Rank-GeoFM: A Ranking based Geographical Factorization Method for Point of Interest Recommendation

Xutao Li[1] Gao Cong[1] Xiao-Li Li[2] Tuan-Anh Nguyen Pham[1] Shonali Krishnaswamy[2]
[1]School of Computer Engineering, Nanyang Technological University, Singapore.
{lixutao@, gaocong@, pham0070@e.}ntu.edu.sg
[2]Institute for Infocomm Research(I2R), A*STAR, Singapore.
{xlli, spkrishna}@i2r.a-star.edu.sg

ABSTRACT

With the rapid growth of location-based social networks, Point of Interest (POI) recommendation has become an important research problem. However, the scarcity of the check-in data, a type of implicit feedback data, poses a severe challenge for existing POI recommendation methods. Moreover, different types of context information about POIs are available and how to leverage them becomes another challenge. In this paper, we propose a ranking based geographical factorization method, called Rank-GeoFM, for POI recommendation, which addresses the two challenges. In the proposed model, we consider that the check-in frequency characterizes users' visiting preference and learn the factorization by ranking the POIs correctly. In our model, POIs both with and without check-ins will contribute to learning the ranking and thus the data sparsity problem can be alleviated. In addition, our model can easily incorporate different types of context information, such as the geographical influence and temporal influence. We propose a stochastic gradient descent based algorithm to learn the factorization. Experiments on publicly available datasets under both user-POI setting and user-time-POI setting have been conducted to test the effectiveness of the proposed method. Experimental results under both settings show that the proposed method outperforms the state-of-the-art methods significantly in terms of recommendation accuracy.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Filtering

Keywords

Collaborative Filtering; Factorization Model; Ranking

1. INTRODUCTION

Recently, location-based social networks (LBSN) have emerged, such as Foursquare. These online systems enable people to check in and share their experiences with friends when they visit a point of interest (POI), e.g., restaurant, and shopping mall. These networks are growing at an unprecedented pace. Taking Foursquare as an example, it had attracted 45 million users with more than 5 billion check-ins until January 2014. The huge volume of data contains valuable information about POIs, and human preference, which can be exploited for POI recommendation [1].

POI recommendation aims at learning the users' visiting preferences and recommending a user the POIs *that s/he may be interested in but has never visited.* This task is important and meaningful, as it not only helps local residents or tourists to explore interesting unknown places in a city, but also creates the opportunities for POI owners to increase their revenues by finding and attracting potential visitors.

POI recommendation is challenging for two reasons. First, the check-in data in LBSN is very sparse, and thus recommendation methods suffer from the data scarcity problem. The check-in data is usually represented as a user-POI matrix, shown as in Figures 1(a) and 1(b). As we will see in the experiments, the density of check-in matrix is usually less than 0.5%. Moreover, when considering context-aware POI recommendation, the user-POI check-in data (matrix) needs to be separated and represented as a tensor, e.g., as shown in Figures 1(c) and 1(d) for time-aware POI recommendation. This will make the data more sparse, and the density of the check-in tensor in experiments is less than 0.05%, which is extremely small compared to 1.2% for Netflix data [2]. Worse still, the check-in is a type of implicit feedback [11], which makes the POI recommendation more difficult. Different from conventional movie rating data, where users explicitly denote their "like" or "dislike" to an item with different rating scores, the check-ins offer only *positive examples* that a user likes, and the POIs without check-ins, marked as "?" in Figures 1(a) and 1(c), are either unattractive or undiscovered but potentially attractive. In other words, we need to infer his/her preference and non-preference based on the check-in data. Most of the existing POI recommendation methods [22, 3, 14, 9, 7, 12] overlook data scarcity and implicit feedback facts, and adapt conventional memory or model-based collaborative filtering for POI recommendation. Therefore, these methods suffer from the data scarcity problem.

Second, in POI recommendation, different types of context information are available, e.g., geographical coordinates of POIs, time stamps of check-ins, friendship of users, categories of POIs, etc. It is important to exploit context information to improve the recommendation accuracy. For

SIGIR'15, August 09 - 13, 2015, Santiago, Chile

DOI: http://dx.doi.org/10.1145/2766462.2767722 .

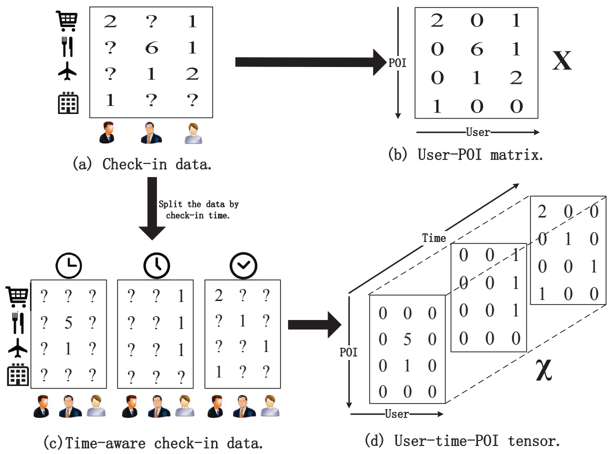


Figure 1: An example of check-in data of three users in two scenarios.

example, geographical coordinates, as an important type of context information which is not available in conventional recommendation tasks, are exploited for POI recommendation [22, 15, 3], since users have a much higher probability to visit nearby POIs [22, 15]. Previous work [22, 3, 12, 11] develops different approaches to exploiting the different types of context information. However, these approaches are usually developed for a particular type of context and it is difficult to generalize them to handle another type of context information.

In this paper, we propose a new ranking based factorization method for the POI recommendation problem. We consider obtaining factorizations in a ranking way because of two reasons: on the one hand it works better for implicit feedback data [16], and on the other hand the sparsity problem can be alleviated because both visited (positive examples) and unvisited POIs will contribute to the learning of a ranking function. Specifically, we assume that the higher the check-in frequency is, the more the POI is preferred by a user; and the unvisited POIs are less preferred than the visited ones. Instead of fitting the check-in frequency as conventional matrix factorization based methods do, we fit the users' preference rankings for POIs to learn the latent factors of users and POIs. In our proposed method, the unvisited POIs also contribute to the learning, which will help alleviate the sparsity problem.

In addition to addressing the data scarcity issue, our factorization model can easily incorporate different types of context information. In the proposed method, we compute the recommendation score of a POI by the sum of pairwise interaction scores between it and each context variable considered. This enables us to easily incorporate different types of context information by simply including more interaction scores. The pairwise prediction also unifies the way to handle matrix and tensor data. That is, the POI recommendation and context-aware POI recommendation can be handled with one unified framework.

Our contributions in this paper can be summarized as follows:

- We propose a new Ranking based Geographical Factorization Method (called Rank-GeoFM) for POI recommendation, which addresses the data scarcity problem and incorporates the geographical influence, an important type of context information.
- To optimize the objective function of the Rank-GeoFM, we use the stochastic gradient descent method. However, there are two difficulties in calculating stochastic gradient descents. First, the objective function is indifferentiable. To address this issue, we propose a continuous approximation to the objective function. Second, to calculate the stochastic gradients, we need compute the prediction scores of all POIs, which costs heavy computations. To tackle this issue, we develop a sampling based fast learning scheme to calculate the gradients.
- The proposed method can be generalized easily when other types of context information are available. We take temporal information as an example, and demonstrate how our model can be used to incorporate it for time-aware POI recommendation.
- Extensive experiments on two real-life datasets Foursquare and Gowalla demonstrate that Rank-GeoFM outperforms the state-of-the-art methods significantly for both POI recommendation and time-aware POI recommendation in terms of accuracy.

The rest of this paper is organized as follows: In Section 2, we briefly review related work. In Section 3, we introduce the proposed method. The experimental results are given in Section 4. Finally, we conclude the paper in Section 5.

