

Article

# Learning to Recommend Point-of-Interest with the Weighted Bayesian Personalized Ranking Method in LBSNs

Lei Guo <sup>1,\*</sup>, Haoran Jiang <sup>2</sup>, Xinhua Wang <sup>3</sup> and Fangai Liu <sup>3</sup>

<sup>1</sup> School of Management Science and Engineering, Shandong Normal University, Jinan 250014, China

<sup>2</sup> Information Technology Bureau of Shandong Province, China Post Group, Jinan 250001, China; haoranjiang@live.com

<sup>3</sup> School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China; wangxinhua@sdsu.edu.cn (X.W.); lfa@sdsu.edu.cn (F.L.)

\* Correspondence: leiguo.cs@gmail.com; Tel.: +86-531-8618-0509

Academic Editor: Kurt Maly

Received: 12 December 2016; Accepted: 26 January 2017; Published: 6 February 2017

**Abstract:** Point-of-interest (POI) recommendation has been well studied in recent years. However, most of the existing methods focus on the recommendation scenarios where users can provide explicit feedback. In most cases, however, the feedback is not explicit, but implicit. For example, we can only get a user's check-in behaviors from the history of what POIs she/he has visited, but never know how much she/he likes and why she/he does not like them. Recently, some researchers have noticed this problem and began to learn the user preferences from the partial order of POIs. However, these works give equal weight to each POI pair and cannot distinguish the contributions from different POI pairs. Intuitively, for the two POIs in a POI pair, the larger the frequency difference of being visited and the farther the geographical distance between them, the higher the contribution of this POI pair to the ranking function. Based on the above observations, we propose a weighted ranking method for POI recommendation. Specifically, we first introduce a Bayesian personalized ranking criterion designed for implicit feedback to POI recommendation. To fully utilize the partial order of POIs, we then treat the cost function in a weighted way, that is give each POI pair a different weight according to their frequency of being visited and the geographical distance between them. Data analysis and experimental results on two real-world datasets demonstrate the existence of user preference on different POI pairs and the effectiveness of our weighted ranking method.

**Keywords:** point-of-interest; location recommendation; LBSNs

## 1. Introduction

With the popularity of smart mobile devices and the development of the global positioning system (GPS), it has become easier for people to acquire real-time information regarding their locations, which has triggered the advent of location-based social networks (LBSNs), such as Foursquare, Gowalla, Facebook place, etc. These online systems enable people to check in and share life experiences with friends when they visit a point-of-interest (POI; e.g., restaurants, tourist spots and stores), which has not only led to location-based socializing becoming a new form of social interaction, but has also created the opportunities for people to explore interesting unknown places. To achieve the latter goal, POI recommendation has become one of the important means.

The task of POI recommendation is to model the users' visiting preferences and recommend to a user the POIs that she/he may be interested in, but has never visited. Compared with traditional recommender systems, POI recommendation is challenging for two reasons. First, the check-in data in LBSNs are not explicit, but implicit. In explicit feedback, such as movie rating data, users can explicitly

denote their “like” or “dislike” of an item with different rating scores. With check-in data, however, we can only get a user’s positive behaviors from the history of which POI she/he has checked-in and never know her/his interest in the POIs without check-ins, which are either unattractive or undiscovered, but potentially attractive. Therefore, the learning task for this kind of data is how to infer the users’ preference and non-preference from only positive check-in data. However, most existing POI recommendation methods overlook the implicit feedback facts and cannot reach reasonable results. Fortunately, some researchers have recently realized this problem and began to consider check-ins as implicit feedback. For example, Lian et al. [1] exploited a weighted matrix factorization for this task. Their work fit the nonzero check-ins by using large weights and zero check-ins by using small weights, but directly fitting zero check-ins may not be very reasonable, because zero check-ins may be missing values. Li et al. [2] considered the POI recommendation based on the ordered weighted pairwise classification (OWPC) criterion and proposed a new ranking-based factorization method. However, their work emphasized the classifications at top-N positions by assigning higher weights. Ying et al. [3] integrated users’ geographical preference and latent preference into the Bayesian personalized ranking (BPR) framework and proposed a hybrid pair-wise POI recommendation approach. However, their work gave equal weight to each POI pair. Intuitively, as a user usually has different preference for different POIs, different POI pairs should not be treated equally in the ranking cost function.

