Point-of-Interest Recommendations: Learning Potential Check-ins from Friends

Huayu Li*, Yong Ge+ , Richang Hong– and Hengshu Zhu×
*The University of North Carolina at Charlotte, + University of Arizona,
– Hefei University of Technology, ×Baidu Research-Big Data Lab
hli38@uncc.edu, ygestrive@gmail.com, hongrc@hfut.edu.cn, zhuhengshu@baidu.com

ABSTRACT

The emergence of Location-based Social Network (LBSN) services provides a wonderful opportunity to build personalized Point-of-Interest (POI) recommender systems. Although a personalized POI recommender system can significantly facilitate users' outdoor activities, it faces many challenging problems, such as the hardness to model user's POI decision making process and the difficulty to address data sparsity and user/location cold-start problem. To cope with these challenges, we define three types of friends (i.e., social friends, location friends, and neighboring friends) in LBSN, and develop a two-step framework to leverage the information of friends to improve POI recommendation accuracy and address cold-start problem. Specifically, we first propose to learn a set of potential locations that each individual's friends have checked-in before and this individual is most interested in. Then we incorporate three types of check-ins (i.e., observed check-ins, potential check-ins and other unobserved check-ins) into matrix factorization model using two different loss functions (i.e., the square error based loss and the ranking error based loss). To evaluate the proposed model, we conduct extensive experiments with many state-of-the-art baseline methods and evaluation metrics on two real-world data sets. The experimental results demonstrate the effectiveness of our methods.

Keywords

Point-of-Interest; Recommendation; Matrix Factorization

1. INTRODUCTION

Recent years have witnessed the prevalence of smart mobile devices and the convenience of accessing wireless network, which makes people much easier to acquire their real-time location information. This development stimulates the emergence of location-based social network (LBSN) services such as Foursquare, Jiepang, and Facebook Places. These LBSNs allow users to build connections with each other, and share their experience and check-in information associated with a Point-of-Interest (POI). A variety of such user interaction data with LBSNs provide a good opportunity for developing personalized POI recommender systems. Indeed, the accurate and personalized POI recommendation is a crucial demand in LBSN services. It not only helps users to explore new locations, but also facilitates users to find relevant POIs without spending too much time on searching, particularly when they are in a new region.

Although developing a personalized POI recommender system is a crucial task and could benefit users' outdoor activities, it is still a very challenging problem due to three reasons. First, a user's check-in decision making process is very complex and could be influenced by many different kinds of factors. For example, it is difficult to model the influence of social friends on user's check-in behaviours. We do not know which friend will actually influence user's POI decision, not mention to know how she affects user's choice. Also, the geographical distance might affect user's POI decision. A user often prefers a nearby POI to another one far away. Second, POI recommender system usually suffers a severe challenge caused by extreme sparse check-in data. In real system, there are over millions of POIs. However, a single user usually checks-in a limited number of POIs, which significantly increases the difficulty of recommendation. Third, when a new POI or a new user enters the system and we do not have its visitor information or her historical check-in information, it is very difficult to recommend new POI to users or recommend POIs to new user.

In the literature, some related works have been proposed to incorporate social network into POI recommendations. For example, [16] placed a social regularization term to constrain the estimation of user feature vectors with the assumption that friends will share the similar interests. Meanwhile, some researchers also proposed to take the geographical influence into account to assist POI recommendation. For instance, [28] leveraged a linear model to combine user interest, social network and geographical distance for POI prediction. On the other hand, [15] modeled the geographical neighborhood influence in both instance and region level. In instance level, one user's preference for a location is predicted as a combination of her special preference on this location and the nearest neighborhoods of this location. In region level, it places a group lasso penalty to learn location-specific latent vectors. However, few of these models incorporate the influence of geographically close users on each other's check-in activities into matrix factorization. The geographically close users may share similar interests and should have potential influence on check-in behaviors [23].

KDD '16, August 13-17, 2016, San Francisco, CA, USA

 ISBN 978-1-4503-4232-2/16/08…$15.00
DOI: http://dx.doi.org/10.1145/2939672.2939767

Moreover, few of these models could address user cold-start problem in location recommendation. Motivated by these, we first formally define three types of friends for each user: social friends, location friends and neighboring friends. The social friends of a user refer to the set of users who are socially connected with this user in LBSNs. The location friends of a user denote the set of users who check-in the same locations as this user does. The neighboring friends of a user are those users who are geographically close to this user. Then we novelly incorporate their historical check-ins into matrix factorization model with different loss functions.

Through our analysis on two real-world data sets, we find that users share the similar interests with their three types of friends. Consequently, we propose a two-step framework to elaborate friends' check-ins. In the first step, we design two approaches (i.e., a linear aggregation based and a random walk based) to learn a set of friends' locations that each user most potentially prefers and she never visited. Thus, a user's check-ins are divided into observed check-ins, potential check-ins and other unobserved check-ins. In the second step, we develop two loss functions to model these three kinds of check-ins: the square error based loss function and the ranking error based loss function. Specifically, the square error based loss treats user's check-ins as an indication of positive, potential and negative preference with varying confidence. The ranking error based loss assumes that the user prefers an observed location over any potential locations, and also prefers a potential location over any unobserved locations. We extensively evaluate our models with many state-of-the-art baseline models and different validation metrics on two real-world data sets. The experimental results not only demonstrate the improvements of our models on POI recommendation, but also show the effectiveness for cold-start problem.

To summarize, the major contributions of this paper are:

- empirically analyzes the correlations between users and their three type of friends using two check-in datasets;
- designs two approaches to learn a set of locations for each individual user that her friends have checked-in before and she is most interested in;
- develops matrix factorization based models via different error loss functions with the learned potential check-ins, and correspondingly proposes two scalable optimization methods;
- designs three different recommendation strategies for standard recommendation, new location recommendation, and new user recommendation.

2. PRELIMINARIES

In this section, we first introduce some mathematical notions, and then provide the definition of friends. At last, we introduce the recommendation framework.

2.1 Notation

Suppose there are N users and M locations. For convenience we will henceforth refer to i as user, f as friend and j as the location unless stated otherwise. Suppose there are C kinds of categories and the category of location j is denoted as c_j. For user i, $\mathcal{F}_i$ denotes a set of friends which will be further defined and explained in Section 2.2, $\mathcal{M}_i^o$ is a set of locations checked-in by her, $\mathcal{M}_i^p$ is a set of potential locations learned in Section 3, and $\mathcal{M}_i^u$ is the remaining unvisisted locations. r_{ij} is the check-in frequency of user i on location j. In addition, all column vectors are represented by bold lower case letters, all matrices are represented by bold upper case letters, and a numeric value is denoted by lower case letter. A predicted value is denoted with a ˆ (hat) over it. The terms *location* and *POI* are used interchangeably.

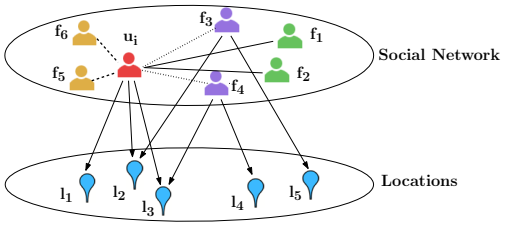


Figure 1: The user u_i's social network and check-ins.