2. RELATED WORK

2.1 POI Recommendation

POI Recommendation. Memory-based collaborative filtering (CF) techniques, such as user-based CF and item-based CF, are exploited for POI recommendation. Ye *et al.* [22] adopt linear interpolation to incorporate both the social and geographical influences into the user-based CF framework for POI recommendation. Their experimental results show that user-based CF outperforms item-based CF for POI recommendation, incorporating the geographical influence into the user-based CF model can significantly improve the recommendation accuracy, and the social influence has little impact on the performance. Levandoski *et al.* extend item-based CF method by considering the travel distance as penalty [9].

Memory-based CF methods easily suffer from the data sparsity problem, since the user-user or item-item similarities need to be calculated based on common check-ins. When the check-ins are scarce, two users or items will share very few common check-ins and thus the resulting similarities are unreliable for making effective recommendations.

Model-based CF techniques are also exploited for POI recommendation. Noulas *et al.* [14] find that matrix factorization (MF) performs worse than user-based CF and item-based CF for POI recommendation. In their work, conventional MF method for explicit feedback data is applied to POI recommendation, which thus is not suitable and delivers bad performance. Based on the observation that individual's check-in locations are usually around several centers, Cheng *et al.* [3] introduce a Multi-center Gaussian Model to compute the geographical influence and then combine it heuristically with MF for POI recommendation. In this approach, MF is performed by fitting nonzero check-ins only and thus it suffers from the data sparsity issue easily. Liu *et al.* [12] propose a geographical probabilistic factor analysis framework, namely GTBNMF, which incorporates geographical influence and textual influence based on the Bayesian Nonnegative Matrix Factorization (BNMF). However, the BNMF is performed by fitting both zero and nonzero check-ins, which might not be reasonable because zero check-ins may be missing values and should not be fitted directly. All these factorization methods do not exploit the implicit feedback property of POI recommendation.

Recently, considering the check-ins as implicit feedbacks, Lian *el al.* [11] develop a model, namely GeoMF, based on Weighted Matrix Factorization (WMF), and the geographical influence is incorporated into WMF. The method fits the nonzero check-ins by using large weights and zero check-ins by using smaller weights. Although assigning large weights can highlight nonzero check-ins, directly fitting zero check-ins may not be very reasonable because zero check-ins may be missing values. Moreover, because of the limitation of WMF, it is not easy to generalize the method to other types of context information.

Context-aware POI recommendation. Most of the aforementioned methods for POI recommendation exploit the geographical influence in making recommendations. However, their approaches of handling the geographical influence cannot handle other types of context information. Time is another important type of context, and time-aware POI recommendation aims to recommend POIs for a user at a given time. Yuan *et al.* propose a method called UTE+SE, which extends the user-based CF to incorporate both the temporal and geographical effects with a linear combination framework [24]. Yuan *et al.* further present a graph based method, called BPP, for time-aware POI recommendation [25], which makes recommendations by preference propagation on a graph constructed from the check-in data. BPP [25] performs better than UTE+SE [24]. In addition, Gao *et al.* [7] study the temporal effect on the POI recommendation, but not the time-aware POI recommendation. They develop a regularized nonnegative matrix factorization method but they do not consider the geographical influence.

Other types of contexts utilized in POI recommendation include category of POI [13], text description of POI [23, 12, 10, 8, 26], and currently-visited POI [4, 6].

2.2 Ranking based Learning Criteria

Bayesian personalized ranking (BPR) [16] is a famous ranking-based objective criterion, which can produce promising performance for implicit feedback problems combining with matrix factorization model. BPR learns the ranking models based on pairwise comparison of items such that the Area Under the ROC Curves (AUC) can be maximized. It gives equal weights to each item pair [19, 21]. Ordered Weighted Pairwise Classification (OWPC) [20] is another recently proposed loss metric for ranking. The method considers ranking as a set of pairwise classification problems and emphasizes the classifications at top-N positions by assigning higher weights. OWPC has been successfully applied in text retrieval [20] and image annotation [21].

In this paper, we consider the POI recommendation based on the OWPC criterion. Our proposed method differs from the existing approaches [20, 21] in two aspects. First, existing OWPC is developed for ranking problem with binary values, i.e., relevance or irrelevance, while in this paper we extend the objective function to rank POIs with different visiting frequencies, and provide the solutions for stochastic gradient descent optimization. Second, we develop a general factorization method for POI recommendation, which is able to exploit different types of context information.

Table 1: List of notations

$\mathcal{U}$	the set of users $\{u_1, u_2, \cdots, u_{\lvert\mathcal{U}\rvert}\}$
$\mathcal{L}$	the set of POIs $\{\ell_1, \ell_2, \cdots, \ell_{\lvert\mathcal{L}\rvert}\}$
$\mathcal{T}$	the set of time slots $\{t_1, t_2, \cdots, t_{\lvert\mathcal{T}\rvert}\}$
$\mathcal{L}^u$	the set of POIs that user u has visited
$\mathbf{X} = [x_{u\ell}]$	a $\lvert\mathcal{U}\rvert \times \lvert\mathcal{L}\rvert$ user-POI check-in matrix
$\mathcal{X} = [x_{ut\ell}]$	a $\lvert\mathcal{U}\rvert \times \lvert\mathcal{T}\rvert \times \lvert\mathcal{L}\rvert$ user-time-POI tensor
$\mathcal{D}_1$	the user-POI pairs: $\{(u,\ell) \lvert x_{u\ell} > 0\}$
$\mathcal{D}_2$	the user-time-POI tuples: $\{(u,t,\ell) \lvert x_{ut\ell} > 0\}$
$d(\ell, \ell')$	the distance between POIs ℓ and ℓ'
$\mathcal{N}_k(\ell)$	the set of k nearest POIs of ℓ
$y_{u\ell}$	recommendation score of POI ℓ for user u

3. PROPOSED METHOD

In this section, we present the proposed ranking based factorization method for POI recommendation. We first formulate the POI recommendation problem by a ranking objective function, and then introduce how to optimize it. Finally, we generalize the model for time-aware POI recommendation.

We first summarize the notations used in this paper in Table 1. Then the POI recommendation problem is defined as follows:

DEFINITION 1. ***POI recommendation****: given a user u, we recommend the POIs that s/he will like to visit, and are not in $\mathcal{L}^u$.*

3.1 Ranking based Geographical Factorization

3.1.1 Preference Ranking Objective Function

In this subsection, we formulate our objective function for POI recommendation. Due to the sparsity of check-in data, we design objective function by fitting user's preference rankings for POIs, instead of fitting his/her check-in frequencies as traditional factorization methods do.

First, we need to infer a user's preference rankings for POIs based on his/her check-in data. Intuitively, we assume that the higher the check-in frequency is, the more the POI is preferred by a user; and the unvisited POIs are less preferred than the visited ones. In other words, for a given user u, POI ℓ should be ranked higher than POI ℓ' if $x_{u\ell} > x_{u\ell'}$, where $x_{u\ell}$ denotes the frequency that user u visited POI ℓ.

Based on the intuition, we develop a method to measure the incompatibility between the inferred rankings and the rankings produced by a factorization model. In particular, for a given user u and POI ℓ, the incompatibility can be measured by:

$$Incomp(y_{u\ell}, \varepsilon) = \sum_{\ell' \in \mathcal{L}} I(x_{u\ell} > x_{u\ell'}) I(y_{u\ell} < y_{u\ell'} + \varepsilon) \quad (1)$$

where $I(\cdot)$ is an indicator function, $I(a) = 1$ when a is true, and 0 otherwise; ε is a positive number; $y_{u\ell}$ denotes the recommendation score of POI ℓ for user u, which will be calculated by a factorization model in this paper. We can see Eq. (1) counts the number of POIs that are supposed to be ranked lower than ℓ for user u according to the check-in data, but ranked higher than ℓ by the factorization model. Note that ε-margin is used to compute the rankings for the factorization model, i.e., we consider that ℓ' is ranked higher than ℓ for user u only if $y_{u\ell} < y_{u\ell'} + \varepsilon$. The $Incomp(y_{u\ell}, \varepsilon)$ measures the number of POIs that are incorrectly ranked higher than ℓ for user u, and we call it "ranking incompatibility" in this paper.