Second, geographical coordinates of POIs are available. Geographical coordinates are an important type of context information, since users have a much higher probability of visiting nearby POIs [4]. Previous works have developed different approaches to exploit this particular type of context to assist POI recommendation. For example, Ye et al. [5] discovered the spatial clustering phenomenon exhibited in user check-in activities and employed a power-law probabilistic model to capture the geographical influence among POIs. Different from the work in [5], Cheng et al. [6] modeled the probability of a user’s check-in on a location as a multi-center Gaussian. Rather than making a universal distribution for all users, Zhang et al. [7] used a kernel density estimation approach to personalize the geographical influence on users’ check-in behaviors as individual distributions. To avoid the cost of computing the distance between paired locations, Liu et al. [8] modeled the spatial clustering phenomenon in terms of geo-clustering and tried to estimate the individual spatial distribution. Recently, Lian et al. [1] incorporated this context into the factorization model by augmenting users’ and POIs’ latent factors with activity area vectors of users and influence area vectors of POIs. Li et al. [2] introduced one extra latent factor matrix to model the interaction between users and POIs to incorporate the geographical influence. Actually, by modeling the spatial clustering phenomenon, it becomes possible to particularly distinguish the preference difference from POI pairs with large geographical distance. In particular, it is much more likely that unvisited POIs with a large distance to a frequently-visited location are very unattractive, and this likelihood depends on the distance to that location. Therefore, the weighted ranking criterion will benefit from the introduction of such a phenomenon.

Based on the above observations, we consider the POI recommendation task as a pair-wise ranking problem and propose a weighted Bayesian personalized ranking method for the POI recommendation task. Specifically, in order to learn from check-in implicit feedback, we reconstruct the user-POI visit frequency matrix using the data policy proposed by Rendle et al. [9] and learn the user and POI latent factors by making use of the partial order of POIs. To investigate the usefulness of visit frequency, we assume that the higher the check-in frequency is, the more the POI is preferred by a user, and accordingly, we incorporate the check-in frequency into the ranking criterion by giving different item pairs with different weights. To explore the impact of geographical distance, we then assume the larger the distance of the unvisited POI to their previous locations is, the smaller the likelihood it will be visited by a user and the larger the contribution of this POI pair is. By merging these two contexts together, we reached our final weighted POI recommendation model. Data analysis and experimental results on two real-world datasets demonstrate the existence of a POI difference in visit frequency and

geographical distance, and the proposed weighted ranking method incorporating these two contexts can achieve higher recommendation results.

The remainder of this paper is organized as follows. First, we briefly review the related recommendation methods. Second, we derive the weighted ranking criterion from the Bayesian analysis of the problem by making two assumptions. Then, we conduct data analysis and experiments to demonstrate the existence of a POI difference in visit frequency and geographical distance and the effectiveness of our proposed method. Finally, we conclude this paper and give our future work.

## 2. Related Work

Due to a wide range of potential applications (e.g., personalized location service in GPS trajectory logs [10,11] and the LBSN data [12,13]), POI recommendation has attracted great research interest in recent years [14–16]. In LBSN data, a pioneer work of POI recommendation was discussed in a work by Ye et al. [17], which studied the location recommendation by exploiting the social and geographical characteristics of users and locations and proposed two friend-based collaborative filtering approaches. To achieve more accurate recommendation, Ye et al. [5] further extended and studied this work, wherein they discovered the spatial clustering phenomenon of user check-in activities and employed a power-law probabilistic model to capture the geographical influence among POIs. Yuan et al. [18] integrated spatial information into POI recommendation by making a different assumption; that is, humans tend to visit POIs near their previous locations, and their willingness to visit a POI decreases as the distance increases. Instead of making a power law distribution assumption, Cheng et al. [6] modeled the probability of a user's check-in at a location as a multi-center Gaussian model. Moreover, Zhang et al. [19] argued that the geographical influence on users should be unique and personal and should not be modeled as a common distribution. In their work, kernel density estimation was used to model the geographical influence as a personalized distance distribution for each user.