2.2 Definition

To better understand users' check-in behaviours, we examine the check-in data collected from Gowalla and Foursquare (Details can be found in Section 5). To clarify the relation between the similarity of pairwise users and their physical distance, we plot their relations in Figure 2(a) and Figure 2(b). The physical distance of two users refers to the distance between their home locations, and the similarity of user i and user f is measured by cosine similarity, given by:

$$Sim_u(i,f) = \Big(\sum_{j\in\mathcal{M}_i^o} r_{ij}^2 \sum_{j\in\mathcal{M}_f^o} r_{fj}^2\Big)^{-\frac{1}{2}} \sum_{j\in\mathcal{M}_i^o\cap\mathcal{M}_f^o} r_{ij}r_{fj}. \quad (1)$$

Based on the observation, we find that the physically closer two users live, the more similar their POI interests are. It motivates us to leverage neighboring friends, who are physically neighbors, to learn user's interest in POIs. In addition, social friends who build connections online share the similar interests in POI decisions [16, 1, 5, 22]. Users who check-in similar locations are treated as location friends, and also might have similar tastes. Thus, three types of friends of user i, i.e., neighboring friends, social friends and location friends, might affect her check-in activity, defined as:

Definition 1 (Social Friends). *The social friends of user i are the set of users who have socially connected with her in LBSNs, which is denoted as $\mathcal{F}_i^s$.*

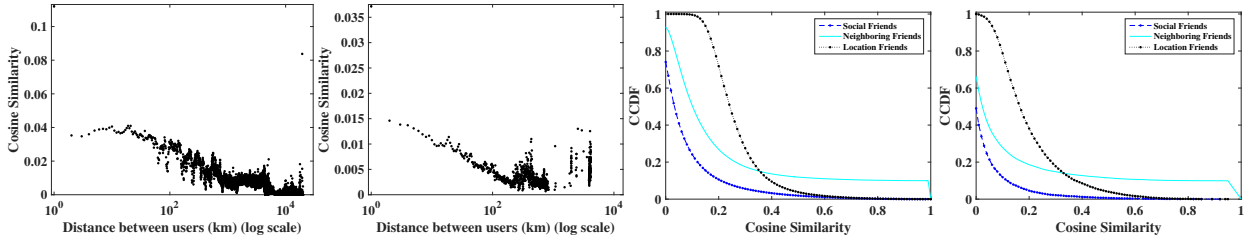
Definition 2 (Location Friends). *Given a set of locations $\mathcal{M}_i^o$, which have been checked-in by user i, her location friends, denoted as $\mathcal{F}_i^l$, are the set of users who have also checked-in these locations, i.e., $\mathcal{F}_i^l = \bigcup_{j\in\mathcal{M}_i^o}\Psi_j$, where Ψ_j is the set of users who also have checked-in location j.*

Definition 3 (Neighboring Friends). *Given the home location of user i, the neighboring friends are the set of users who live physically closest to her and denoted as $\mathcal{F}_i^n$.*

In the example of Figure 1, the target user u_i has checked-in locations $\{l_1, l_2, l_3\}$. $\{f_1, f_2\}$ are her social friends who socially connect with her online. User f_3 has check-ins at locations l_2 and l_5, and user f_4 has check-ins at locations l_3 and l_4. f_4 and f_5 have common POIs with user u_i, i.e., l_2 and l_3, respectively. Thus, both of them are the location friends of user u_i. In addition, f_5 and f_6 are the target user's neighboring friends due to their physically short distance to her. Thus, $\{f_1, \cdots, f_6\}$ are regarded as the friends of user i. In this paper, the **friends** of user i are defined as:

$$\mathcal{F}_i = \mathcal{F}_i^s \cup \mathcal{S}(\mathcal{F}_i^l) \cup \mathcal{S}(\mathcal{F}_i^n), \quad (2)$$

where $\mathcal{S}(\mathcal{F}_i^l)$ is the set of S most similar friends with the highest cosine similarities and $\mathcal{S}(\mathcal{F}_i^n)$ is the set of S physically nearest friends with the shortest distance among their



(a) Gowalla Data (b) Foursquare Data (c) Gowalla Data (d) Foursquare Data

Figure 2: (a) ∼ (b) Cosine similarity as a function of distance between users' home locations. (c) ∼ (d) Complementary Cumulative Distribution Function (CCDF) of cosine similarity between friends.

homes[1]. To examine the correlation between friends, we report the complementary cumulative distributions of their similarities in Figure 2(c) and Figure 2(d) on Gowalla and Foursquare, respectively. There are over 5%, 20% and 40% pairs of social, neighboring and location friends which have similarities larger than 0.2. Particularly, the friends' correlation is much stronger in Gowalla than Foursquare. The observation shows the importance of friends in LBSNs and motivates us to use friends' historical check-ins to improve recommendation accuracy.

2.3 The Recommendation Framework

The recommendation task in this paper is defined as: given users' historical checked-in locations, we aim at recommending each user with top-K locations that she might be interested in but has not visited before. In this paper, we propose a two-step recommendation framework. Specifically, in the first step, we learn a set of potential locations from three types of friends, which will be introduced in Section 3. In the second step, we incorporate the learned potential locations of each individual into matrix factorization model with different error loss functions, which will be presented in Section 4. At last, we introduce different recommendation strategies for standard recommendation, location cold-start recommendation and user cold-start recommendation.

3. LEARNING POTENTIAL LOCATIONS

Social network plays an important role in recommendations [16, 28, 5, 22]. However, only leveraging the historical locations of social friends cannot successfully model user's preference for locations due to that it is difficult to appropriately model the preference of users who have no social friends, not mention to handle user cold-start problem (i.e., a user has never checked-in any location before). To address these problems, we will exploit the characteristics of three types of friends: social friends, location friends and neighboring friends. The earlier section has shown their significance in LBSNs, i.e., friends would share the similar preferences for POIs. In other words, users might be interested in those locations which have been checked-in by their friends, and have a high probability to check-in them next time. However, the extremely large number of these locations will lead to the inefficiency of computation with the increase of locations, and the inaccuracy of prediction with the increase of noise. Hence, the problem in this section is to find the most potential locations for the target user, defined as:

Definition 4 (Problem of Potential Locations). *Given the set of locations $\mathcal{M}_i^f = \bigcup_{f \in \mathcal{F}_i} \mathcal{M}_f^o \backslash \mathcal{M}_i^o$, that the friends of target user i have checked-in before but she never visits, the problem is to find top S most potential locations that she might be interested, denoted as $\mathcal{M}_i^p$.*

[1]In the experiment, we set S as 10.

To obtain the potential locations of each user i, we propose two methods, i.e., *Linear Aggregation* and *Random Walk*, to estimate the probability p_{ij}^{pot} of this user on each location j that her friends have checked-in. Then we rank them by the estimated probabilities and select S locations with the highest probabilities[2]. The learned potential locations will assist to make accurate recommendation in Section 4.

3.1 Linear Aggregation

In this section, we propose *Linear Aggregation* method, denoted as *LA*, to predict the probability p_{ij}^{pot} that user i prefers location j which has been visited by her friends. Suppose $Sim(i, f; j)$ is the similarity between user i and friend f on the preference for location j. A location is possibly checked-in by more than one friends, so we define p_{ij}^{pot} as:

$$p_{ij}^{pot} \propto \max_{f \in \mathcal{F}_i^j} \{Sim(i, f; j)\},$$

where $\mathcal{F}_i^j$ is the set of user i's friends who have checked-in location j. The similarity $Sim(i, f; j)$ incorporates two parts: (1) the similarity of user interest, and (2) the similarity of geographical location. The similarity of user interest can be measured by cosine similarity in Eq.(1). Since a user's check-in probability and the distance from her home to the corresponding location follow a power law distribution [9], we exploit this characteristic to model geographical similarity. Hence, we define the probability that a user checks-in a location d-km far away as the following:

$$Pr_G(d) = a \cdot d^b, \tag{3}$$

where a and b are the parameters of power law distribution and could be learned by maximum likelihood estimation. Then the probability of user i to check-in a POI j due to the geographical influence is normalized as:

$$p_{ij}^G = \frac{Pr_G(d(h_i, j))}{Pr_G(d_{min})}, \tag{4}$$

where h_i is the home location of user i, and $d(h_i, j)$ indicates the distance between the home location of user i and the POI j, and d_{min} is the minimum distance. The distance could be computed by Haversine formula with latitude and longitude. Thus, $Sim(i, f; j)$ is the linear aggregation of similarities on both user interest and geographical location, given by

$$Sim(i, f; j) = \zeta \; Sim_u(i, f) + (1 - \zeta) \; p_{ij}^G,$$

where $\zeta \in [0, 1]$ is a tuning parameter to control the importance of the similarity of user interest.