Next, we design our preference ranking objective function for learning a factorization model. Specifically, a good factorization method should minimize the ranking incompatibility as much as possible, and thus we propose the following objective function for minimizing:

$$\mathcal{O} = \sum_{(u,\ell)\in\mathcal{D}_1} E\left(Incomp(y_{u\ell}, \varepsilon)\right) \tag{2}$$

where $E(\cdot)$ is a function used to convert the ranking incompatibility $Incomp(y_{u\ell}, \varepsilon)$ into a loss:

$$E(r) = \sum_{i=1}^{r} \frac{1}{i} \tag{3}$$

and we define $E(0) = 0$.

In Eq. (2), we aggregate the losses incurred for all the user-POI pairs in $\mathcal{D}_1$ to compute the overall loss. We note that $Incomp(y_{u\ell}, \varepsilon)$, according to Eq. (1), is always equal to zero for the user-POI pairs $(u, \ell) \notin \mathcal{D}_1$. Following OWPC [20, 21], we adopt a smooth weighting scheme to convert $Incomp(y_{u\ell}, \varepsilon)$ into a loss. Specifically, the function $E(r)$ in Eq. (3) embodies such a conversion. It can be seen that $E(r)$ calculates the sum over losses at each rank position (from 1 to r) for the incorrectly-ranked POIs, where each position i is assigned with a loss $1/i$. For example, assume we have $Incomp(y_{u\ell}, \varepsilon) = 3$, i.e., three POIs are incorrectly ranked higher than POI ℓ for user u. The loss for this pair (u, ℓ) is thus given by $E(3) = 1 + \frac{1}{2} + \frac{1}{3}$.

One merit of our objective function is its ability to overcome the data sparsity issue. According to Eq. (1), the ranking incompatibility of a POI ℓ for a user is determined by all the other POIs $\ell' \in \mathcal{L}$, which are mostly unvisited POIs (because a user often visits very few POIs). Therefore, the unvisited POIs also contribute to learning the model, while they are ignored in conventional MF. Hence, by leveraging the objective function, we can address the sparsity problem of check-in data, without directly fitting zero check-ins.

3.1.2 Geographical Factorization Method

In this subsection, we propose a ***geographical factorization method*** for calculating the recommendation score. Our factorization model is capable of characterizing the user's preferences over POIs. In addition, it also incorporates the influence of the geographical context for POI recommendation. On the one hand, we parameterize the latent factors of users and POIs into a K-dimensional space as matrices $\mathbf{U}^{(1)} \in \mathbb{R}^{|\mathcal{U}|\times K}$ and $\mathbf{L}^{(1)} \in \mathbb{R}^{|\mathcal{L}|\times K}$, respectively. They are used to model the user's own preference as traditional matrix factorization methods do. On the other hand, we introduce one extra latent factor matrix $\mathbf{U}^{(2)} \in \mathbb{R}^{|\mathcal{U}|\times K}$ for users, and employ $\mathbf{U}^{(2)}$ to model the interaction between users and POIs for incorporating the geographical influence. To this end, we further construct an $|\mathcal{L}| \times |\mathcal{L}|$ geographical influence matrix $\mathbf{W}$, where $w_{\ell\ell'}$ is the probability that POI ℓ is visited given that POI ℓ' has been visited. By following previous studies [22, 3, 12], we set $w_{\ell\ell'} = (0.5+d(\ell, \ell'))^{-1}$ if $\ell' \in \mathcal{N}_k(\ell)$, and 0 otherwise. Here we consider only k-nearest neighbors $\mathcal{N}_k(\ell)$ of each POI ℓ. The intuition behind the formula is that users usually tend to visit nearby POIs. We normalize each row of the matrix $\mathbf{W}$ such that $\sum_{\ell'\in\mathcal{L}} w_{\ell\ell'} = 1$, because it represents the influence probabilities.

Let $\Theta = \{\mathbf{U}^{(1)}, \mathbf{L}^{(1)}, \mathbf{U}^{(2)}\}$ denote the parameters of our geographical factorization model. We will present the proposed method of learning these parameters in Section 3.2. Suppose that these parameters are already learned. Given user u and POI ℓ, we compute recommendation score $y_{u\ell}$ as follows:

$$y_{u\ell} = \mathbf{u}_u^{(1)} \cdot \mathbf{l}_\ell^{(1)} + \mathbf{u}_u^{(2)} \cdot \sum_{\ell^*\in\mathcal{N}_k(\ell)} w_{\ell\ell^*} \mathbf{l}_{\ell^*}^{(1)} \tag{4}$$

where operator $\cdot$ denotes the inner product, and $\mathbf{u}_u^{(1)}$ represents the u-th row of matrix $\mathbf{U}^{(1)}$. Similar notations are used for other matrices. In Eq. (4), the first term models the ***user-preference score***, while the second term models the ***geographical influence score*** that a user likes a POI because of its neighbors.

To avoid the overfitting problem, we constrain the latent factors of our model into a ball, which acts as a regularizer [21]. Specifically, we have the following constraints:

$$\left\|\mathbf{u}_u^{(1)}\right\|_2 \le C, \qquad u = 1, 2, \ldots, |\mathcal{U}| \tag{5}$$

$$\left\|\mathbf{l}_\ell^{(1)}\right\|_2 \le C, \qquad \ell = 1, 2, \ldots, |\mathcal{L}| \tag{6}$$

$$\left\|\mathbf{u}_u^{(2)}\right\|_2 \le \alpha C, \qquad u = 1, 2, \ldots, |\mathcal{U}| \tag{7}$$

where $C > 0$ and $0 \le \alpha \le 1$ are hyperparameters. We constrain the latent factors from $\mathbf{U}^{(1)}$ and $\mathbf{L}^{(1)}$ into a small ball with radius C, and constrain the latent factor from $\mathbf{U}^{(2)}$ into a smaller ball with radius αC. Here, we introduce the hyperparameter α to balance the contributions of user-preference and geographical influence scores. Using basic algebra, we have $|\mathbf{a} \cdot \mathbf{b}| \le \|\mathbf{a}\|_2 \|\mathbf{b}\|_2$ for any two vectors $\mathbf{a}$ and $\mathbf{b}$. Thus, we know that the user-preference score $\mathbf{u}_u^{(1)} \cdot \mathbf{l}_\ell^{(1)}$ is always in $[-C^2, C^2]$. As we have $\left\|\sum_{\ell^*\in\mathcal{N}_k(\ell)} w_{\ell\ell^*} \mathbf{l}_{\ell^*}^{(1)}\right\|_2 \le \sum_{\ell^*\in\mathcal{N}_k(\ell)} w_{\ell\ell^*} \left\|\mathbf{l}_{\ell^*}^{(1)}\right\|_2 \le \sum_{\ell^*\in\mathcal{N}_k(\ell)} w_{\ell\ell^*} C = C$, the geographical influence score $\mathbf{u}_u^{(2)} \cdot \sum_{\ell^*\in\mathcal{N}_k(\ell)} w_{\ell\ell^*} \mathbf{l}_{\ell^*}^{(1)}$ is thus in $[-\alpha C^2, \alpha C^2]$. As a result, tuning the hyperparameter α can balance the contributions of user-preference and geographical influence scores to the final recommendation score.

3.2 Optimization and Learning Algorithm

Next we present the proposed method of learning model parameters Θ such that $\mathcal{O}$ in Eq. (2) is minimized. As $\mathcal{O}$ is computed by summing the loss for each user-POI pair, we adopt the stochastic gradient descent (SGD) method for optimization. That is, we aim to minimize $E\left(Incomp(y_{u\ell}, \varepsilon)\right)$ given each training instance $(u, \ell) \in \mathcal{D}_1$. However, there are two difficulties: (1) $E\left(Incomp(y_{u\ell}, \varepsilon)\right)$ is non-continuous and indifferentiable, which makes it hard to optimize; (2) we need to know $Incomp(y_{u\ell}, \varepsilon)$ for optimization; however, calculating it is time-consuming. Next we introduce how to address the two issues.