Some other related works focus on how to utilize other different contexts (e.g., social influence, temporal influence) to enhance the POI recommendation approach [18,20,21]. For example, social influence-enhanced POI recommendation approaches assume that friends in LBSNs share much more common interests than non-friends and utilize social relationships to enhance POI recommendation [6,17]. Temporal influence-enhanced POI recommendation approaches assume that users' interests vary with time and that users' visiting behaviors are often influenced by time, since users visit different places at different times in a day [18,22]. However, the above existing approaches are usually developed to fit the check-in frequency under a rating-based learning criterion, and ignoring the check-in data is a type of implicit feedback. Different from explicit rating data, check-in data can only provide us positive samples that a user likes, and the unvisited POIs are either unattractive or undiscovered, but potentially attractive locations.

Recently, some researchers have realized this problem and began to consider check-in data as a type of implicit feedback. For example, Lian et al. [1] proposed a weighted matrix factorization approach for POI recommendation. They incorporated the geographical clustering phenomenon by augmenting users' and POIs' latent feature vectors with an activity area vector of users and influence area vectors of POIs. This method fit the nonzero check-ins by using large weights and zero check-ins by using small 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 otherwise negative values. Li et al. [2] considered that the check-in frequency characterizes users' visiting preference and learned the factorization by ranking the POIs correctly. Ying et al. [3] proposed a hybrid pair-wise POI recommendation approach that integrates POI coordinates into the Bayesian personalized framework. However, these existing works give each POI pair equal weight and did not consider the POI difference in visit frequency and geographical distance. Intuitively, different POI pairs should impact the ranking criterion differently.

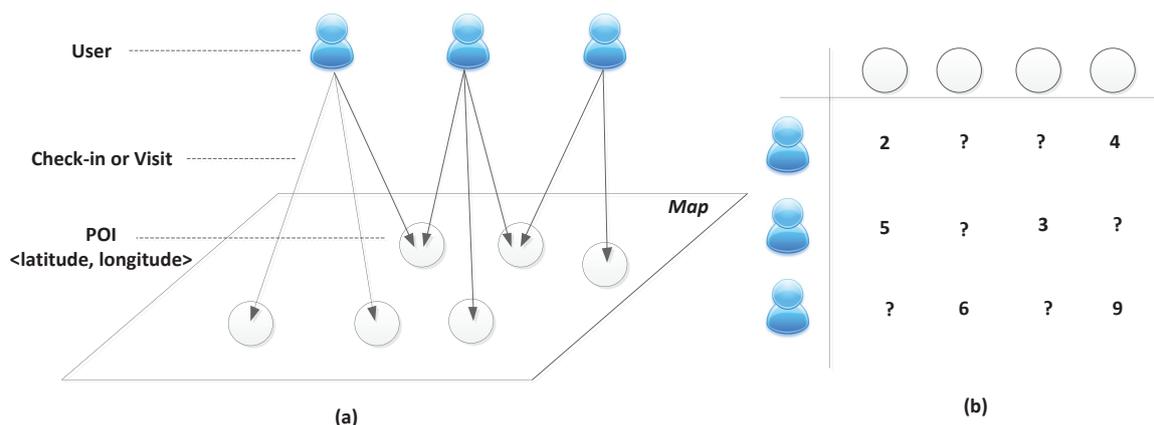
In this paper, we consider POI recommendation based on the Bayesian personalized ranking criterion [9]. Our proposed method differs from the existing approach in two aspects. First, the existing BPR was developed for the ranking problem with binary values, while in this work, we extend the objective function to rank POIs with different visiting frequencies. Second, we develop a weighted means to improve the learning criterion, which is able to exploit the visit frequency and geographical distance context information.

### 3. POI Recommendation with Weighted Ranking Criterion

In this section, we will systematically interpret how to model the visit frequency and geographical information as weighted terms to constrain the ranking-based POI recommendation method. We first describe the POI recommendation problem with only positive check-in observations and then derive the ranking criterion from the Bayesian analysis of the problem. Finally, we explore the impact of visit frequency and geographical distance by giving each POI pair different weights.