3.2 Random Walk

Random walk with restart has successfully measured the correlation between two nodes in a graph [7, 24]. In this section, we propose a *Random Walk* method, denoted as *RW*, to learn the probability p_{ij}^{pot} of user i on location j which has been visited by her friends. We construct a directed

[2]In the experiments, we set S as 500.

graph with two kinds of nodes: the users (i.e., the target user and her friends), and the locations checked-in by her and her friends. Let $\mathbf{y}$ be a column vector where y_i refers to the probability that the random walk is at node i. Also let $\mathbf{A}$ be the column normalized transition matrix where a_{ij} denotes the probability that node i jumps to node j. Here we consider three types of transition probabilities: (1) the probability between users measured by the cosine similarity in Eq.(1); (2) the probability from each user to each location which is one if the user checks-in the corresponding location and otherwise is zero; (3) the similarity between a pair of locations (j and k) measured by the normalized power-law function which is defined as the following:

$$Sim_G(j,k) = \frac{Pr_G(d(j,k))}{Pr_G(d_{min})}, \tag{5}$$

where $d(j,k)$ is the distance between these two locations and the power law parameters are learned with the check-in probabilities and corresponding distances of pairwise locations. Hence, the iteration equation for updating the steady-state probability of each node is given as follows:

$$\mathbf{y} = (1-\beta)\mathbf{A}\mathbf{y} + \frac{\beta}{|\mathcal{M}_i^o \cap \mathcal{M}_i^f| + |\mathcal{F}_i| + 1}\mathbf{x}, \tag{6}$$

where $\mathbf{x}$ is the column vector of zeros with the elements corresponding to the target user and her checked-in locations as one, and $\beta \in [0,1]$ is the restart probability to return to the target user and her checked-in locations. The steady-state probability is achieved by recursively performing Eq.(6) until convergence. Thus, the probability p_{ij}^{pot} is the steady-state probability corresponding to the location j.

4. RECOMMENDATION MODELS

In Section 3, for each individual user, we have learned the potential locations from her friends' information. In this section, the learned potential locations are utilized to make accurate recommendation and address user cold-start problem. Overall, for each user i, we have her three kinds of locations: observed locations $\mathcal{M}_i^o$, potential locations $\mathcal{M}_i^p$ and other unobserved locations $\mathcal{M}_i^u$.

In this paper, we build our recommendation models by leveraging the widely-used matrix factorization techniques [20, 19, 8, 6, 11], where both user and location are mapped into latent low-dimension spaces. Let $\mathbf{U} \in \mathbb{R}^{K\times N}$ and $\mathbf{V} \in \mathbb{R}^{K\times M}$ be the latent user and location feature matrices, with column vectors $\mathbf{u}_i$ and $\mathbf{v}_j$ representing the K-dimensional user-specific and location-specific feature vectors of user i and location j, respectively. A typical prediction for the preference of user i to location j is taken by an inner product of latent vectors, i.e., $\hat{p}_{ij} = \mathbf{u}_i^T\mathbf{v}_j$, where $\mathbf{P} \in \mathbb{R}^{N\times M}$ is the preference matrix.

However, in LBSNs the category information of POIs affects user's check-in decision making process. Users are often used to visiting those POIs which belong to the same category due to their specific hobbies. For example, users who like eating would have a much higher probability to choose a new POI relevant to *food* next time, but they have much less chance to check-in a POI about *sight*. Thus, the preference of category is another important factor to affect user's decision on a new POI. Here, we introduce the category feature matrix $\mathbf{Q} \in \mathbb{R}^{N\times C}$, where each entry q_{ic} indicates the preference of user i to category c. Hence, the preference of user i for location j is refined as follows:

$$\hat{p}_{ij} = (q_{ic_j} + \varepsilon)\mathbf{u}_i^T\mathbf{v}_j, \tag{7}$$

where c_j is the category of location j and ε is a tuning parameter to indicate that user has a small probability to prefer one location with another category.

Many of the recent works suppose to only model the observed rating, which is adapted to explicit feedback datasets. However, the check-in dataset is implicit feedback dataset, where we do not have explicit feedback for user's preference to locations. In other words, we lack substantial evidence on which location the user dislikes. To address user cold start problem and data sparseness problem, we propose to model the observed preference, potential preference and unobserved preference of users for location, simultaneously. Let j, k, h denote the observed location, potential location and unobserved location, respectively. The loss function of general form is given as follows:

$$\underset{\mathbf{U},\mathbf{V},\mathbf{Q}}{\operatorname{argmin}} \sum_i E_i(p_{ij}, p_{ik}, p_{ih}, \hat{p}_{ij}, \hat{p}_{ik}, \hat{p}_{ih}) + \Theta(\mathbf{U},\mathbf{V},\mathbf{Q}), \tag{8}$$
$$\forall j \in \mathcal{M}_i^o, \forall k \in \mathcal{M}_i^p, \forall h \in \mathcal{M}_i^u,$$

where $E_i(\cdot)$ is the loss function for the observed, potential and unobserved preference of user i for locations, and $\Theta(\cdot)$ is a regularization term with ℓ_2 norm which is defined as:

$$\Theta(\mathbf{U},\mathbf{V},\mathbf{Q}) = \frac{\lambda_u}{2}||\mathbf{U}||_2^2 + \frac{\lambda_v}{2}||\mathbf{V}||_2^2 + \frac{\lambda_q}{2}||\mathbf{Q}||_2^2, \tag{9}$$

where λ_u, λ_v and λ_q are the regularization constants. We develop two different types of models which use different loss function for $E_i(\cdot)$, i.e., the square error based and the ranking error based loss functions, and will be described in the next two sections, respectively.

4.1 The Square Error based Model

In this section, we present the Augmented Square error based Matrix Factorization (*ASMF*) model constrained with the square error loss function and its optimization method.

4.1.1 The ASMF Model

Due to the similar interests between friends, one user might have opportunity to visit those potential locations that her friends have visited before but she never checks-in. We treat each individual user's check-ins as an indication of positive, potential and negative preference associated with different confidence. One user has a high confidence for the positive preference to their checked-in POIs. However, she will have a low confidence for the potential preference to those potential locations and the negative preference to other unvisited locations. Correspondingly, we augment the binary preference variable p_{ij} to a ternary value as follows:

$$p_{ij} = \begin{cases} 1 & \text{if } j \in \mathcal{M}_i^o \\ \alpha & \text{if } j \in \mathcal{M}_i^p \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $\alpha \in [0,1]$ is a potential preference constant, indicating user i has a probability α to choose an unvisited location j that her friends have visited before.

Therefore, we propose the augmented square error based matrix factorization model (*ASMF*) to compute the loss $E_i(\cdot)$ by using the squared error loss function with the ternary variable defined in Eq.(10), given by:

$$E_i(\cdot) = \sum_{j=1}^{M} w_{ij}\left(p_{ij} - \hat{p}_{ij}\right)^2, \tag{11}$$

where $\mathbf{W}$ is the confidential matrix with element w_{ij} as the confidential weight for user i to location j, given by:

$$w_{ij} = \begin{cases} 1 + \gamma \times r_{ij} & \text{if } j \in \mathcal{M}_i^o \\ 1 & \text{otherwise,} \end{cases} \tag{12}$$

where γ is the tuning parameter.

4.1.2 The Parameter Estimation

In *ASMF* model, based on the Eq.(7), Eq.(8) and Eq.(11), the matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Q}$ are learned by minimizing the following regularized optimization problem:

$$\mathcal{L} = \min_{\mathbf{U},\mathbf{V},\mathbf{Q}} \sum_{i=1}^{N}\sum_{j=1}^{M} w_{ij}\left(p_{ij} - (q_{ic_j}+\varepsilon)\mathbf{u}_i^T\mathbf{v}_j\right)^2 + \Theta(\mathbf{U},\mathbf{V},\mathbf{Q}), \quad (13)$$

To solve the above optimization problem, we adopt Alternating Least Squares (ALS) [8] optimization method due to the accurate parameter estimation and fast convergence rate. We perform ALS method to compute each latent variable by fixing the other variables when minimizing the objective function. The updating formulas with respect to $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{Q}$ are given as follows:

$$\mathbf{u}_i = (\lambda_u \mathbf{I}_K + \sum_j w_{ij}\tilde{q}_{ic_j}^2 \mathbf{v}_j\mathbf{v}_j^T)^{-1}\sum_j \mathbf{w}_{ij}\tilde{q}_{ic_j}p_{ij}\mathbf{v}_j, \quad (14)$$

$$\mathbf{v}_j = (\lambda_v \mathbf{I}_K + \sum_i w_{ij}\tilde{q}_{ic_j}^2 \mathbf{u}_i\mathbf{u}_i^T)^{-1}\sum_i w_{ij}\tilde{q}_{ic_j}p_{ij}\mathbf{u}_i, \quad (15)$$

$$\mathbf{q}_{ic} = (\sum_{j\in\mathcal{N}_c} w_{ij}(p_{ij}-\varepsilon)\mathbf{u}_i^T\mathbf{v}_j)/(\lambda_q + \sum_{j\in\mathcal{N}_c} w_{ij}(\mathbf{u}_i^T\mathbf{v}_j)^2), \quad (16)$$

where $\mathbf{I}_K$ is the K-dimension unit matrix, $\mathcal{N}_c$ is the set of locations with category c, and $\tilde{q}_{ic_j}$ is equal to $q_{ic_j}+\varepsilon$. The detailed algorithm is reported in Algorithm 1. Specifically, we place the non-negative constraints on $\mathbf{Q}$ and project the negative variables to 0 in each iteration.