3.2.1 Continuous Approximation

In order to make $E\left(Incomp(y_{u\ell},\varepsilon)\right)$ continuous over parameters Θ, we rewrite it as follows:

$$
\begin{aligned}
& E\left(Incomp(y_{u\ell},\varepsilon)\right)\cdot 1 & (8)\\
= \;& E\left(Incomp(y_{u\ell},\varepsilon)\right)\frac{\sum_{\ell'\in\mathcal{L}} I(x_{u\ell}>x_{u\ell'})I(y_{u\ell}<y_{u\ell'}+\varepsilon)}{Incomp(y_{u\ell},\varepsilon)} \\
\approx \;& E\left(Incomp(y_{u\ell},\varepsilon)\right)\frac{\sum_{\ell'\in\mathcal{L}(u,\ell)} s(y_{u\ell'}+\varepsilon-y_{u\ell})}{Incomp(y_{u\ell},\varepsilon)}
\end{aligned}
$$

where $\mathcal{L}(u,\ell) := \{\ell' | I(x_{u\ell}>x_{u\ell'})I(y_{u\ell}<y_{u\ell'}+\varepsilon)=1\}$, and $s(a) := \frac{1}{1+\exp(-a)}$ is the sigmoid function, which is used to approximate the indicator function. Based on the rewriting, we can compute the stochastic gradient for updating Θ. Specifically, we have

$$
\begin{aligned}
& \frac{\partial E\left(Incomp(y_{u\ell},\varepsilon)\right)}{\partial\Theta} & (9)\\
\approx \;& E\left(Incomp(y_{u\ell},\varepsilon)\right)\frac{\sum_{\ell'\in\mathcal{L}(u,\ell)} \frac{\partial s(y_{u\ell'}+\varepsilon-y_{u\ell})}{\partial\Theta}}{Incomp(y_{u\ell},\varepsilon)} \\
= \;& \frac{E\left(Incomp(y_{u\ell},\varepsilon)\right)}{Incomp(y_{u\ell},\varepsilon)}\sum_{\ell'\in\mathcal{L}(u,\ell)}\delta_{u\ell\ell'}\frac{\partial(y_{u\ell'}+\varepsilon-y_{u\ell})}{\partial\Theta}
\end{aligned}
$$

where $\delta_{u\ell\ell'} = s(y_{u\ell'}+\varepsilon-y_{u\ell})(1-s(y_{u\ell'}+\varepsilon-y_{u\ell}))$. We note that Eq. (9) is not a standard gradient computation, because $E\left(Incomp(y_{u\ell},\varepsilon)\right)$ and $Incomp(y_{u\ell},\varepsilon)$ are also related to Θ, but we do not consider their derivatives. Our analogous gradient calculation here follows the idea in [21]. Although the stochastic gradient can be calculated by Eq. (9), it is infeasible in practice. This is because both the summation and $Incomp(y_{u\ell},\varepsilon)$ in Eq. (9) require to compute the recommendation scores of all POIs as Eq. (4), which costs $O(K|\mathcal{L}|k)$ operations and is time-consuming.

In the next subsection, we introduce a fast learning scheme to address the issue.

3.2.2 Fast Learning Scheme

Our key idea of fast learning is to eliminate the summation and estimate $Incomp(y_{u\ell},\varepsilon)$ with a sampling method.

Let us first revisit Eq. (8). We can see from the first equality in Eq. (8) that only incorrectly-ranked POI, i.e., the POI ℓ' satisfying $I(x_{u\ell}>x_{u\ell'})I(y_{u\ell}<y_{u\ell'}+\varepsilon)=1$, contributes to the loss $E\left(Incomp(y_{u\ell},\varepsilon)\right)$. Thus, Eq. (8) can be reinterpreted as an expectation of the loss incurred by a set of incorrectly-ranked POI samples, where each POI sample ℓ' incurs a loss:

$$\bar{E} = E\left(Incomp(y_{u\ell},\varepsilon)\right) s(y_{u\ell'}+\varepsilon-y_{u\ell}) \tag{10}$$

and each POI sample has the probability $\frac{1}{Incomp(y_{u\ell},\varepsilon)}$ to be chosen. This motivates us to approximately calculate the stochastic gradient by sampling one incorrectly-ranked POI. In this case, we have

$$\frac{\partial\bar{E}}{\partial\Theta} = E\left(Incomp(y_{u\ell},\varepsilon)\right)\delta_{u\ell\ell'}\frac{\partial(y_{u\ell'}+\varepsilon-y_{u\ell})}{\partial\Theta} \tag{11}$$

which is an approximation of Eq. (9). Apparently, the summation is eliminated.

To calculate gradients by Eq. (11), we still need to know $Incomp(y_{u\ell},\varepsilon)$. We compute its approximate value by sampling one incorrectly-ranked POI. Specifically, given a user-POI pair (u,ℓ), we repeat sampling one POI from $\mathcal{L}$ until we obtain an incorrectly-ranked POI ℓ' such that $I(x_{u\ell} > x_{u\ell'})I(y_{u\ell}<y_{u\ell'}+\varepsilon)=1$. Let n denote the number of sampling trials before obtaining such a POI ℓ'. Apparently, n follows a geometric distribution with parameter $p=\frac{Incomp(y_{u\ell},\varepsilon)}{|\mathcal{L}|}$. Since we know the expectation of a geometrical distribution with parameter p is $\frac{1}{p}$, we have $n\approx\left\lfloor\frac{1}{p}\right\rfloor = \left\lfloor\frac{|\mathcal{L}|}{Incomp(y_{u\ell},\varepsilon)}\right\rfloor$. Thus, we can estimate $Incomp(y_{u\ell},\varepsilon)\approx\left\lfloor\frac{|\mathcal{L}|}{n}\right\rfloor$. Our idea here is similar to that used in [21] for a different problem.

Algorithm 1: Rank-GeoFM

input : check-in data $\mathcal{D}_1$, geographical influence matrix $\mathbf{W}$, hyperparameters ε, C and α, and learning rate γ
output: model parameters $\Theta=\{\mathbf{U}^{(1)},\mathbf{L}^{(1)},\mathbf{U}^{(2)}\}$

Initialize Θ with Normal distribution $\mathcal{N}(0, 0.01)$;
Shuffle the samples in $\mathcal{D}_1$ randomly;
repeat
  for $(u,\ell)\in\mathcal{D}_1$ **do**
    Compute $y_{u\ell}$ as Eq. (4), and set n=0;
    repeat
      Sample a POI ℓ';
      Compute $y_{u\ell'}$ as Eq. (4), and set $n=n+1$;
    until $I(x_{u\ell}>x_{u\ell'})I(y_{u\ell}<y_{u\ell'}+\varepsilon)=1$ *or* $n>|\mathcal{L}|$;
    if $I(x_{u\ell}>x_{u\ell'})I(y_{u\ell}<y_{u\ell'}+\varepsilon)=1$ **then**
      $\eta = E\left(\left\lfloor\frac{|\mathcal{L}|}{n}\right\rfloor\right)\delta_{u\ell\ell'}$;
      $\mathbf{g} = \left(\sum_{\ell^*\in\mathcal{N}_k(\ell')} w_{\ell,\ell^*}\mathbf{l}^{(1)}_{\ell^*} - \sum_{\ell^+\in\mathcal{N}_k(\ell)} w_{\ell,\ell^+}\mathbf{l}^{(1)}_{\ell^+}\right)$;
      $\mathbf{u}^{(1)}_u \leftarrow \mathbf{u}^{(1)}_u - \gamma\eta(\mathbf{l}^{(1)}_{\ell'}-\mathbf{l}^{(1)}_{\ell})$;
      $\mathbf{u}^{(2)}_u \leftarrow \mathbf{u}^{(2)}_u - \gamma\eta\mathbf{g}$;
      $\mathbf{l}^{(1)}_{\ell'} \leftarrow \mathbf{l}^{(1)}_{\ell'} - \gamma\eta\mathbf{u}^{(1)}_u$;
      $\mathbf{l}^{(1)}_{\ell} \leftarrow \mathbf{l}^{(1)}_{\ell} + \gamma\eta\mathbf{u}^{(1)}_u$;
    Project the updated latent factors to enforce constraints in Eqs. (5)∼(7), e.g., if $\left\|\mathbf{u}^{(1)}_u\right\|_2 > C$, then set $\mathbf{u}^{(1)}_u \leftarrow C\frac{\mathbf{u}^{(1)}_u}{\left\|\mathbf{u}^{(1)}_u\right\|_2}$;
until *convergence*;
return $\Theta=\{\mathbf{U}^{(1)},\mathbf{L}^{(1)},\mathbf{U}^{(2)}\}$