#### 3.1. Problem Description

The POI recommendation problem we studied in this paper is different from traditional recommender systems for two reasons: first, the former can only provide implicit feedback, which means we can only learn user preferences from their positive check-in behaviors; that is, we can only know which POI a user has visited, but do not know which POI the user is not willing to visit. Second, a POI usually has geographical information, from which we can get the exact location of a POI and the distance among POI pairs. Figure 1a shows an overview of POI recommendation in LBSNs. This process includes two central elements: the user-POI check-in frequency matrix (as shown in Figure 1b) and the geographical information of POIs (as shown in Figure 1a), i.e., latitude and longitude. Take a real-world recommendation scenario for example; suppose a user lives in Beijing, and she/he often checks in near the area of Zhongguancun (a famous technology hub of China). In this case, when we recommend the unvisited POIs to this user, both her/his check-in behaviors and the POI location need to be considered, and the closer to the area of Zhongguancun, the more priority the POIs have to be recommended.



**Figure 1.** Example of point-of-interest (POI) recommendation in location-based social networks (LBSNs). (a) Overview of location-based social network; (b) user-POI check-in frequency matrix.

In this toy example, each user visited some POIs with different check-in frequencies to express their favor toward POIs, but only positive behaviors can be observed. The remaining unknown data (denoted as “?”) are a mixture of actually negative and missing values. We cannot use a common approach to learn user features directly from unobserved data, as they are no longer able to be distinguished from the two levels. Moreover, different from traditional items (e.g., movies, books or products), each POI also has geographic coordinate information, from which one can get the exact POI

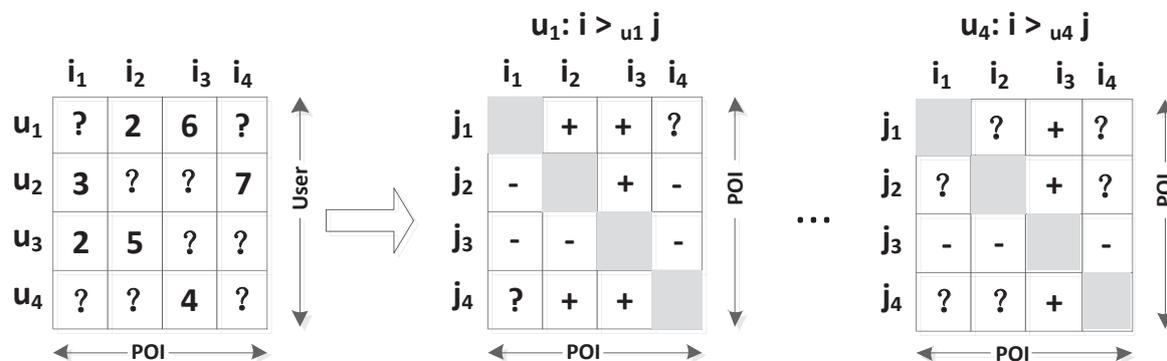
locations and distances. In the real world, the willingness of a user to move from one POI to another is a function of their distance. The problem we studied in this paper is how to effectively and efficiently predict the personalized POI rankings by employing these two central elements.

### 3.2. Bayesian Personalized Ranking Criterion

Let  $\mathcal{U}$  be the set of all users and  $\mathcal{I}$  be the set of all items. In our POI recommendation scenario, the check-in frequency matrix  $R \subseteq \mathcal{U} \times \mathcal{I}$  is available (see the left side of Figure 2), where each entry  $r_{ui} \in R$  indicates the visit frequency of user  $u$  to POI  $i$ , and  $r_{ui} = ?$  indicates that  $i$  is unvisited by  $u$  in the past (the status of  $u$  to  $i$  is unknown). To learn from this implicit feedback data, we reconstruct the user-POI visit frequency matrix using the following data policy; that is, if a POI  $i$  has been visited by user  $u$  (i.e.,  $(u, i) \in R$ ), then we assume that the user prefers this POI over all other unvisited POIs (e.g., POI  $i_1$  and  $i_4$  for user  $u_1$  in Figure 2). For the POIs that have both been visited by a user  $u$ , if the visit frequency to POI  $i$  is higher than  $j$  (i.e.,  $r_{ui} > r_{uj}$ ), then we assume that the user prefers  $i$  over  $j$ , and the higher the check-in frequency is, the more the POI is preferred by the user. For example, in Figure 2, POIs  $i_2$  and  $i_3$  have both been visited by user  $u_1$ , and the visit frequency  $r_{u_1 i_3} > r_{u_1 i_2}$ ; so we assume that this user prefers POI  $i_3$  over  $i_2$ :  $i_3 >_{u_1} i_2$ . For POIs that have both been visited by a user, but have the same visit frequency, we cannot infer any preference. The same is true for two POIs that a user has not visited yet (e.g., POIs  $i_1$  and  $i_4$  for user  $u_1$  in Figure 2). To formalize this, we create training data  $\mathcal{D}_{\mathcal{R}}: \mathcal{U} \times \mathcal{I} \times \mathcal{I}$  by:

$$\mathcal{D}_{\mathcal{R}} := \{(u, i, j) | i \in \mathcal{I}_u^+ \wedge (j \in \mathcal{I} \setminus \mathcal{I}_u^+ \vee j \in \mathcal{L}_i)\}$$

where  $\mathcal{I}_u^+$  is the visited POI set,  $\mathcal{I} \setminus \mathcal{I}_u^+$  is the unvisited POI set and  $\mathcal{L}_i$  is the visited POI set, but the visit frequency is less than POI  $i$ . The semantics of  $(u, i, j) \in \mathcal{D}_{\mathcal{R}}$  is that user  $u$  is assumed to prefer  $i$  over  $j$ . As  $>_u$  is antisymmetric, the negative cases are regarded implicitly.



**Figure 2.** On the left side, the observed data  $R$  are shown. Our approach creates user-specific pairwise preferences  $i >_u j$  between a pair of POIs. On the right side, a plus sign (+) indicates that a user prefers POI  $i$  over POI  $j$ ; a minus sign (-) indicates that a user prefers  $j$  over  $i$ .

In order to find the correct personalized ranking for all POIs  $i \in \mathcal{I}$ , we introduce the BPR [9] method as our basic recommendation framework. BPR is derived by a Bayesian analysis of the problem and maximizes the following posterior probability:

$$p(\Theta | >_u) \propto p(>_u | \Theta)p(\Theta),$$

where  $p(>_u | \Theta)$  represents the user-specific likelihood function,  $p(\Theta)$  is the prior probability function and  $\Theta$  denotes the parameter vector of an arbitrary model class (e.g., k-nearest-neighbor and matrix factorization). With the assumption that all users act independently and the ordering of each pair

of POIs for a specific user is independent of the ordering of every other pair, the likelihood function  $p(>_u | \Theta)$  of all users can be written as:

$$\prod_{u \in \mathcal{U}} p(>_u | \Theta) = \prod_{(u,i,j) \in D_R} p(i >_u j | \Theta)$$

where  $p(i >_u j | \Theta)$  is the individual probability that a user prefers item  $i$  to item  $j$  and can be defined as:

$$p(i >_u j | \Theta) := \sigma(\hat{r}_{uij}(\Theta))$$

$\sigma$  is the logistic sigmoid function [9] and  $\hat{r}_{uij}(\Theta)$  is an arbitrary real-valued function of the model parameter vector  $\Theta$ . By decomposing the estimator  $\hat{r}_{uij}$  as:

$$\hat{r}_{uij} := \hat{r}_{ui} - \hat{r}_{uj}$$

any standard collaborative filtering model can be applied to predict  $\hat{r}_{uv}$  (the preference of user  $u$  to POI  $v$ ).

By viewing the problem of predicting  $\hat{r}_{uv}$  as the task of estimating a matrix  $R : \mathcal{U} \times \mathcal{I}$  and using matrix factorization (MF) to approximate the target matrix  $R$  ( $\hat{r}_{uv} = U_u^T V_v$ ), the objective function of the BPR method for POI recommendation (BPR-POI) can be achieved:

$$\begin{aligned} BPR - POI = & - \sum_{(u,i,j) \in D_R} \ln \sigma(\hat{r}_{ui} - \hat{r}_{uj}) + \\ & \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 \end{aligned} \tag{1}$$

where  $\lambda_U$  and  $\lambda_V$  are the regularization parameters and  $U \in R^{l \times m}$  and  $V \in R^{l \times n}$  are the latent user and POI matrices, with column vectors  $U_u$ ,  $V_i$  or  $V_j$  representing user-specific and POI-specific latent feature vectors, respectively. As in MF, the zero-mean spherical Gaussian priors [23] are placed on user and item feature vectors:

$$\begin{aligned} p(U | \sigma_U^2) &= \prod_{u=1}^m \mathcal{N}(U_u | 0, \sigma_U^2 I) \\ p(V | \sigma_V^2) &= \prod_{k=1}^n \mathcal{N}(V_k | 0, \sigma_V^2 I) \end{aligned} \tag{2}$$

where  $I$  is the identity matrix and  $\mathcal{N}(x | \mu, \sigma^2)$  is the probability density function of the Gaussian distribution with mean  $\mu$  and variance  $\sigma^2$ .

