N^P(p_j) \text{ or } p_j \in N^P(p_i) \\
0 & \text{otherwise}
\end{cases}
\]

(22)

where \( N^P(p_i) \) is the set of \( k \) nearest neighbours of POI \( p_i \), and we set \( k = 50 \) here.

With weight matrix \( W \), we compute eigenvalues and eigenvectors for the generalized eigenvector problem.

\[
\mathcal{L}f = \Lambda Df,
\]

(23)

where \( D \) is a diagonal weight matrix and its entries are column sums of \( W \), i.e., \( D_{ii} = \sum_j W_{ij} \). \( \mathcal{L} = D - W \) is the Laplacian matrix. Laplacian is a symmetric, positive semidefinite matrix which can be thought of as an operator on functions defined on vertices of \( W \).

Let \( f_0, \ldots, f_{n-1} \) be the solutions of Equation 23, ordered by their eigenvalues,

\[
\begin{align*}
\mathcal{L}f_0 &= \Lambda_0 Df_0 \\
\mathcal{L}f_1 &= \Lambda_1 Df_1 \\
&\vdots \\
\mathcal{L}f_{n-1} &= \Lambda_{n-1} Df_{n-1}
\end{align*}
\]

(24)

We then select the the eigenvectors \( \{ f_{i_1}, f_{i_2}, \ldots, f_{i_s} \} \) whose eigenvalues are the smallest positive values, to get vector \( \vec{c}_i \).

\[
\vec{c}_i = ( f_{i_1}(i), f_{i_2}(i), \ldots, f_{i_s}(i) ),
\]

(25)

Here, \( \vec{c}_i \) is the vector representation of POI \( p_i \), and \( f_{i_j}(i) \) is the \( j \)-th element of eigenvector \( f_{i_j} \). The vector \( \vec{c}_i \) is used as \( p_i \)'s feature for the clustering method. In our method, k-means algorithm is applied on these features to group POIs into \( n_y \) temporal-geographical regions: \( \{ y_{(g)} \}_{g=1}^{n_y} \).
Once we get the clustered regions of POIs, we propose a region-based sample method to exploit the geographic context in global perspective. In our work, we split all POIs into visited and unvisited areas for every user with the temporal-geographical regions. Given user \( u \), we define her unvisited area as

\[
B_u^i = \{ j | j \in \mathcal{Y}_{(g)}, i \notin \mathcal{Y}_{(g)}; y_{ui} = 1 \},
\]

\( B_u^i \) denotes the set of POIs that in user \( u \)'s inactive regions.

### Algorithm 2. AUC-MF-G

**Input:** The observed interactions \( D \), unvisited regions \( \{ B_1^u, \ldots, B_n^u \} \), learning rate \( \eta \), regularization weight \( \alpha \), and number of iterations \( n_{iter} \).

**Output:** The learned user latent matrix \( P \) and POI latent matrix \( Q \).

1. Initialize \( P \) and \( Q \) randomly;
2. for \( t = 1 \) to \( n_{iter} \) do
3. for \( u = 1 \) to \( n \) do
4. \( \frac{\partial \mathcal{L}}{\partial \eta} = 0; \)
5. for \( i = 1 \) to \( m \) do
6. \( r = \text{argsort}(p_u * Q^\top) \)
7. for all \((u, i) \in \mathcal{D}\) do
8. Set \( T = 0 \)
9. while \( T < |B_u^i| \) do
10. Randomly draw an POI \( j \) from \( B_u^i \);
11. if \( r_j < r_t \) then
12. break
13. end if
14. \( T++ \)
15. end while
16. \( \lambda = \frac{\sum_{i=1}^{n} |B_u^i|}{|B_u^i|} \)
17. \( \frac{\partial \mathcal{L}}{\partial \eta} = \lambda (q_j - q_i) + \alpha p_u \)
18. \( \frac{\partial \mathcal{L}}{\partial \eta} = \lambda (q_j - q_i) + \alpha q_i \)
19. \( \frac{\partial \mathcal{L}}{\partial \eta} = \lambda q_j + \alpha q_j \)
20. end for
21. \( p_u = -\eta \frac{\partial \mathcal{L}}{\partial \eta} \)
22. for \( i = 1 \) to \( n \) do
23. \( q_i = -\eta \frac{\partial \mathcal{L}}{\partial \eta} \)
24. end for
25. end for
26. end for
27. end for
28. end for
29. return \( P, Q \).