By using the estimation, we rewrite Eq. (11) by:

$$\frac{\partial\bar{E}}{\partial\Theta} \approx E\left(\left\lfloor\frac{|\mathcal{L}|}{n}\right\rfloor\right)\delta_{u\ell\ell'}\frac{\partial(y_{u\ell'}+\varepsilon-y_{u\ell})}{\partial\Theta} \tag{12}$$

Here $E\left(\left\lfloor\frac{|\mathcal{L}|}{n}\right\rfloor\right)$ acts as an adaptive scaling factor for the gradient. When n is small, the gradient will have a large magnitude, which is reasonable because a small n implies that $Incomp(y_{u\ell},\varepsilon)$ is large, and in this case the parameters should be updated with large steps. Similarly, when n is large, we will obtain a reasonable gradient with a small magnitude. With Eq. (12), the SGD based optimization is performed as follows:

$$\Theta \leftarrow \Theta - \gamma\frac{\partial\bar{E}}{\partial\Theta} \tag{13}$$

where γ is the learning rate.

Using the gradient calculation in Eq. (12) can gain significant speedups. The complexity of Eq. (9) is $O(K|\mathcal{L}|k)$ while the complexity of Eq. (12) is $O(Knk)$. In general, we have $n \ll |\mathcal{L}|$ at the start of training and $n < |\mathcal{L}|$ when the training reaches a stable phase. At the beginning, the model is not well-trained and thus $Incomp(y_{u\ell},\varepsilon)$ is often large, which leads to a very small n, i.e., $n \ll |\mathcal{L}|$; when the training reaches a stable phase, it is expected that more visited POIs are ranked correctly, and thus $Incomp(y_{u\ell}, \varepsilon)$ becomes smaller and n will become a bit larger. However, it is very unlikely that every visited POI is ranked correctly, so in general we still have $n < |\mathcal{L}|$. We find that the speedup is in orders of magnitude in our experiments.

We summarize the proposed Ranking based Geographical Factorization Method (Rank-GeoFM) in **Algorithm** 1. In the algorithm, we iterate through all the user-POI check-in pairs in $\mathcal{D}_1$ and update the latent factors until the procedure converges (lines 3∼16). In each iteration, given a user-POI pair, the sampling process is first performed so as to estimate $Incomp(y_{u\ell}, \varepsilon)$ and obtain one POI sample (lines 6∼8). Based on the estimation of $Incomp(y_{u\ell}, \varepsilon)$ and the sampled POI ℓ', we update the relevant latent factors by using S-GD method (lines 9∼15). The norm constraints are checked for the updated latent factors, and the ones violating the constraints are projected (line 16).

3.3 Time-aware POI Recommendation

In this subsection, we use the temporal information as an example context to illustrate how our method can be easily generalized to incorporate other types of context. We consider time-aware POI recommendation.

DEFINITION 2. ***Time-aware POI recommendation**: given a user u and time slot t (e.g., 3:00pm∼4:00pm), we recommend the POIs that s/he will like to visit in this time slot, and are not in $\mathcal{L}^u$.*

The objective function in Eq. (2) can be easily extended to incorporate the temporal context. To calculate the recommendation score, we extend Eq. (4) with two additional terms capturing the temporal factor. One term is ***temporal popularity score*** which indicates whether this POI is popular in the time slot. The other term is referred to as ***temporal influence score***, which is based on the following observation made in previous work [24, 25]: the popularity of a POI at one time slot is always influenced by some close or similar time slots, i.e., popularity are correlated among close or similar time slots.

We introduce three more latent factor matrices besides $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and $\mathbf{L}^{(1)}$ for the two additional terms. Specifically, we parameterize the latent factors of time slots as a $|\mathcal{T}| \times K$ matrix $\mathbf{T}$; and parameterize another two latent factor matrices $\mathbf{L}^{(2)}$ and $\mathbf{L}^{(3)}$ for POIs, where $\mathbf{L}^{(2)}$ is to model interactions with time slot for temporal popularity score, and $\mathbf{L}^{(3)}$ is to model interactions with close or similar time slot for temporal influence score. We further construct a $|\mathcal{T}| \times |\mathcal{T}|$ matrix $\mathbf{M}$, where m_{tt^*} is the probability that the popularity scores of POIs in time slot t are influenced by those in time slot t^*. By following previous studies [24, 7], we compute m_{tt^*} as:

$$m_{tt^*} = \frac{\sum_{u\in\mathcal{U}}\sum_{\ell\in\mathcal{L}} x_{ut\ell}x_{ut^*\ell}}{\sqrt{\sum_{u\in\mathcal{U}}\sum_{\ell\in\mathcal{L}} x_{ut\ell}^2}\sqrt{\sum_{u\in\mathcal{U}}\sum_{\ell\in\mathcal{L}} x_{ut^*\ell}^2}} \quad (14)$$

and we normalize $\mathbf{M}$ into a matrix such that each row is a probability vector. Given a user u and time slot t, the recommendation score of POI ℓ is computed as follows:

$$\begin{aligned} y_{ut\ell} \;=\;& \mathbf{u}_u^{(1)} \cdot \mathbf{l}_\ell^{(1)} + \mathbf{t}_t \cdot \mathbf{l}_\ell^{(2)} + \\ & \mathbf{u}_u^{(2)} \cdot \sum_{\ell^*\in\mathcal{N}_k(\ell)} w_{\ell\ell^*}\mathbf{l}_{\ell^*}^{(1)} + \mathbf{l}_\ell^{(3)} \cdot \sum_{t^*\in\mathcal{T}} m_{tt^*}\mathbf{t}_{t^*} \end{aligned} \quad (15)$$

where the four terms indicate the user-preference score, temporal popularity score, geographical influence score and temporal influence score, respectively. We denote the parameters of our factorization model in this setting as $\Theta = \{\mathbf{U}^{(1)}, \mathbf{L}^{(1)}, \mathbf{U}^{(2)}, \mathbf{T}, \mathbf{L}^{(2)}, \mathbf{L}^{(3)}\}$. Similar to POI-user setting, norm constraints are imposed to prevent overfitting:

$$\|\mathbf{t}_t\|_2 \le C, \qquad t = 1, 2, \ldots, |\mathcal{T}| \quad (16)$$

$$\left\|\mathbf{l}_\ell^{(2)}\right\|_2 \le C, \qquad \ell = 1, 2, \ldots, |\mathcal{L}| \quad (17)$$

$$\left\|\mathbf{l}_\ell^{(3)}\right\|_2 \le \beta C, \qquad \ell = 1, 2, \ldots, |\mathcal{L}| \quad (18)$$

where $0 \le \beta \le 1$ is to control the importance of temporal influence in computing the recommendation score.

Algorithm 1 can be adapted easily for time-aware POI recommendation. Specifically, we iterate through all user-time-POI tuple in $\mathcal{D}_2$ to update the latent factors. Given each $(u, t, \ell) \in \mathcal{D}_2$, we keep sampling POIs until obtain one POI ℓ' satisfying $I(x_{ut\ell} > x_{ut\ell'})I(y_{ut\ell} < y_{ut\ell'} + \varepsilon) = 1$, and again $Incomp(y_{ut\ell}, \varepsilon)$ is estimated as $\left\lfloor \frac{|\mathcal{L}|}{n} \right\rfloor$. The stochastic gradient can be calculated similarly by:

$$\frac{\partial \bar{E}}{\partial \Theta} \approx E\left(\left\lfloor \frac{|\mathcal{L}|}{n} \right\rfloor\right)\delta_{ut\ell\ell'}\frac{\partial(y_{ut\ell'} + \varepsilon - y_{ut\ell})}{\partial \Theta} \quad (19)$$

where $\delta_{ut\ell\ell'} = s(y_{ut\ell'}+\varepsilon-y_{ut\ell})(1-s(y_{ut\ell'}+\varepsilon-y_{ut\ell}))$. Based on it, the relevant latent factors are updated accordingly.