Algorithm 1: ASMF Optimization

Input: $\mathbf{W}$, $\mathbf{P}$, λ_u, λ_v, λ_q, α, ε, τ, $maxIter$
Output: $\mathbf{U}^{(t)}$, $\mathbf{V}^{(t)}$, $\mathbf{Q}^{(t)}$

```
Randomly initialize U^(0) and V^(0), t ← 1, ω ← ∞
Initialize Q^(0) by using Eq.(16)
while t ⩽ maxIter && ω ⩾ τ do
    Update U^(t) by using Eq.(14)
    Update V^(t) by using Eq.(15)
    Update Q^(t) by using Eq.(16)
    ω ← |L(U^(t),V^(t),Q^(t)) − L(U^(t−1),V^(t−1),Q^(t−1))| / |L(U^(t−1),V^(t−1),Q^(t−1))|
    t ← t + 1
end
return U^(t), V^(t), Q^(t)
```

Complexity Analysis. The complexity of direct computation is $\mathcal{O}(NMK^2)$ which is extremely inefficient particularly with the increase of locations and users. To improve the efficiency, we design the following updating strategies. For updating $\mathbf{u}_i$, we employ the similar trick in [6], i.e., $\sum_j w_{ij}\tilde{q}_{ic_j}^2\mathbf{v}_j\mathbf{v}_j^T = \sum_j \tilde{q}_{ic_j}^2\mathbf{v}_j\mathbf{v}_j^T + \sum_j \gamma r_{ij}\tilde{q}_{ic_j}^2\mathbf{v}_j\mathbf{v}_j^T$. The first term can be written as $\sum_j \tilde{q}_{ic_j}^2\mathbf{v}_j\mathbf{v}_j^T = \sum_c \tilde{q}_{ic}^2\sum_{j\in\mathcal{N}_c}\mathbf{v}_j\mathbf{v}_j^T$. For each category c, $\sum_{j\in\mathcal{N}_c}\mathbf{v}_j\mathbf{v}_j^T$ is independent of i and already pre-computed, so the time complexity of this term is $\mathcal{O}(CK^2)$ and C is usually very small. The cost time of the second term is $\mathcal{O}(n_iK^2)$, where the potential part can be pre-computed and n_i is the number of observed locations for which $r_{ij}>0$. In addition, the inverse of a $K\times K$ matrix costs $\mathcal{O}(K^3)$. Consequently, the re-computation of $\mathbf{u}_i$ is performed in time $\mathcal{O}(CK^2+n_iK^2+K^3)$. This procedure is performed over each user, so the total time is $\mathcal{O}(NCK^2+nK^2+NK^3)$, where n is defined as $n=\sum_i n_i$.

Similarly, when updating $\mathbf{v}_j$, we have $\sum_i w_{ij}\tilde{q}_{ic_j}^2\mathbf{u}_i\mathbf{u}_i^T = \sum_i \tilde{q}_{ic_j}^2\mathbf{u}_i\mathbf{u}_i^T + \sum_i \gamma r_{ij}\tilde{q}_{ic_j}^2\mathbf{u}_i\mathbf{u}_i^T$. For each category c, the first term is independent of j and was already pre-computed. Thus, the total cost time over M locations is $\mathcal{O}(nK^2+MK^3)$.

To update q_{ic}, we can rewrite the crucial expression as $\sum_{j\in\mathcal{N}_c} w_{ij}(\mathbf{u}_i^T\mathbf{v}_j)^2 = \sum_{j\in\mathcal{N}_c}(\mathbf{u}_i^T\mathbf{v}_j)^2 + \sum_{j\in\mathcal{N}_c}\gamma r_{ij}(\mathbf{u}_i^T\mathbf{v}_j)^2$. The first term can be written as $\sum_{j\in\mathcal{N}_c}(\mathbf{u}_i^T\mathbf{v}_j)^2 = \mathbf{u}_i^T(\sum_{j\in\mathcal{N}_c}\mathbf{v}_j\mathbf{v}_j^T)\mathbf{u}_i$, where $\sum_{j\in\mathcal{N}_c}\mathbf{v}_j\mathbf{v}_j^T$ was already pre-computed, so it costs $\mathcal{O}(K^2)$. The second term costs $\mathcal{O}(Kn_{ic})$, where n_{ic} is the number of locations for which $r_{ij}>0$ and belongs to category c. The total complexity of updating $\mathbf{Q}$ is $\mathcal{O}(NCK^2+Kn)$.

In a summary, for each iteration of optimization, the total time is $\mathcal{O}(nK^2)$, where $n > max\{M,N\}K$, and $n > CN$ are usually satisfied. In other words, the time complexity of one optimization iteration is in linear proportion to the number of observed check-ins.

4.2 The Ranking Error based Model

In this section, we present the Augmented Ranking error based Matrix Factorization ($ARMF$) model constrained with the ranking error loss and its optimization method.

4.2.1 The ARMF Model

In check-in dataset, we only have a user's check-in record and do not know how much she dislikes a location. In other words, an unvisited location does not necessarily indicate the user dislikes it. The unobserved data actually is a mixture of negative preference for locations and missing values. It motivates us to consider a ranking error based loss function for modeling the ranking order of user's preference for observed locations, potential locations and unobserved locations. We assume that the user prefers an observed location over all potential locations, and at the same time she prefers a potential location over all other unobserved locations. Thus, for user i, the ranking order of her preference over an observed location $j\in\mathcal{M}_i^o$, a potential location $k\in\mathcal{M}_i^p$ and an unobserved location $h\in\mathcal{M}_i^u$ is given as the following:

$$\begin{cases}\hat{p}_{ij}>\hat{p}_{ik}\\ \hat{p}_{ik}>\hat{p}_{ih}\end{cases} \Rightarrow \begin{cases}(q_{ic_j}+\varepsilon)\mathbf{u}_i^T\mathbf{v}_j > (q_{ic_k}+\varepsilon)\mathbf{u}_i^T\mathbf{v}_k\\ (q_{ic_k}+\varepsilon)\mathbf{u}_i^T\mathbf{v}_k > (q_{ic_h}+\varepsilon)\mathbf{u}_i^T\mathbf{v}_h\end{cases}. \quad (17)$$

To this end, we propose the augmented ranking error based matrix factorization ($ARMF$) to compute the loss $E_i(\cdot)$ by using the ranking error loss function, given by,

$$E_i(\cdot) = -\sum_{j\in\mathcal{M}_i^o}\sum_{k\in\mathcal{M}_i^p}\ln\sigma(\hat{p}_{ij}-\hat{p}_{ik}) - \sum_{k\in\mathcal{M}_i^p}\sum_{h\in\mathcal{M}_i^u}\ln\sigma(\hat{p}_{ik}-\hat{p}_{ih}), \quad (18)$$

where $\sigma(x)=\frac{1}{1+e^{-x}}$ is the logistic sigmoid function which is introduced to penalize the violated constraints in Eq.(17). As we can see in Eq.(18), the error function does not focus on predicting the right value, but on the ordering of the preference for observed, potential and unobserved locations.