Without utilizing global geographic context, for every user \( u \) and positive POI \( i \), we randomly select one negative sample \( j \) from \( |L_{ui}^i| \) (Line 11 in Alg. 1). However, as we noted before, POI \( j \in |L_{ui}^i| \) does not explicitly indicate that \( u \) is not interested in \( j \). It may be the result that user \( u \) does not know the existence of \( j \). If we randomly select negative one from \( |L_{ui}^i| \), we may select the POI that the user will visit in future. As the data is extremely sparse for POI recommendation, the selected negative sample may never get positive reward in the following update. Thus avoiding selecting the true sample as negative one would be important. To achieve this, for every user \( u \) and positive POI \( i \), we only select the negative POI \( j \) from her unvisited area \( B_u^i \). Moreover, we select the negative sample whose rank is higher than the positive one. If we cannot get such a sample in a specified iteration, we use the last ranked sample as the negative one, instead of skipping the update as in [42]. The whole process could be found in Algorithm 2.

### 5.3 Local Level Context

As discussed earlier, the proposed AUC-MF can be easily extended by incorporating contextual factors through recommendation refinement. As a case study, we illustrate how to extend AUC-MF by incorporating local context of POIs and users.

Local context has been reported effective in improving POI recommendation [12], [22]. However, most previous works utilize the local context in the training process e.g., through regularization. This makes the optimization more time-consuming and difficult to tune parameters. We propose to refine the recommendations made by AUC-MF to reflect the local context. This approach makes it possible to refine the recommendations of AUC-MF according to the different contextual factors depending on user preference or data availability.

To incorporate local context of POIs, we assume that users are more willing to visit POIs that are near the locations they have visited before, as in most other works incorporating local geographical context. We define a local \( k \) nearby neighborhood matrix \( N^P \in \mathbb{R}^{n \times n} \) as:

\[
N_{ij} = \begin{cases} 
S_{ij}^P & \text{if } p_j \in N_i(p_i), \\
0 & \text{otherwise} 
\end{cases},
\]

where \( N_i(p_i) \) is the set of \( p_i \)'s \( k \) nearest POIs with contextual similarity \( S^P \).

Given the predicted ranking matrix \( R_{pre}^P = PQ^\top \) by AUC-MF, we compute a ranking matrix with geographical context \( R^P \in \mathbb{R}^{m \times n} \) as

\[
R_{ij}^P = p_i Q^\top (N_j \otimes Z_i)^\top,
\]

where \( \otimes \) denotes the Hadamard product of two matrices. Under this way, we could guarantee that users will visit the POIs that is close to their visited POIs.

To incorporate the contextual information of users, we assume that users with social relationship will visit the similar POIs. Once we have user’s contextual similarity \( S^U \), we could get its Laplacian matrix \( L^U \). Then we define the objective function to obtain the ranking matrix \( R^U \in \mathbb{R}^{m \times n} \) as follows:

\[
\min_{R^U} \gamma \|R^U - R_{pre}^P\|^2_F + (1 - \gamma) \text{trace}((R^U)^\top L^U R^U),
\]

where \( \gamma \in [0, 1] \) is the weight to control the influence of user’s contextual similarity. The optimal solution for the above objective function is

\[
R^U = ((1 - \gamma)L^U + \gamma I)^{-1} R_{pre}^P.
\]

By solving the above objective function, we could guarantee that users with social relationships will visit similar POIs.

To adjust the importance of local geographical context in different datasets, we linearly combine \( R^U \) and \( R^P \) to get the final recommendation:

\[
\hat{R} = \mu R^U + (1 - \mu) R^P,
\]

where \( \mu \) represents the weight of the local geographical context.
where $\mu \in [0, 1]$ is the weight to control the ratio of local contextual influence.

## 6 Experiments

In this section, we empirically evaluate AUC-MF against state-of-the-art methods that use the same settings as ours. We also study the impact of different parameters to the performances of AUC-MF.

### 6.1 Experimental Settings

#### 6.1.1 Dataset

We use the dataset that has been used in [22], where the authors collected check-in data from Gowalla from November 2010 to June 2011. The other dataset is from the Yelp which has been used in [28], we sampled some users and check-ins from it. The statistics of the check-in data in the two datasets are summarized in Table 2. In our experiments, both datasets are divided into three parts. For each user, we sort the check-ins by timestamp, and choose the first 70 percent as training set, then 10 percent as development set, and the remaining 20 percent as test set.