4. EXPERIMENTS

We conduct comprehensive experiments to evaluate the performance of the proposed method for both POI recommendation and time-aware POI recommendation.

4.1 Experimental Setup

4.1.1 Datasets

We use two real-world datasets [24][1] in our experiments. One is the Foursquare check-in data made in Singapore between Aug. 2010 and Jul. 2011, and the other is the Gowalla check-in data made in California and Nevada between Feb. 2009 and Oct. 2010. The Foursquare data comprises 194,108 check-ins made by 2,321 users at 5,596 POIs, and the Gowalla data comprises 456,988 check-ins made by 10,162 users at 24,250 POIs. Each check-in is associated with a time stamp.

For each user, we mask off 20% of his/her most recent check-ins as *testing set* to evaluate the performance of different algorithms. The earliest 70% of check-ins are used as *training set*, and the remaining 10% check-ins are used as *validation set*. Based on the training sets, we construct a user-POI matrix $\mathbf{X}$ with 73,011 and 210,894 nonzero entries for Foursquare and Gowalla, respectively, which will be used in the POI recommendation. The densities of $\mathbf{X}$ are 0.56% and 0.085% for Foursquare and Gowalla matrices, respectively. For time-aware POI recommendation, we split the data into 24 hours (time slots), and then obtain a tensor $\mathcal{X}$ with 91,228 and 244,580 nonzero entries for Foursquare and Gowalla, respectively. The density of $\mathcal{X}$ are 0.029% and 0.0041% for the two data, respectively. We can see that both matrix $\mathbf{X}$ and tensor $\mathcal{X}$ are very sparse.

[1]Available at http://www.ntu.edu.sg/home/gaocong/data/poidata.zip

4.1.2 Metrics

We use two widely used metrics to evaluate the performance of different recommendation algorithms, namely precision@N and recall@N (denoted by Pre@N and Rec@N), where N is the number of recommended POIs.

For POI recommendation, given a user u, we compute Pre@N and Rec@N as follows [22]:

$$\text{Pre@N} = \frac{tp_u}{tp_u + fp_u} \quad \text{and} \quad \text{Rec@N} = \frac{tp_u}{tp_u + tn_u} \tag{20}$$

where tp_u is the number of POIs contained in both the ground truth and the top-N results produced by algorithms; fp_u is the number of POIs in the top-N results by algorithms but not in the ground truth; and tn_u is the number of POIs contained in ground truth but not in the top-N results by algorithms. The Pre@N (Rec@N) reported is an average of precision(recall) values of all users [22].

For time-aware POI recommendation, given a user u and a time slot t, we let tp_{ut}, fp_{ut} and tn_{ut} be a time-specific extension of tp_u, fp_u and tn_u, respectively. Then, at time t, the Pre@N(t) and Rec@N(t) are calculated as follows [24]:

$$\text{Pre@N(t)} = \frac{\sum_{u \in \mathcal{U}} tp_{ut}}{\sum_{u \in \mathcal{U}} (tp_{ut} + fp_{ut})} \tag{21}$$

$$\text{Rec@N(t)} = \frac{\sum_{u \in \mathcal{U}} tp_{ut}}{\sum_{u \in \mathcal{U}} (tp_{ut} + tn_{ut})} \tag{22}$$

As in [24, 25], the average Pre@N (Rec@N) is then reported by averaging the precision (recall) values of all time slots. For both metrics, we consider N=5, 10 and 20 in our experiments respectively (by default N=5), as top recommendations are more important.

4.1.3 Baseline methods

For POI recommendation, we compare our model with the following baseline methods.

- **UCF**: This is user-based CF method, where user-user similarity is calculated based on the check-in data.
- **UCF+G**: This method incorporates the geographical influence into user-based CF in a linear interpolation way [22], which is a representative method for POI recommendation.
- **PMF**: Probabilistical matrix factorization [18] is a well-known factorization developed for recommendation systems.
- **BPR-MF**: As our method is ranking-based factorization, we also consider BPR-MF as a baseline, which is the most popular ranking-based matrix factorization with Bayesian Personalized Ranking criterion [16]. Note that this method has not been evaluated in previous work on POI recommendation.
- **GTBNMF**: As mentioned in Section 2, GTBNMF is a recent method for POI recommendation [12], which combines Bayesian matrix factorization with the topic model. In our data, no text information is available and thus we only use the factorization part of this model to compare.
- **GeoMF**: GeoMF is the state-of-the-art method for POI recommendation [11].

For time-aware POI recommendation, the following methods are used as baselines.

- **UCF(+G)**: These are UCF and UCF+G described above. Both of them do not make use of temporal information and hence produce the same recommendations for all the time slots.
- **UTF**: UTF is a user-based temporal collaborative filtering method [5], which computes the similarity between users by weighting the check-ins with a time decay function.
- **UCLAF**: UCLAF is a PARAFAC-based tensor decomposition model, originally proposed to recommend locations and activities with user-location-activity tensor data [27], where Laplician regularization terms are imposed on decompositions to incorporate extra information. We apply this model to the tensor $\mathcal{X}$, and use the matrices $\mathbf{W}$ and $\mathbf{M}$ to construct Laplician regularization terms for incorporating geographical influence and temporal influence, respectively.
- **PITF**: This is a ranking-based tensor factorization method [17], where the Bayesian Personalized Ranking criterion is employed.
- **LRT**: This is a recently developed matrix factorization method for POI recommendation with time information [7]. LRT incorporates temporal influence by constraining the latent factors of a user to be similar in two consecutive time slots.
- **UTE+SE**: As introduced in Section 2, this method utilizes both the geographical influence and temporal influence for time-aware POI recommendation [24]. The model is denoted by UTE when considering temporal influence only, otherwise it is denoted by UTE+SE.
- **BPP**: This is the state-of-the-art method for time-aware POI recommendation [25], which incorporates both the geographical and temporal influences.

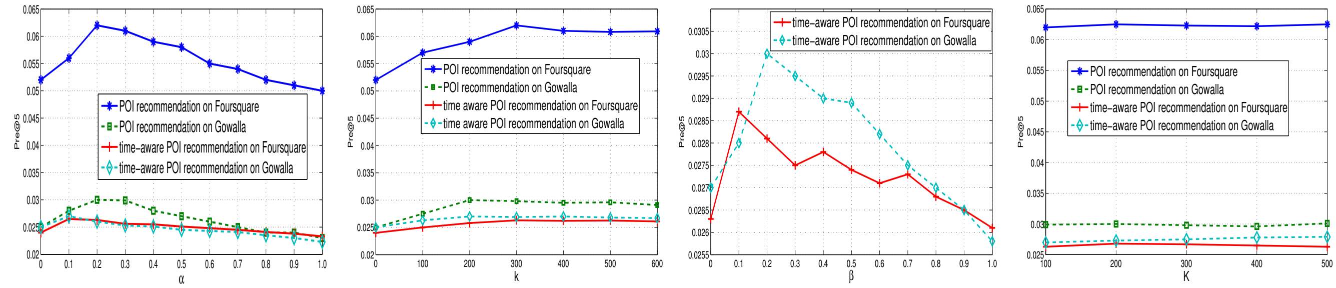
Among these methods, UTF, PITF and LRT do not exploit the geographical influence.