4.2.2 The Parameter Estimation

In ARMF model, based on the Eq.(7), Eq.(8) and Eq.(18), the matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Q}$ are learned by minimizing the following regularized optimization problem:

$$\underset{\mathbf{U},\mathbf{V},\mathbf{Q}}{\operatorname{argmin}} -\sum_i\left(\sum_{j\in\mathcal{M}_i^o}\sum_{k\in\mathcal{M}_i^p}\ln\sigma(\hat{p}_{ij}-\hat{p}_{ik}) + \sum_{k\in\mathcal{M}_i^p}\sum_{h\in\mathcal{M}_i^u}\ln\sigma(\hat{p}_{ik}-\hat{p}_{ih})\right) + \Theta(\mathbf{U},\mathbf{V},\mathbf{Q}). \quad (19)$$

As there is no close-form for each variable with ALS approach, a Stochastic Gradient Descent (SGD) using the boostrap sampling with replacement algorithm is employed to solve the optimization problem in Eq.(19). The optimization algorithm is iteratively performed by sampling a tuple

(i, j, k, h) and updating the corresponding variables, where i is a user, $j \in \mathcal{M}_i^o$ is her observed location, $k \in \mathcal{M}_i^p$ is her potential location, and $h \in \mathcal{M}_i^u$ is other unobserved location. More details of optimization are provided in Algorithm 2. Specifically, we define $g'(x) = \sigma(x) - 1$.

Complexity Analysis. The run time of sampling a tuple (i, j, k, h) is quite small in each update and can be neglected. Hence, the complexity of the optimization algorithm is $\mathcal{O}(mK)$, where m is the total iteration number. In the experiments, m is proportional to the number of observed check-ins.

Algorithm 2: ARMF Optimization

Input: λ_u, λ_v, λ_q, η, $maxIter$
Output: $\mathbf{U}$, $\mathbf{V}$, $\mathbf{Q}$

Randomly initialize $\mathbf{U}$ and $\mathbf{V}$, $\mathbf{Q}$, $t \leftarrow 1$
while $t \leqslant maxIter$ **do**
  Randomly sample a (i, j, k, h), where i is a user, and j, k, h are her one observed, potential, and unobserved location
  $\tilde{\mathbf{v}}_j^i \leftarrow (q_{ic_j} + \varepsilon)\mathbf{v}_j$
  $\tilde{\mathbf{u}}_i^j \leftarrow (q_{ic_j} + \varepsilon)\mathbf{u}_i$
  $\tilde{p}_{ijk} \leftarrow g'(\hat{p}_{ij} - \hat{p}_{ik})$
  $\tilde{p}_{ikh} \leftarrow g'(\hat{p}_{ik} - \hat{p}_{ih})$
  $\mathbf{u}_i \leftarrow \mathbf{u}_i - \eta(\tilde{p}_{ijk}(\tilde{\mathbf{v}}_j^i - \tilde{\mathbf{v}}_k^i) + \tilde{p}_{ikh}(\tilde{\mathbf{v}}_k^i - \tilde{\mathbf{v}}_h^i) + \lambda_u \mathbf{u}_i)$
  $\mathbf{v}_j \leftarrow \mathbf{v}_j - \eta(\tilde{p}_{ijk}\tilde{\mathbf{u}}_i^j + \lambda_v \mathbf{v}_j)$
  $\mathbf{v}_k \leftarrow \mathbf{v}_k - \eta((\tilde{p}_{ikh} - \tilde{p}_{ijk})\tilde{\mathbf{u}}_i^k + \lambda_v \mathbf{v}_k)$
  $\mathbf{v}_h \leftarrow \mathbf{v}_h - \eta(-\tilde{p}_{ikh}\tilde{\mathbf{u}}_i^h + \lambda_v \mathbf{v}_h)$
  $\mathbf{q}_{ic_j} \leftarrow q_{ic_j} - \eta(\tilde{p}_{ijk}\mathbf{u}_i^T\mathbf{v}_j + \lambda_q q_{ic_j})$
  $\mathbf{q}_{ic_k} \leftarrow q_{ic_k} - \eta((\tilde{p}_{ikh} - \tilde{p}_{ijk})\mathbf{u}_i^T\mathbf{v}_k + \lambda_q q_{ic_k})$
  $\mathbf{q}_{ic_h} \leftarrow q_{ic_h} - \eta(-\tilde{p}_{ikh}\mathbf{u}_i^T\mathbf{v}_h + \lambda_q q_{ic_h})$
  $t \leftarrow t + 1$
end
return $\mathbf{U}$, $\mathbf{V}$, $\mathbf{Q}$

4.3 Incorporating Geographical Influence

Different from online product consuming, a POI's geographical distance significantly affects the user's check-in decision making process. One user would have a small probability to check-in a location far away, even though she is interested in it. In the example shown in Figure 1, user u_i has more chance to check-in the locations in the left side than those in the right side. It motivates us to incorporate the geographical influence into user's decision on POIs. Thus, the probability that user i prefers a POI j is:

$$\hat{p}_{ij} \propto p_{ij}^G \times \sigma(\hat{p}_{ij}) \Rightarrow \hat{p}_{ij} \propto p_{ij}^G \times \sigma((q_{ic_j} + \varepsilon)\mathbf{u}_i^T\mathbf{v}_j), \quad (20)$$

where p_{ij}^G is the geographical influence shown in Eq.(4).

4.4 Recommendation Strategies

Our goal is to recommend unvisited locations for users which they might be interested in. For each individual user, we first predict the probability that this user would check-in each unvisited location and then recommend the top-K locations with the highest probabilities for her. In particular, we adopt the following strategies for recommendation.

- **Standard Recommendation.** Similar to traditional recommendation, we consider to recommend the existing users with the existing locations. After learning the model from training data, we exploit Eq.(20) to predict the probability that one user prefers each unvisited location.
- **New User Recommendation.** When new users enter the system, we consider to recommend them with the existing locations. First, we need to re-train the models with these new users by leveraging the historical check-ins of their social friends and neighboring friends. As new users do not have check-ins, they do not have location friends but they have neighboring friends. After the latent factors are learned, Eq.(20) is employed for recommendations.
- **New Location Recommendation.** When new locations enter the system, we consider to recommend existed users for them. By utilizing the neighboring location characteristics, the probability that user i checks-in a new location j is defined as follows:

$$\hat{p}_{ij} \propto p_{ij}^G \times \sigma\left(\frac{\sum_{l \in \hat{\psi}_j} Sim_G(j, l)\hat{p}_{il}}{\sum_{l \in \hat{\psi}_j} Sim_G(j, l)}\right),$$

where $\hat{\psi}_j$ is the set of S nearest neighboring locations of location j in the training data and in the experiments S is set as 10. The advantage to exploit the similarity of neighboring locations is that we can handle new locations as soon as they are generated in the system, without needing to re-train the model and estimate new parameters.

5. EXPERIMENTS

In this section, we evaluate the proposed models with baseline methods on two real-world data sets.

5.1 Experimental Setup

Datasets. In this paper, we use Gowalla and Foursquare datasets to evaluate the performance of the proposed models. Gowalla contains check-in data ranging from January 2009 to August 2010, and Foursquare includes the check-in data of users who live in California, ranging from December 2009 to June 2013. Each check-in record in the datasets includes a user ID, a location ID and a timestamp, where each location has latitude, longitude and category information. Totally, there are 262 and 10 categories in Gowalla and Foursquare, respectively. Also, data sets have undirected friendship information and user's home information[3].

To evaluate model's cold-start recommendation performance, for each data set, we divide it in three steps. First, we remove those users who have visited less than 10 locations and those locations which are visited by less than 10 users. These check-ins are used to evaluate our model's performance for standard POI recommendation. In recommendation system, we aim to recommend those unvisited locations for users. Therefore, we split the training and testing data as follows: for each individual user, (1) aggregating the check-ins for each location; (2) sorting the location according to the first time that the user checked-in; (3) selecting the earliest 80% to train the model and using the next 20% as testing. Second, in the rest of check-ins (i.e., locations that are visited by less than 10 users and not included in the training), we use those check-ins whose locations are visited by users in the training data to evaluate the model's performance for new location recommendation. Third, in the rest of check-ins (i.e., users who have visited less than 10 locations), we use those check-ins where users are not in training data to evaluate user cold-start recommendation performance. The data statistics are shown in Table 1.

Experimental Settings. In the experiments, the parameters β, λ_u, λ_v, ζ, η and ε are set to 0.15, 0.015, 0.015, 0.5, 0.001, and 0.1, respectively. In Gowalla dataset, α, and λ_q are set to 0.3 and 500. In Foursquare dataset, α and λ_q are set to 0.1 and 300. We will discuss the influence of α in the Section 5.4.4. The latent feature number is set as 10.