### 3.3. Frequency-Based Weighted Ranking Criterion

As we have shown in Equation (1), BPR-POI is a pair-wise POI ranking approach, which does not try to regress a single predictor  $\hat{r}_{uv}$  to a single number, but instead tries to classify the difference of two predictions  $\hat{r}_{ui} - \hat{r}_{uj}$ . Compared with the point-wise or rating-based approach [24], it is closer to the concept of “ranking”, as it does not focus on accurately predicting the rating of each POI. However, this method gives equal weight to each POI pair and does not distinguish between their different contributions in learning the objective function. Intuitively, the higher the visit frequency of a user to a POI, the more preference this user expresses for that POI. In other words, if the frequencies of two POIs in one POI pair are obviously different, we are more confident to derive this as a positive training sample. Based on this intuition, we propose our first weighted Bayesian personalized ranking model with visit frequency (WBPR-F):

$$WBPR - F = - \sum_{(u,i,j) \in D_R} \ln \sigma(w_{uij}(\hat{r}_{ui} - \hat{r}_{uj})) + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 \quad (3)$$

where  $w_{uij} \in [0.5, 1]$  denotes the weight of the difference of two predictions  $\hat{r}_{ui} - \hat{r}_{uj}$ , and its value is determined by the difference of two visit frequencies  $f_{uij} = f_{ui} - f_{uj}$ ,

$$w_{uij} = 0.5 \frac{f_{uij} - \min_f}{\max_f - \min_f} + 0.5 \quad (4)$$

Equation (4) is the normalization version of  $w_{uij}$ , in which the Min-Max normalization method [25] is used to bound the range of  $w_{uij}$  into  $[0.5, 1]$ .  $\min_f$  and  $\max_f$  are the minimum value and maximum of all of the frequency differences, respectively. In our experiments, we found that choosing 0.5 as the minimum value of  $w_{uij}$  can reach the best result. A very small value of  $w_{uij}$  indicates that only the later information of Equation (5) is used, and the former information of Equation (5) is lost.  $f_{uv}$  is the visit frequency of user  $u$  to POI  $v$ , and when a POI is unvisited by user  $u$ ,  $f_{uv}$  is defined as  $f_{uv} = 0$ . When the frequency difference  $f_{ui} - f_{uj}$  is equal to  $\min_f$ ,  $w_{uij}$  achieved its minimum value of 0.5, and when  $f_{ui} - f_{uj}$  is equal to  $\max_f$ ,  $w_{uij}$  achieved its maximum value of one.

The weight factor  $w_{uij}$  allows the ranking function to treat each POI pair differently. For example, suppose that user  $u$  has visited POI  $i$  and  $j$ ; if their visit frequency is very close, a small weight value of this POI pair is achieved (say  $w_{uij} = 0.55$ ), then this POI pair  $i >_u j$  (suppose the visit frequency of  $i$  is higher than  $j$ ) should contribute less in learning the ranking order of  $u$ , since we cannot confidently deduce the ranking preference from item  $i$  and  $j$  for user  $u$ . On the other hand, if the visit frequencies of two POIs have a great difference, a larger weight value of this POI pair is achieved (say  $w_{uij} = 0.95$ ), and this POI pair should contribute more to the learning function, since we are very confident in deriving the user preference from these two POIs.