4.2 Experimental Results

4.2.1 Parameter Tuning

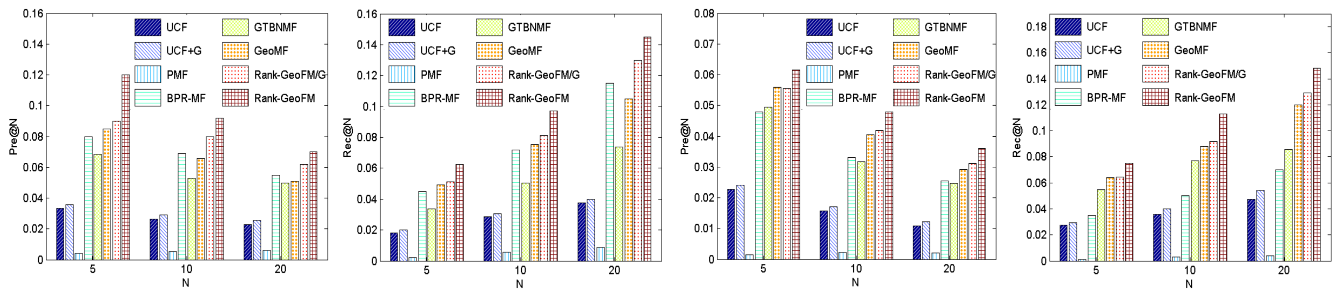
In the experiments, we set the hyperparameters $\varepsilon = 0.3$ and $C = 1.0$ for all the data sets. For the learning rate γ, we set a small value 0.0001 in our experiments to ensure the generalization accuracy. For other parameters, we tune them based on the validation set to find the optimal values, and subsequently use them in the test set.

Figure 2(a) shows the performance of Rank-GeoFM under both settings on both data sets as we vary parameter α for geographical influence. We find that Rank-GeoFM perfoms the best at $\alpha = 0.2$ for POI recommendation on both data, and performs the best at $\alpha = 0.1$ for time-aware POI recommendation on both data. Figure 2(b) shows the performance of Rank-GeoFM as we vary parameter k used in the construction of geographical influence matrix. We can see that the best performance is achieved at $k = 300$ for all cases. Figure 2(c) demonstrates the effect of parameter β,



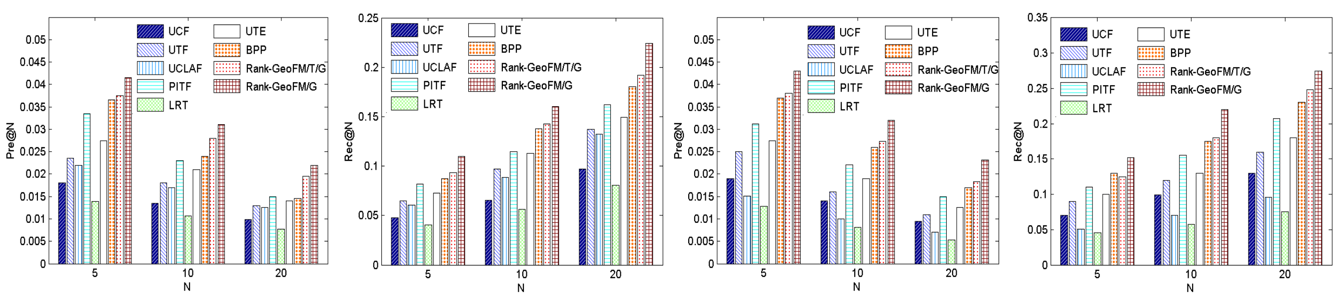
(a) Tuning α (b) Tuning k (c) Tuning β (d) Tuning K

Figure 2: Parameter tuning for Rank-GeoFM.



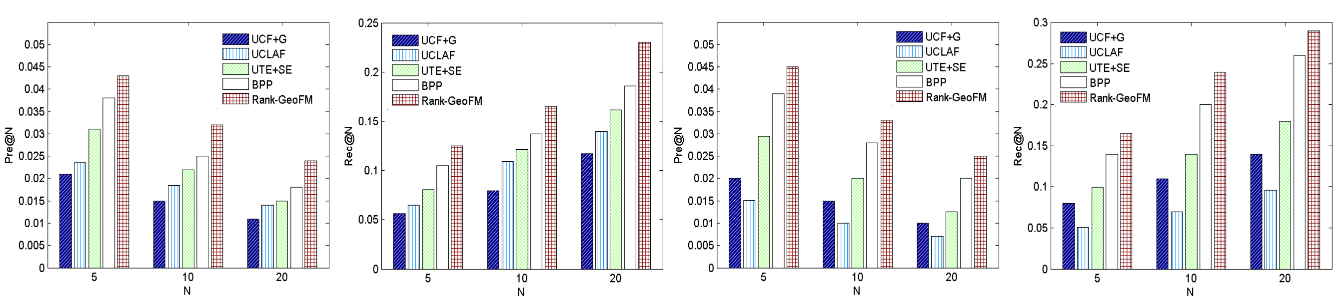
(a) Pre@N-Foursquare (b) Rec@N-Foursquare (c) Pre@N-Gowalla (d) Rec@N-Gowalla

Figure 3: Performance comparison on POI recommendation.



(a) Pre@N-Foursquare (b) Rec@N-Foursquare (c) Pre@N-Gowalla (d) Rec@N-Gowalla

Figure 4: Performance comparison on time-aware POI recommendation without geographical influence.



(a) Pre@N-Foursquare (b) Rec@N-Foursquare (c) Pre@N-Gowalla (d) Rec@N-Gowalla

Figure 5: Performance comparison on time-aware POI recommendation with geographical influence.

which is for temporal influence under time-aware POI recommendation. We can see that the best performance are produced at $\beta = 0.1$ and $\beta = 0.2$ on Foursquare and Gowalla, respectively. Finally, we show the effect of dimension K of latent factors on performance in Figure 2(d). We find that the performance of Rank-GeoFM is insensitive to the dimension K, and we use $K = 100$ in our experiments.

4.2.2 Results on POI Recommendation

Figures 3(a)–(d) show the performance of all the methods on both datasets for POI recommendation. First, we can see that memory-based methods UCF and UCF+G perform worse than the other methods except for PMF, which are all factorization based methods, and UCF+G improves UCF due to the consideration of geographical influence. PMF performs the worst because it is developed for explicit feedback data such as user-movie ratings. It is not suitable for POI recommendation, where check-ins are implicit feedbacks. This is in accordance with the finding in [14].

Among the other factorization based methods, the performance of BPR-MF is very promising, although this method cannot utilize the geographical influence, and has not been employed for POI recommendation in previous work. The reason is that BPR-MF, as a ranking-based factorization method, is more appropriate for handling implicit feedback data. We observe GeoMF performs better than GTBNMF. This is because GTBNMF conducts the factorization by fitting the zero and nonzero entries in $\mathbf{X}$ equally, while GeoMF solves the factorization by assigning higher weights to nonzero entries, which is more suitable for implicit data.

We can see that the proposed Rank-GeoFM consistently outperforms the state-of-the art method GeoMF. The improvements, in terms of Pre@5, are more than 41.6% and 10% on Foursquare and Gowalla datasets, respectively. The reason is that GeoMF addresses the data sparsity problem by fitting both nonzero and zero check-ins with different weights, which is less reasonable than our ranking methodology because zero check-ins may be missing values and should not be fitted directly. Rank-GeoFM/G denotes our model without considering the geographical influence. It can be seen that Rank-GeoFM/G performs better than BPR-MF, a ranking based MF method without considering geographical influence. This may be attributed to our ranking objective function. Moreover, we observe Rank-GeoFM improves Rank-GeoFM/G by 30.3% and 10.5% in terms of Pre@5 on both datasets, respectively, due to the incorporation of geographical influence.

Finally, we note that all improvements of our method over baselines are statistically significant in terms of paired t-test with p-value < 0.01.

4.2.3 Results on Time-aware POI Recommendation

Figures 4(a)–(d) show the performance of all the methods for time-aware POI recommendation when the geographical influence is not utilized. Among all memory-based methods, UCF performs worse than UTF and UTE because it does not capture the correlations between time slots. Although both UTF and UTE consider the correlations between time slots, UTF still performs worse than UTE because UTE smooths the check-in data for handling data sparsity issue.