[3]Our model can be applied in the general check-in datasets. The home location can be estimated by using the existing approach in [2, 3]

Table 1: The statistics of data sets.

							New Location Rec		New User Rec	
Data Set	#User	#Location	#Checkin	#Train	#Test	Sparsity	#New Location	#Test	#New User	#Test
Gowalla	52,216	98,351	2,577,336	2,049,630	527,706	0.0399%	78,881	568,937	9,326	79,153
Foursquare	2,551	13,474	124,933	100,033	24,900	0.2910%	93,311	119,876	1,221	17,964

Table 2: The performance comparison of standard recommendation in terms of MAP.

Data Set	ASMF-RW	ASMF-LA	ARMF-RW	ARMF-LA	USG	IRenMF	WRMF	BPR	LOCABAL	RegPMF	PMF
Gowalla	0.05700	0.05713	0.05715	0.05705	0.05205	0.02554	0.02470	0.03652	0.01446	0.01388	0.01357
Foursquare	0.04167	0.04064	0.03857	0.03907	0.03464	0.03683	0.03626	0.01923	0.02344	0.02325	0.02288

5.2 Evaluation Metrics

As POI recommender system only recommends the limited locations for users, we quantitatively evaluate our models versus other models in terms of ranking performance, i.e., Precision@K and Recall@K metrics. MAP metric, the mean of the average precision (AP) over all locations in the testing, is also adopted in the experiments to evaluate models' performance. They are formally defined as follows:

$$Precision@K = \frac{1}{N}\sum_{i=1}^{N}\frac{S_i(K)\cap T_i}{K}, Recall@K = \frac{1}{N}\sum_{i=1}^{N}\frac{S_i(K)\cap T_i}{|T_i|},$$

$$MAP = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j=1}^{\hat{m}_i} p(j)\times rel(j)}{|T_i|},$$

where $S_i(K)$ is a set of top-K unvisited locations recommended to user i excluding those locations in the training, and T_i is a set of locations that are visited by user i in the testing. $\hat{m}_i$ is the number of the returned locations in the list for user i, $p(j)$ is the precision of a cut-off rank list from 1 to j, and $rel(j)$ is an indicator function that equals to 1 if the location is visited in the testing, otherwise equals to 0.

5.3 Baseline Methods

To comparatively demonstrate the effectiveness of our models, we compare them with the following seven models:

- **USG** [28], which combines geographical influence, social network and user interest with collaborative filtering;
- **IRenMF** [15], which models geographical influence by incorporating neighboring characteristics into weighted matrix factorization in both instance level and region level;
- **LOCABAL** [22], which models two types of social relations: social friends and the users with high global reputations, in the framework of matrix factorization;
- **RegPMF** [16], which models the influence of social network by placing a social regularization constraint on learning user-specific feature vectors between friends;
- **PMF** [20], which minimizes the square error loss only using the observed check-ins based on matrix factorization.
- **WRMF** [6], which minimizes the square error loss by assigning both observed and unobserved check-ins with different confidential values based on matrix factorization;
- **BRP** [18], which optimizes the ordering of the preference for the observed location and the unobserved location.

In this paper, we develop two methods to learn the potential locations for each user (i.e., *LA*, *RW*) and then design two loss functions: *ASMF* and *ARMF*. Thus we consider the following combinations: *ASMF + LA*, *ASMF + RW*, *ARMF + LA*, *ARMF + RW*, which are denoted as **ASMF-LA**, **ASMF-RW**, **ARMF-LA**, **ARMF-RW**, respectively.

5.4 Performance Comparison

In this section, we evaluate the proposed models for standard recommendation, new location and new user recommendation in terms of Precision@K, Recall@K and MAP. In addition, we discuss the influence of α in *ASMF-LA* model.

5.4.1 Performance of Standard Recommendation

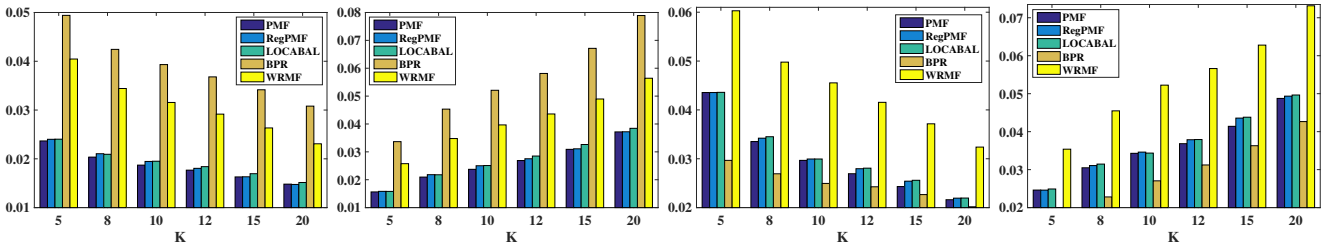
The performance comparison of our models and baseline models in terms of Precision@K, Recall@K, and Map are shown in Figure 3, Figure 4 and Table 2.

Modeling observed check-ins v.s. modeling all check-ins. From the results, we can see that *WRMF* and *BPR* almost outperform *LOCABAL*, *RegPMF* and *PMF*. Even though *LOCABAL* and *RegPMF* incorporate social network into matrix factorization, the sparseness of data due to only modeling the observed check-ins results in their poor performance. Both *LOCABAL* and *RegPMF* are slightly superior to *PMF*. One possible explanation is that social network assists to make more accurate recommendation. Different from them, *WRMF* not only utilizes the observed check-ins, but also models negative preference for all unvisited locations with a low confidence. But *BPR* easily leads to bias by only sampling some unvisited locations, which explains why it performs not good in Foursquare data set.

Our models v.s. baseline models. Our models achieve the best performance in both data sets with all evaluation metrics, illustrating the superiority of our approaches. Although *USG* exploits social influence, geographical effect and user interest, its simple linear combination results in the poor performance. As Gowalla covers a much larger area than Foursquare, the clustering result is not good, leading to the worse performance of *IRenMF* in Gowalla than in Foursquare. *ARMF* and *ASMF* have different performance in two datasets which is consistent with performance of *WRMF* and *BPR*. Their similar performance in Gowalla is due to the much more evident spatial clustering phenomenon. Two approaches to learn potential locations perform similarly. But *LA* is more efficient than *RW* because it does not require any matrix operation. In addition, the better performance of our models in Gowalla than in Foursquare is for the sake of (1) the stronger correlation in Gowalla which is reflected in Figure 2(c) and Figure 2(d), and (2) the more detailed category in Gowalla than in Foursquare, where Gowalla has 262 kinds of categories while Foursquare only has 10 different categories.

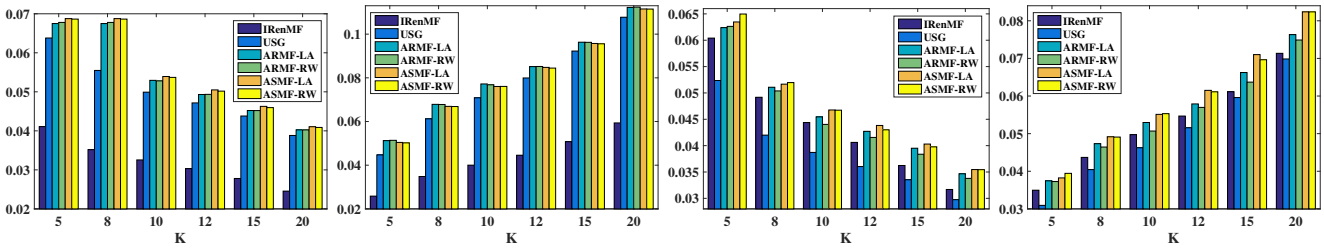
5.4.2 Performance of New POI Recommendation

In this section, we evaluate the model performance of addressing location cold-start problem. To recommend new locations, we predict the check-in probability for each new location and then recommend the top-K locations with the highest probabilities. Note that, among all the baseline methods, only *USG* and *IRenMF* can be applied here. Since new locations are never checked-in by any users, *USG* is reduced to only model the geographical influence. In addition, the latent location vectors for new locations are not learned in the training, so one user's preference for a new location in *IRenMF* model is actually dependent on her preferences for this location's neighborhoods. The model performance in terms of precision, recall and MAP is shown in Table 3.