As the optimization criterion denoted by Equation (3) is differentiable, gradient descent-based algorithms are an obvious choice for minimization. As we can see, however, due to the huge number of preference pairs ( $\mathcal{O}(|\mathcal{R}||\mathcal{I}|)$ ), standard gradient descent is expensive to update the latent features over all pairs. To solve this issue, we exploit the strategy proposed in the BPR method, which is a stochastic gradient descent algorithm based on bootstrap sampling of the training triples  $(u, i, j)$ . Then, the corresponding latent factors  $U_u, V_i$  and  $V_j$  can be updated by the following gradients:

$$\begin{aligned} \frac{\partial WBPR - F}{\partial U_u} &= - \frac{1}{1 + e^{\hat{r}_{uij}}} w_{uij} (V_i - V_j) + \lambda_U U_u \\ \frac{\partial WBPR - F}{\partial V_i} &= - \frac{1}{1 + e^{\hat{r}_{uij}}} w_{uij} U_u + \lambda_V V_i \\ \frac{\partial WBPR - F}{\partial V_j} &= \frac{1}{1 + e^{\hat{r}_{uij}}} w_{uij} U_u + \lambda_V V_j \end{aligned} \quad (5)$$

### 3.4. Geographically-Based Weighted Ranking Criterion

The WBPR-F model imposes a weighted factor to constrain the contribution of each POI pair by exploring the visit frequency of users, which provides a more effective way to learn the ranking function. However, this approach does not consider the geographical information of POIs. In fact, when we derive user preferences from POI pairs, if the POIs are neighborhood locations, according to the geographical clustering phenomenon of human mobility behaviors, these two POIs are both much more likely to be visited by this user. We cannot deduce user preference from two indistinguishable POIs, and this POI pair should contribute less to the ranking function. On the other hand, if the POIs

have a large distance from each other, we can deduce the user preference from these POI pairs with high confidence. The large distance makes these samples contribute more to the ranking function. Based on the above intuition, we propose our second weighted Bayesian personalized ranking model with geographical distance (WBPR-D),

$$WBPR - D = - \sum_{(u,i,j) \in D_R} \ln \sigma(d_{ij}(\hat{r}_{ui} - \hat{r}_{uj})) + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 \tag{6}$$

where  $d_{ij} \in [0.5, 1]$  denotes the weight factor of geographical distance between location  $i$  and  $j$ , and its normalization version can be written as:

$$d_{ij} = 0.5 \frac{dis_{ij} - min_d}{max_d - min_d} + 0.5$$

The distance  $dis_{ij}$  between POI  $i$  and  $j$  can be computed by the Haversine formula [26] with latitude and longitude.  $min_d$  and  $max_d$  are the minimum and maximum value of all of the location distances, respectively. In our experiments, we chose 0.5 as the minimum value of  $d_{ij}$ .

Similar to the frequency-based weight factor, geographically-based weight factor  $d_{ij}$  also allows the ranking function to treat each POI pair differently. For example, suppose user  $u$  has visited POI  $i$  and  $j$ ; if the distance between  $i$  and  $j$  is very short, a small value of  $d_{ij}$  is achieved (e.g.,  $d_{ij} = 0.55$ ), and this POI pair should contribute less to the ranking function. The closer these two POIs are, the less they should contribute to the ranking function. On the other hand, if the distance between  $i$  and  $j$  is very large, a large value of  $d_{ij}$  is achieved (say  $d_{ij} = 0.95$ ), and this POI pair should contribute much to the ranking function.

As in the first model, a local minimum of the objective function given by Equation (6) can be found by performing stochastic gradient descent in latent feature vectors  $U_u, V_i$  and  $V_j$ :

$$\begin{aligned} \frac{\partial WBPR - D}{\partial U_u} &= - \frac{1}{1 + e^{\hat{r}_{uij}}} d_{ij} (V_i - V_j) + \lambda_U U_u \\ \frac{\partial WBPR - D}{\partial V_i} &= - \frac{1}{1 + e^{\hat{r}_{uij}}} d_{ij} U_u + \lambda_V V_i \\ \frac{\partial WBPR - D}{\partial V_j} &= \frac{1}{1 + e^{\hat{r}_{uij}}} d_{ij} U_u + \lambda_V V_j \end{aligned} \tag{7}$$

### 3.5. Fused Weighted Ranking Criterion

In order to further improve the recommendation result, we further combine the visit frequency and geographical distance in a linear model and arrive at our final weighted Bayesian personalized ranking model with visit frequency and distance (WBPR-FD). The ranking criterion of this model can be written as:

$$WBPR - FD = - \sum_{(u,i,j) \in D_R} \ln \sigma(((1 - \alpha)w_{uij} + \alpha d_{ij})(\hat{r}_{ui} - \hat{r}_{uj})) + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 \tag{8}$$

In Equation (8), the weight factors  $w_{uij}$  and  $d_{ij}$  are smoothed by the parameter  $\alpha$ , which naturally fuses two central contexts into the POI recommender systems. The parameter  $\alpha$  controls how the weight of the ranking function depends on the geographical distance.