Among the factorization methods UCLAF, PITF and LRT, LRT performs the worst because it obtains the factorization by fitting the nonzero entries in tensor $\mathcal{X}$, which suffers

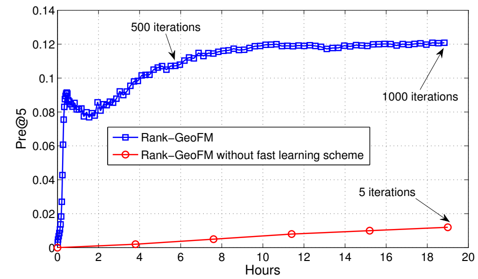


Figure 6: Learning rate comparison of Rank-GeoFM with and without fast learning. Each square denotes ten iterations and each circle denotes one. Each iteration comprises the updates for all user-POI pairs.

from the sparsity problem significantly. The performance of UCLAF is also not good because the method fits both zero and nonzero entries in $\mathcal{X}$, which is not very reasonable. PITF performs the best among the three and its performance is also very promising compared to other methods. The reason is that PITF is a ranking-based factorization method that can alleviate the sparsity problem. Moreover, we observe that the graph-based method BPP, which is the state-of-the-art method, outperforms all the memory-based and factorization based baseline methods.

Rank-GeoFM/G denotes our model with temporal influence but without geographical influence. We observe our model consistently outperforms the state-of-the-art method BPP, which is 13.7% and 16.2% better than BPP, in terms of Pre@5, on Foursquare and Gowalla, respectively. Rank-GeoFM/T/G denotes our model without considering both the temporal and geographical influences. We observe Rank-GeoFM/T/G always outperforms another ranking-based method PITF. Moreover, due to the incorporation of temporal influence, we observe Rank-GeoFM/G improves Rank-GeoFM/T/G by 10.6% and 13.1% in terms of Pre@5 on both data, respectively.

Figures 5(a)–(d) show the performance of the methods incorporating the geographical influence further. Note that we report the results only for the methods that can utilize geographical influence. It shows that Rank-GeoFM outperforms all the other methods by 13.1% and 15.3%, in terms of Pre@5, on Foursquare and Gowalla, respectively.

We note that all improvements of our method over baselines are statistically significant in terms of paired t-test with p-value < 0.01.

4.2.4 Comparison of Rank-GeoFM with and without Fast Learning Scheme

We compare the learning rate of Rank-GeoFM with and without the fast learning scheme introduced in Section 3.2.2. Figure 6 shows the result, where Foursquare data is taken as an example for the comparison. Rank-GeoFM without fast learning scheme means we use Eq. (9) to calculate gradients and perform updates in Algorithm 1. We observe Rank-GeoFM (with fast learning scheme) finishes 1000 iterations of updates in 19 hours while the counterpart without fast learning scheme finishes only 5 iterations. Rank-GeoFM obtains a well-trained model within 8 hours, which leads to the best performance, while the counterpart cannot build an acceptable model after 19 hours of training. This result demonstrates the efficiency as well as necessity of using fast learning scheme. Moreover, we find that the first 500 iterations of Rank-GeoFM take 5.5 hours, while the next 500 iterations take 13.5 hours. The reason is that at the start of training, POIs are not well-ranked by our model and thus it takes less time to sample an incorrectly-ranked POI; however, as the training process goes on, the ranking becomes better and we need more time to sample an incorrect one. Another interesting observation is that we find there is a dip in precision after around 100 iterations for Rank-GeoFM. The observation may be because our learning procedure is transiting from coarse search to fine tuning, i.e., $E\left(\left\lfloor \frac{|\mathcal{L}|}{n} \right\rfloor\right)$ (or the magnitude of gradient) is decreasing from a large value to a small one.

5. CONCLUSIONS

In this paper, we propose a ranking based factorization method, Rank-GeoFM, for POI recommendation. In the proposed model, we learn the factorization by fitting the user's preference rankings for POIs, which alleviates the data sparsity problem. Extensive experimental results on both POI recommendation and time-aware POI recommendation show that Rank-GeoFM outperforms the state-of-the-art methods significantly.

Rank-GeoFM is very flexible to incorporate context information. In the future, it would be interesting to investigate with Rank-GeoFM how the other context information impacts the performance of POI recommendation. For example, if we know the category of each POI ℓ is $cat(\ell)$, the recommendation score in Eq. (4) can be modified as follows to incorporate this information:

$$y_{u\ell} = \mathbf{u}_u^{(1)} \cdot \mathbf{l}_\ell^{(1)} + \mathbf{u}_u^{(2)} \cdot \sum_{\ell^* \in \mathcal{N}_k(\ell)} w_{\ell\ell^*} \mathbf{l}_{\ell^*}^{(1)} + \mathbf{u}_u^{(3)} \cdot \mathbf{c}_{cat(\ell)}$$

where $\mathbf{c}_i$ is the latent factor of category i and the last term in this equation denotes the interaction score between category $cat(\ell)$ and user u. We will leave this as our future work.

6. ACKNOWLEDGEMENT

This work is supported in part by a grant awarded by a Singapore MOE AcRF Tier 2 Grant (ARC30/12), a Singapore MOE AcRF Tier 1 Grant (RG66/12), and a grant awarded by Microsoft Research Asia.

7. REFERENCES

[1] J. Bao, Y. Zheng, D. Wilkie, and M. F. Mokbel. A survey on recommendations in location-based social networks. *GeoInformatica*, 2014.

[2] R. M. Bell and Y. Koren. Lessons from the Netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[3] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, 2012.

[4] C. Cheng, H. Yang, M. R. Lyu, and I. King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, pages 2605–2611, 2013.

[5] Y. Ding and X. Li. Time weight collaborative filtering. In *CIKM*, pages 485–492. ACM, 2005.

[6] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan. Personalized ranking metric embedding for next new POI recommendation. In *IJCAI*, 2015.

[7] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Recsys*, pages 93–100. ACM, 2013.

[8] H. Gao, J. Tang, X. Hu, and H. Liu. Content-aware point of interest recommendation on location-based social networks. In *AAAI*, 2015.

[9] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. LARS: A location-aware recommender system. In *ICDE*, pages 450–461. IEEE, 2012.

[10] Y. Li, J. Nie, Y. Zhang, B. Wang, B. Yan, and F. Weng. Contextual recommendation based on text mining. In *COLING*, pages 692–700. ACL, 2010.

[11] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *SIGKDD*, pages 831–840. ACM, 2014.

[12] B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning geographical preferences for point-of-interest recommendation. In *SIGKDD*, pages 1043–1051. ACM, 2013.

[13] X. Liu, Y. Liu, K. Aberer, and C. Miao. Personalized point-of-interest recommendation by mining users' preference transition. In *CIKM*, pages 733–738. ACM, 2013.

[14] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. A random walk around the city: New venue recommendation in location-based social networks. In *PASSAT*, pages 144–153. IEEE, 2012.

[15] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in Foursquare. *ICWSM*, 11:70–573, 2011.

[16] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press, 2009.

[17] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, pages 81–90. ACM, 2010.

[18] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[19] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver. TFMAP: optimizing map for top-N context-aware recommendation. In *SIGIR*, pages 155–164. ACM, 2012.

[20] N. Usunier, D. Buffoni, and P. Gallinari. Ranking with ordered weighted pairwise classification. In *ICML*, pages 1057–1064. ACM, 2009.

[21] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine Learning*, 81(1):21–35, 2010.

[22] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334. ACM, 2011.

[23] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: A location-content-aware recommender system. In *SIGKDD*, pages 221–229. ACM, 2013.

[24] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *SIGIR*, pages 363–372. ACM, 2013.

[25] Q. Yuan, G. Cong, and A. Sun. Graph-based point-of-interest recommendation with geographical and temporal influences. In *CIKM*, pages 659–668. ACM, 2014.

[26] K. Zhao, G. Cong, Q. Yuan, and K. Zhu. SAR: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews. In *ICDE*, 2015.

[27] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *AAAI*, pages 236–241, 2010.