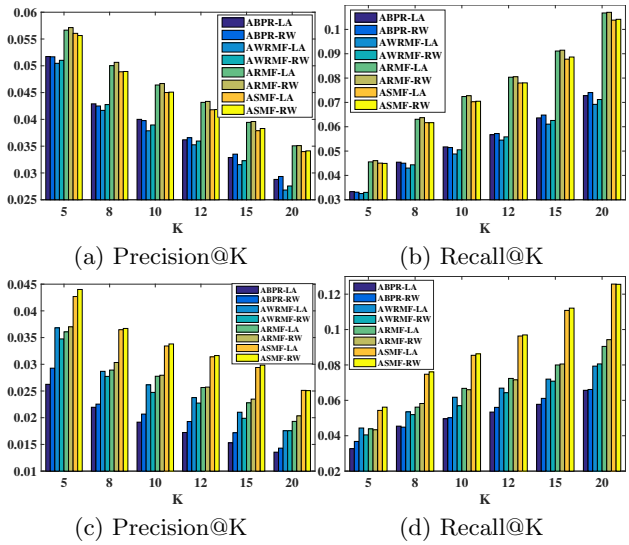
(a) Precision@K on Gowalla (b) Recall@K on Gowalla (c) Precision@K on Foursquare (d) Recall@K on Foursquare

Figure 3: The performance comparison of standard recommendation of basic methods for precision and recall.



(a) Precision@K on Gowalla (b) Recall@K on Gowalla (c) Precision@K on Foursquare (d) Recall@K on Foursquare

Figure 4: The performance comparison of standard recommendation of our models and other models.



(a) Precision@K (b) Recall@K

(c) Precision@K (d) Recall@K

Figure 5: The performance comparison of new user recommendation on Gowalla data set (top) and Fousquare data set(bottom).

Based on the results, we can observe that *IRenMF* performs the worst among all methods on both datasets. Although taking advantage of the similarities of neighboring locations, *IRenMF* fails to appropriately model user's check-in behaviours. It happens likely due to that it does not well exploit the inherent characteristics of geographical distance. On the other hand, our models and *USG* utilize the power-law distribution to capture the spatial clustering phenomenon for user's check-in activities, which is based on the observation over data. Therefore, they have much better performance than *IRrenMF* model in location cold-start recommendation. Also, our models gain superior performance over *USG*. A possible reason is that a user's preference latent vector has been learned in the training so that her preference on the target location's neighborhoods can be accurately predicted. However, *USG* only leverages the geographical similarity between a new location and her historical POIs as prediction. In addition, the performance of *ASMF* and *ARMF* is consistent with earlier experimental results.

5.4.3 Performance of New User Recommendation

In this section, we evaluate model's recommendation performance for user cold-start problem. When a new user enters the system, we do not have her historical check-in information. As a result, her latent vector cannot be learned and all of these baseline methods could not address this problem. The proposed models elaborate the historical check-ins of a user's neighboring friends (and social friends if she has) to learn her preference vector. Thus, they can be adopted to cope with user cold-start problem. As the proposed augmenting framework could be adapted to *WRMF* and *BPR* based models, we construct the following baseline methods with the similar loss functions in Eq.(18) and Eq.(11): (1) *WRMF+LA*, denoted as *AWRMF-LA*; (2) *WRMF+RW*, denoted as *AWRMF-RW*; (3) *BPR+LA*, denoted as *ABPR-LA*; (4) *BPR+RW*, denoted as *ABPR-RW*. The precision, recall and MAP of these models over two datasets are shown in Figure 5 and Table 4.

From the results, we find that all models have good performance to address user cold-start problem. The augmenting approach with friends' historical check-ins significantly benefits the location recommendation, in particular user cold-start recommendation. Meanwhile, the successful application of the augmenting strategy in *WRMF* and *BPR* demonstrates that the proposed augmenting strategy can be easily applied in any square error and ranking error based matrix factorization models. Moreover, our models perform much better than the baseline approaches for the sake of exploiting geographical influence and category information. *ARMF* and *ASMF* perform consistently as above results. Overall, we can see that our models can handle user cold-start problem very well.

5.4.4 Study of Influence of Parameter α

The *ASMF* model treats user's check-in as an indication of positive, potential and negative preference with difference confidence. The parameter α shown in Eq.(10) indicates the probability that users will check-in an unvisited location

Table 3: The performance comparison of new location recommendation.

	P@5	P@8	P@10	P@12	P@15	R@5	R@8	R@10	P@12	R@15	MAP
Gowalla Data Set											
ASMF-RW	0.08956	0.08543	0.08320	0.08095	0.07807	0.06032	0.08029	0.09259	0.10344	0.11883	0.08424
ASMF-LA	0.08967	0.08549	0.08323	0.08100	0.07807	0.06020	0.08035	0.09268	0.10353	0.11887	0.08430
ARMF-RW	0.09463	0.08946	0.08666	0.08401	0.08038	0.06457	0.08576	0.09822	0.10914	0.12437	0.08768
ARMF-LA	0.09456	0.08927	0.08643	0.08375	0.08022	0.06449	0.08564	0.09794	0.10901	0.12440	0.08766
USG	0.08632	0.07826	0.07407	0.07044	0.06587	0.05448	0.07645	0.08885	0.10007	0.11515	0.07578
IRenMF	0.00073	0.00094	0.00104	0.00113	0.00120	0.00033	0.00057	0.00079	0.00105	0.00140	0.00271
Foursquare Data Set											
ASMF-RW	0.04195	0.04276	0.04171	0.04144	0.04121	0.00419	0.00697	0.00843	0.01006	0.01247	0.02036
ASMF-LA	0.04171	0.04257	0.04230	0.04111	0.04116	0.00411	0.00680	0.00860	0.00992	0.01257	0.02040
ARMF-RW	0.04061	0.04085	0.04057	0.04052	0.04048	0.00382	0.00664	0.00823	0.00993	0.01296	0.02010
ARMF-LA	0.04022	0.04000	0.04002	0.03951	0.03949	0.00457	0.00701	0.00856	0.01017	0.01258	0.02051
USG	0.03551	0.03594	0.03452	0.03375	0.03301	0.00268	0.00561	0.00714	0.00886	0.01126	0.01592
IRenMF	0.00401	0.00339	0.00314	0.00304	0.00317	0.00038	0.00050	0.00055	0.00068	0.00108	0.00346

Table 4: The performance comparison of new user recommendation in terms of MAP.

Data Set	ASMF-RW	ASMF-LA	ARMF-RW	ARMF-LA	AWRMF-RW	AWRMF-LA	ABPR-RW	ABPR-LA
Gowalla	0.05442	0.05427	0.05589	0.05562	0.03021	0.02921	0.02423	0.02396
Foursquare	0.04831	0.04836	0.03770	0.03718	0.03242	0.03683	0.02971	0.02813

which her friends have checked-in before. In this section, we study the influence of variable α. Due to limited space, we only show the performance of *ASMF* model with *Linear Aggregation*. The precision, recall and MAP of *ASMF-LA* with different α value over two datasets are reported in Figure 6.

Based on the results, we can observe that the performance in all evaluation metrics has similar behaviour with the varying value of α. It is observed that *ASMF-LA* achieves the best performance when α is 0.3 and 0.1 on Gowalla and Foursquare, respectively. The performance then drops dramatically when α goes far away from the maximum point. If the α is set as a very small value, it will have no major difference to optimize those potential check-ins and other unobserved check-ins, which makes *ASMF-LA* difficult to obtain more accuracy prediction. If the α is set with a very large value, it will easily generate noise when optimizing both her own and friends' historical check-ins. It occurs possibly due to some other locations that a user's friends have checked-in but she might be in fact not interested in. As a result, α with a large value would probably affect the entire optimization process. Furthermore, the maximum point of α on Gowalla is larger than the one on Foursqaure, which indicates that users would have a larger chance to check-in their friends' POIs on Gowalla. This result is consistent with the observation that the correlation between users in Gowalla is much stronger than that in Foursquare. At last, we find that the performance with different α value on Gowalla changes much smaller than that on Foursquare. The more evident spatial clustering phenomenon on Gowalla than that on Foursquare is a reasonable explanation why *ASMF-LA* has such small change. On Gowalla dataset, geographical distance plays an extremely important role on affecting user's POI decision so that it compromises the prediction result even though user interest has a big change.