#### 6.1.2 Evaluation Metrics

We evaluate the quality of POI recommendation by five metrics, Precision ($Pr$), Recall ($Re$), AUC, MAP and NDCG, as in most other works [19], [21]. Given a user $u$, $L^T_u$ denotes the set of corresponding visited locations in the testing data, and $L^R_u$ denotes the set of recommended locations by the algorithm. The definitions of precision and recall are:

$$Pr = \frac{1}{|U^T|} \sum_{u \in U^T} \frac{|L^T_u \cap L^R_u|}{K},$$

$$Re = \frac{1}{|U^T|} \sum_{u \in U^T} \frac{|L^T_u \cap L^R_u|}{|L^T_u|},$$

where $U^T$ is the set of users in the testing dataset and $K = 10$ in our experiments.

#### 6.1.3 Evaluated Methods

We compare the performances of 11 following models. Two of them utilize geographical context and achieve the top two performances in the latest experimental evaluation of POI recommendation task [28] (see Section 2 for more details).

- **Pop**: is a naive baseline that we recommend the most popular POIs for every user.
- **BRPRMF**: is a state-of-the-art approach designed for item recommendation with users’ implicit feedback [13].
- **AoBRP**: [43] proposed a non-uniform item sampler to overcome the long tail distribution.
- **IRenMF**: This model [44] exploited geographical characteristics from a location perspective, by modeling the geographical neighborhood of a location.
- **WRMF**: This is the weighted regularized matrix factorization model, designed for processing large scale implicit feedbacks.
- **RankGeoFM**: This is a ranking-based MF model [23] that (i) learns users’ preference rankings for POIs, and (ii) includes the geographical influence of neighboring POIs.
- **LRT**: LRT [44] models each user by different latent vectors for different time slots and the final recommendation score is computed from all the latent vectors.
- **NGCF**: NGCF [37] is a GNN-based recommendation method which utilizes GNN to do recommendation.
- **PRFPF**: PRFPF [33] is a POI recommendation method based on poisson factorization.
- **SSTPMF**: SSTPMF [34] is designed for POI recommendation by incorporating social spatial-temporal contextual information.
- **AUC-MF**: Our framework proposed in this paper which maximizes AUC metric by AUC-MF.
- **AUC-MF+G**: Our framework which incorporates the global geographic information into AUC-MF.
- **AUC-MF+GL**: Our framework combines the global and local geographic information into AUC-MF.

We use the development set to choose parameters for all algorithms. For AUC-MF and AUC-MF+G, the dimensionality of the latent space $d$ is selected from $\{50, 100, 200\}$, and for the other models, $d$ is chosen from $\{30, 50, 100, 200\}$. The regularization parameters are chosen from $10^{-4, -6, -8}$ (see Eq. (13)), and the optimal learning rates $\eta$ are selected from $10^{-10, -12, -14}$ (see Algorithm 1). For IRenMF, the geographic weight is chosen from $[0 : 0.1 : \ldots : 1]$. The number of iteration in AUC-MF and AUC-MF+G are all set to 50. All experiments are repeated for 5 times, each with a different random seed. The results reported are average of the 5 runs.

### 6.2 Experiments Results

The recommendation accuracies of AUC-MF and other baseline methods are summarized in Tables 3 and 4. We make the following observations:

**TABLE 2**

<table>
<thead>
<tr>
<th>Statistics of Datasets</th>
<th>Gowalla</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td>#users</td>
<td>5,880</td>
<td>6832</td>
</tr>
<tr>
<td>#POIs</td>
<td>46,617</td>
<td>10,398</td>
</tr>
<tr>
<td>#check-ins</td>
<td>1,130,331</td>
<td>273,214</td>
</tr>
<tr>
<td>average #users per POI</td>
<td>2.0</td>
<td>11.7</td>
</tr>
<tr>
<td>average #POI per user</td>
<td>15.9</td>
<td>17.2</td>
</tr>
</tbody>
</table>

**TABLE 3**

<table>
<thead>
<tr>
<th>Method</th>
<th>$Pr$</th>
<th>$Re$</th>
<th>AUC</th>
<th>MAP</th>
<th>NDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pop</td>
<td>1.17%</td>
<td>2.48%</td>
<td>57.5%</td>
<td>2.15%</td>
<td>6.34%</td>
</tr>
<tr>
<td>BRPRMF</td>
<td>4.17%</td>
<td>5.47%</td>
<td>70.1%</td>
<td>5.43%</td>
<td>17.47%</td>
</tr>
<tr>
<td>AoBRP</td>
<td>4.33%</td>
<td>6.29%</td>
<td>71.4%</td>
<td>5.74%</td>
<td>17.32%</td>
</tr>
<tr>
<td>WRMF</td>
<td>5.63%</td>
<td>7.00%</td>
<td>71.7%</td>
<td>6.77%</td>
<td>18.15%</td>
</tr>
<tr>
<td>LRT</td>
<td>5.62%</td>
<td>6.49%</td>
<td>71.9%</td>
<td>6.02%</td>
<td>18.19%</td>
</tr>
<tr>
<td>IRenMF</td>
<td>5.75%</td>
<td>7.10%</td>
<td>72.4%</td>
<td>6.90%</td>
<td>19.63%</td>
</tr>
<tr>
<td>RankGeoFM</td>
<td>5.29%</td>
<td>6.17%</td>
<td>72.0%</td>
<td>6.15%</td>
<td>18.57%</td>
</tr>
<tr>
<td>NGCF</td>
<td>5.04%</td>
<td>6.12%</td>
<td>71.1%</td>
<td>6.13%</td>
<td>18.15%</td>
</tr>
<tr>
<td>PRFPF</td>
<td>5.37%</td>
<td>6.84%</td>
<td>71.7%</td>
<td>6.72%</td>
<td>19.37%</td>
</tr>
<tr>
<td>SSTPMF</td>
<td>5.51%</td>
<td>7.03%</td>
<td>72.1%</td>
<td>6.81%</td>
<td>19.44%</td>
</tr>
<tr>
<td>AUC-MF</td>
<td>5.86%</td>
<td>7.82%</td>
<td>74.1%</td>
<td>6.93%</td>
<td>19.84%</td>
</tr>
<tr>
<td>AUC-MF+G</td>
<td>5.90%</td>
<td>7.91%</td>
<td>75.5%</td>
<td>7.04%</td>
<td>20.18%</td>
</tr>
<tr>
<td>AUC-MF+GL</td>
<td>6.06%</td>
<td>8.12%</td>
<td>77.3%</td>
<td>7.19%</td>
<td>20.50%</td>
</tr>
</tbody>
</table>
AUC-MF+GL outperforms all state-of-the-art baselines on both datasets on all metrics. In particular, on Gowalla dataset, in terms of recall, our method AUC-MF+GL outperforms BPRMF, AoBPR, WSMF, LRT, IRenMF, RankGeoFM, NGCF, PRFPF and SSTPMF by 48, 29, 16, 25, 14, 32, 33, 19 and 16 percent respectively.

AUC-MF gets better performance than all other algorithms without contextual information, i.e., BPRMF, WRMF, AoBPR and NGCF. The main reason is that AUC-MF focuses on the data sparsity problem and considers the unbalance between users, which are not considered in other methods.

Compared with AUC-MF, AUC-MF+G has a better performance on both datasets. This results show that the way we utilize global geographic context is effective.

AUC-MF+GL outperforms AUC-MF+G by a big margin on Yelp dataset, and outperforms all other state-of-the-art baselines. These results show linear combination could work well for incorporating contextual information.

### 6.3 Impact of Global Geographical Context

To evaluate the sensitivity to parameters for AUC-MF+G, we study its performance with respect to different numbers of regions and different dimensions of the clustering features. From Fig. 3, we observe that the number of regions has negligible impact on the performance of our method on two datasets. The reason may be that user would only be active in some small regions. As long as all negative samples are selected from the user’s inactive areas, the accuracy of the algorithm would not be affected. The other observation is that the dimension of the clustering features will influence the performance and convergence on both dataset. But the difference of performance will not increase with the increasing of dimensions. The reason may be that the clustering results will not change too much when the number of dimensions reaches a threshold.