6. RELATED WORK

Related works about POI recommendation can be grouped into two categories. The first category focuses on modeling geographical influence [28, 3, 1, 30, 13, 14, 12, 15, 17]. Specifically, there are several approaches to model geographical distance. For example, some approaches leveraged Gaussian mixture model to characterize user's check-in activities [3, 1]; While some approaches utilized the kernel density estimation (KDE) to study user's check-in behavior and avoid employing a specific distribution[30, 13]. [28] proposed to use a power law distribution to estimate the check-in probability with the distance of any pair of visited POIs due to the spatial clustering phenomenon exhibited in LBSNs. The user's preference for one location is predicted by a linear model with combining users' interest, social friends' interests and geographical influence. Later, [15] considered two types of geographical neighborhood characteristics: instance level and region level. Specifically, in instance level, a user's preference for one location is modeled by a combination of her preference for this location and the nearest neighborhoods of this location. In region level, it places a group lasso penalty to learn location-specific latent vectors and capture the region effect.

The second category throws light on elaborating social network information [16, 5, 22, 28, 25, 4, 27, 7, 21, 9]. For example, [27, 28] proposed user-based collaborative filtering to estimate the unobserved rating by directly using the check-in information of friends. [16] assumed friends would share similar interests and then placed a social regularization term to constrain the objective functions for learning accurate user feature vectors. [5] proposed to model four types of social correlations (i.e., local friends, distant friends, local non-friends and distant non-friends) by using a geo-social correlation model with users' check-in activities, where the check-in probability was measured as a linear combination of these four geo-social correlations and the corresponding coefficients were learned by a group of features in a logistic regression like fashion. [22] modeled local and global social relations for all users. Specifically, in local context, it models the correlation between users and their friends; while in global context, it uses the reputation of a user in the whole social network as weight to fit observed ratings.

In addition, there are some recommendation works based on content, sentiments and temporal effect [10, 14, 26, 17, 29, 31]. However, our work is different from these existing works. To be specific, we first learn user's potential locations with the check-in information of social friends, location friends and neighboring friends, and then incorporate them into matrix factorization model using different error loss functions.

7. CONCLUSION

In this paper, we proposed a two-step framework for POI recommendation problem, which considers the check-in information of three types of friends, i.e., social friends, loca-

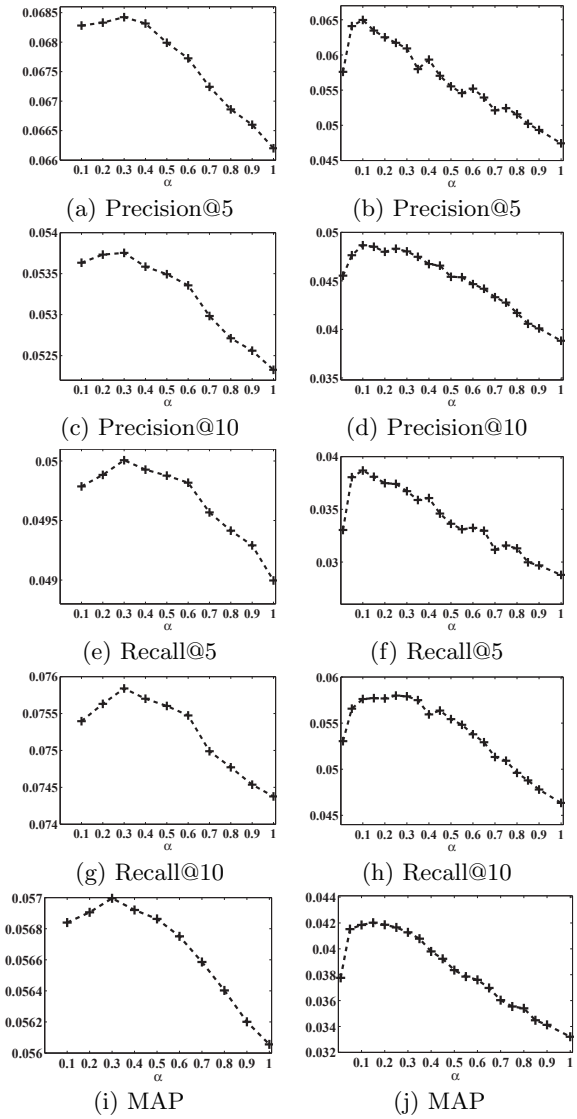


Figure 6: The influence of α on Gowalla data set (left) and Foursquare data set (right).

tion friends and neighboring friends. Specifically, in the first step, we designed two approaches to learn the locations that a user's friends had checked-in before and she was most interested in. In the second step, we developed matrix factorization based models with two different error loss functions using the learned potential locations. Specifically, the square error based loss extended a binary preference to a ternary variable for the observed check-ins, potential check-ins and other unobserved check-ins, and the ranking error based loss modeled the ranking of user's preference for her visited locations, potential locations, and unvisited locations. Finally, experimental results on two real-world data sets clearly validated the improvement of our models over many baseline methods based on different validation metrics.

Acknowledgments

This work is partially supported by NIH (1R21AA023975-01) and NSFC (71571093, 71372188, 61572032).

8. REFERENCES

[1] 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.

[2] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui. Exploring millions of footprints in location sharing services. In *ICWSM*, 2011.

[3] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: User movement in location-based social networks. In *KDD*, pages 1082–1090, 2011.

[4] H. Gao, J. Tang, and H. Liu. Exploring social-historical ties on location-based social networks. In *ICWSM*, 2012.

[5] H. Gao, J. Tang, and H. Liu. gscorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *CIKM*, pages 1582–1586, 2012.

[6] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[7] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR*, 2009.

[8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems, 2009.

[9] H. Li, R. Hong, S. Zhu, and Y. Ge. Point-of-interest recommender systems: A separate-space perspective. In *ICDM*, pages 231–240, 2015.

[10] H. Li, Z. W. Richang Hong, and Y. Ge. A spatial-temporal probabilistic matrix factorization model for point-of-interest recommendation. In *SDM*, pages 117–125, 2016.

[11] H. Li, H. Zhu, Y. Ge, Y. Fu, and Y. Ge. Personalized TV recommendation with mixture probabilistic matrix factorization. In *SDM*, pages 352–360, 2015.

[12] 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 *KDD*, 2014.

[13] M. Lichman and P. Smyth. Modeling human location data with mixture of kernel densities. In *KDD*, 2014.

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

[15] Y. Liu, W. Wei, A. Sun, and C. Miao. Exploiting geographical neighborhood characteristics for location recommendation. In *CIKM*, pages 739–748, 2014.

[16] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, 2011.

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

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

[19] S. Ruslan and M. Andriy. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*, 2008.

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

[21] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *KDD*, pages 1046–1054, 2011.

[22] J. Tang, X. Hu, H. Gao, and H. Liu. Exploiting local and global social context for recommendation. In *IJCAI*, 2013.

[23] W. Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography*, pages 234–240, 1970.

[24] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *ICDM*, pages 613–622, 2006.

[25] H. Wang, M. Terrovitis, and N. Mamoulis. Location recommendation in location-based social networks using user check-in data. In *SIGSPATIAL*, pages 374–383, 2013.

[26] D. Yang, D. Zhang, Z. Yu, and Z. Wang. A sentiment-enhanced personalized location recommendation system. In *HT*, 2013.

[27] M. Ye, P. Yin, and W.-C. Lee. Location recommendation for location-based social networks. In *GIS*, pages 458–461, 2010.

[28] 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, 2011.

[29] Z. Yu, H. Xu, Z. Yang, and B. Guo. Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. *IEEE Trans. Human-Machine Systems*, 46(1):151–158, 2016.

[30] J. Zhang and C. Chow. igslr: Personalized geo-social location recommendation - a kernel density estimation approach. In *SIGSPATIAL*, pages 334–343, 2013.

[31] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian. Mining mobile user preferences for personalized context-aware recommendation. *ACM (TIST)*, 5(4):58:1–58:27, 2014.