### 6.4 Impact of Local Geographical Context

In AUC-MF+GL, the impact of local geographic context is controlled by a weight $\gamma$ (see Equation 30) and a weight $\mu$ (see Eq. (31)). Fig. 4 plots the impact of setting different values of these parameters in the range of $[0 : 0.1 : 1]$ to the final results. We could observe that, the benefit of incorporating local geographical context is dataset dependent. On the Gowalla dataset, the best results are obtained when $\mu = 0.8$ and $\gamma = 0.4$. On the Yelp dataset, the best results are obtained when $\mu$ is set as 0.7 and $\gamma$ is set as 0.5. This experimental results show the benefit of the flexility introduced by AUC-MF: the local contextual factors are considered by refining the results by AUC-MF+G. Because the refinement process is independent of the training process, it cost much less to tuning the weight to obtain best results using AUC-MF+G. Most existing methods require the contextual factors to be part of the model and the whole model needs to be retrained to evaluate the impact of different contextual factors.

### 6.5 Impact of Latent Vector Dimension

To measure the impact of latent vector, we study the performance of AUC-MF+G with different dimensions $d$. From Fig. 5, we observe that the dimension of the latent vector has impact on the convergence rate of AUC-MF+G on both datasets. On Yelp Dataset, AUC-MF+GL may converge at 14, 16

---

### TABLE 4

<table>
<thead>
<tr>
<th>Method</th>
<th>Pr</th>
<th>Re</th>
<th>AUC</th>
<th>MAP</th>
<th>NDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pop</td>
<td>0.81%</td>
<td>1.57%</td>
<td>63.8%</td>
<td>1.08%</td>
<td>3.45%</td>
</tr>
<tr>
<td>BPRMF</td>
<td>1.63%</td>
<td>3.03%</td>
<td>69.3%</td>
<td>2.65%</td>
<td>9.16%</td>
</tr>
<tr>
<td>AoBPR</td>
<td>1.72%</td>
<td>3.22%</td>
<td>69.8%</td>
<td>2.71%</td>
<td>9.32%</td>
</tr>
<tr>
<td>WSMF</td>
<td>1.83%</td>
<td>3.42%</td>
<td>71.7%</td>
<td>3.04%</td>
<td>10.05%</td>
</tr>
<tr>
<td>LRT</td>
<td>2.04%</td>
<td>4.01%</td>
<td>71.9%</td>
<td>3.18%</td>
<td>10.41%</td>
</tr>
<tr>
<td>IRenMF</td>
<td>2.51%</td>
<td>5.33%</td>
<td>72.5%</td>
<td>3.48%</td>
<td>11.17%</td>
</tr>
<tr>
<td>RankGeoFM</td>
<td>2.45%</td>
<td>5.41%</td>
<td>72.1%</td>
<td>3.24%</td>
<td>10.87%</td>
</tr>
<tr>
<td>NGCF</td>
<td>1.92%</td>
<td>3.57%</td>
<td>71.4%</td>
<td>3.11%</td>
<td>10.13%</td>
</tr>
<tr>
<td>PRFPF</td>
<td>2.53%</td>
<td>5.54%</td>
<td>71.3%</td>
<td>3.67%</td>
<td>11.21%</td>
</tr>
<tr>
<td>SSTPMF</td>
<td>2.48%</td>
<td>5.49%</td>
<td>72.4%</td>
<td>3.61%</td>
<td>11.19%</td>
</tr>
<tr>
<td>AUC-MF</td>
<td>2.11%</td>
<td>4.87%</td>
<td>73.5%</td>
<td>3.45%</td>
<td>10.29%</td>
</tr>
<tr>
<td>AUC-MF+G</td>
<td>2.37%</td>
<td>5.61%</td>
<td>73.9%</td>
<td>3.71%</td>
<td>11.28%</td>
</tr>
<tr>
<td>AUC-MF+GL</td>
<td>2.85%</td>
<td>5.75%</td>
<td>74.3%</td>
<td>3.79%</td>
<td>12.15%</td>
</tr>
</tbody>
</table>

---

![Fig. 3. Performance trend of AUC-MF+G measured by recall. The number of the clustered regions $n_c \in \{100, 200, 500\}$ and the dimension of feature $n_s \in \{1, 10, 50\}$.](image)

![Fig. 4. Impact of local context on the two datasets.](image)

Fig. 4. Impact of local context on the two datasets.
and 18 iterations respectively for 50, 100 and 200 dimensions of latent vector. And on Gowalla dataset, it would take 7, 10 and 13 iterations with respect to the different dimensions. On the recommendation accuracy, different dimensions lead to similar performance on each dataset. This could show the stability of our algorithm, which does not change too much under the variation of latent vector dimension.

6.6 Optimization Rate
With the same learning rate and same latent vector dimension, convergence rate varies on different datasets (Fig. 5). On the Yelp dataset, AUC maximization converges at about 14 iterations with 200 dimensions vector, but it only takes less than 10 iterations on the Gowalla dataset. That is because the number of check-ins in Gowalla dataset is bigger than that in Gowalla dataset. The latent vectors on the Gowalla dataset would be updated more times in one iteration.

7 CONCLUSION
In this paper, we propose a new framework, named AUC-MF, for POI recommendation with AUC maximization. To optimize the AUC metric, we utilize lambda-based method to generate an implicit cost function for this non-smooth function. To satisfy the two conditions of that method, we define a new constant lambda for AUC, which could make the implicit cost function exist and ensure it is convex. Then we combine the lambda-based method with matrix factorization as LambdaMF. We use SGD to optimize the cost function and draw random negative samples with universal distribution to reduce the optimizing time. We show that the proposed AUC-MF is flexible in incorporating contextual factors. With our proposed temporal-spatial similarity, we utilize geographic context from both local and global perspectives. A region-based sampling method and linear combination are proposed to incorporate contextual information into AUC-MF. Other contextual factors can be incorporated in a similar manner. Experiments on two real datasets show that our algorithm outperforms state-of-the-art baselines.

ACKNOWLEDGMENTS
This paper was supported by NSFC. U2001212, 62032001 and 61932004.

REFERENCES

Authorized licensed use limited to: Tencent. Downloaded on November 25,2022 at 07:22:36 UTC from IEEE Xplore. Restrictions apply.
Shuo Shang is a professor of computer science with the University of Electronic Science and Technology of China. He is also a guest editor of IEEE Transactions on Industrial Informatics, Geo-informatica, WWW Journal, Neural Computing with Applications, Neurocomputing, Big Data Research, and DAPD Journal. He is a regular TPC member of many top-ranked conferences including SIGMOD, VLDB, ICDE, KDD, AAAI, and IJCAI. He is on the reviewer board of several data science journals such as ACM Transactions on Database Systems, IEEE Transactions on Knowledge and Data Engineering, The VLDB Journal, ACM Transactions on Intelligent Systems and Technology, ACM Transactions on Spatial Algorithms and Systems, IEEE Transactions on Intelligent Transportation Systems, GeoInformatica, Information Sciences, Knowledge and Information Systems, Data Knowl. Eng., and WWW Journal.

Aixin Sun received the BASc (First Class Honors) and PhD degrees in computer engineering from the School of Computer Science and Engineering, Nanyang Technological University, Singapore, in 2001 and 2004, respectively. Currently, he is an associate professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research areas include text mining, social computing, multimedia, and digital libraries.

Peilin Zhao received the bachelor’s degree from the Department of Mathematics, Zhejiang University, and the PhD degree from the School of Computer Engineering, Nanyang Technological University. He is currently a research scientist in Tencent AI Lab, China. His research interests include large scale machine learning and its applications to big data analytics. In his research areas, he has published more than 70 papers in top conferences and journals. In addition, he has served as member or reviewer for many premier international conferences and journals.

Kai Zheng (Member, IEEE.) received the PhD degree in computer science from the University of Queensland, in 2012. He is a professor of computer science with the University of Electronic Science and Technology of China. He has been working in the area of spatial-temporal databases, uncertain databases, social-media analysis, in-memory computing and blockchain technologies. He has published more than 100 papers in prestigious journals and conferences in data management field such as SIGMOD, ICDE, VLDB, ACM Transactions and IEEE Transactions.

Xiangliang Zhang received the PhD degree in computer science from INRIA-University Paris-Sud, France, in July 2010. She is currently an associate professor and directs the Machine Intelligence and Knowledge Engineering (MINE) Laboratory at the Division of Computer, Electrical, and Mathematical Sciences and Engineering, King Abdullah University of Science and Technology (KAUST), Saudi Arabia. She has authored or co-authored more than 130 refereed papers in various journals and conferences. Her current research interests and experiences include machine learning, and data mining.

Peng Han is currently working toward the PhD degree in computer science from King Abdullah University of Science and Technology. His research interests include computer vision, recommendation system and distributed deep learning. He has published more than 10 papers in top conferences and journals.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.