Similar to the first and second models, a local minimum of this objective function can also be found by performing stochastic gradient descent in latent feature vectors  $U_u$ ,  $V_i$  and  $V_j$ :

$$\frac{\partial \text{WBPR} - \text{FD}}{\partial U_u} = -\frac{1}{1 + e^{\hat{r}_{uij}}}((1 - \alpha)w_{uij} + \alpha d_{ij})(V_i - V_j) + \lambda_U U_u \quad (9)$$

$$\frac{\partial \text{WBPR} - \text{FD}}{\partial V_i} = -\frac{1}{1 + e^{\hat{r}_{uij}}}((1 - \alpha)w_{uij} + \alpha d_{ij})U_u + \lambda_V V_i \quad (10)$$

$$\frac{\partial \text{WBPR} - \text{FD}}{\partial V_j} = \frac{1}{1 + e^{\hat{r}_{uij}}}((1 - \alpha)w_{uij} + \alpha d_{ij})U_u + \lambda_V V_j \quad (11)$$

where  $\hat{r}_{uij} = ((1 - \alpha)w_{uij} + \alpha d_{ij})(\hat{r}_{ui} - \hat{r}_{uj})$ . The learning algorithm for estimating the latent low rank matrices  $U$  and  $V$  is described in Algorithm 1.

**Algorithm 1: Learning procedure of WBPR-FD.**

- 1 **Input:**
- 2     The check-in frequency matrix  $R$ , weight factors  $w$  and  $d$ , parameter  $\alpha$   
learning rate  $\eta$ , regularization parameter  $\lambda_U$  and  $\lambda_V$
- 3 **Output:**
- 4      $U, V$
- 5 initialize  $U$  and  $V$
- 6 **repeat**
- 7     draw  $(u, i, j)$  from  $\mathcal{U} \times \mathcal{I} \times \mathcal{I}$
- 8      $\hat{r}_{uij} \leftarrow \hat{r}_{ui} - \hat{r}_{uj}$
- 9     Update  $U_u$ , the  $u$ -th row of  $U$  according to Equation (9);
- 10    Update  $V_i$ , the  $i$ -th row of  $V$  according to Equation (10);
- 11    Update  $V_j$ , the  $j$ -th row of  $V$  according to Equation (11);
- 12    Compute the objective function WBPR-FD( $t$ ) in step  $t$  according to Equation (8);
- 13 **until** WBPR-FD( $t$ )-WBPR-FD( $t - 1$ )  $< \epsilon$  (tolerate error);
- 14 **return**  $U$  and  $V$ ;

### 3.6. Complexity Analysis

The main cost of training the objective function of WBPR-FD is in computing the loss function (denoted by Equation (8)) and its gradients against feature vectors  $U_u, V_v$  offline, as a real-time online prediction can be performed immediately by computing  $\hat{r}_{uv} = U_u^T V_v$ . The complexity of Equation (8) is  $O(|R||I|l)$ , where  $l$  is the dimension of latent feature vectors,  $|R|$  is the number of check-in frequency matrix and  $|I|$  is the number of POIs. Since in our experiments,  $l$  is a very small number (set as 10), the complexity of Equation (8) mainly depends on the huge number of training triples  $O(|R||I|)$ . To reduce the training complexity, we exploit the strategy proposed in the BPR method and use a stochastic gradient descent algorithm based on bootstrap sampling of training triples in each update step. With this approach, WBPR-FD chooses the triples randomly (uniformly distributed) and can converge very quickly. The computational complexities for gradients  $\frac{\partial \text{WBPR} - \text{FD}}{\partial U}$ ,  $\frac{\partial \text{WBPR} - \text{FD}}{\partial V}$  are both  $O(Nl)$ , where  $N$  is the number of sampled triples. Therefore, the total computational complexity is  $O(|R||I|l) + O(Nl)$ , which is linear with respect to the number of sampled triples.

## 4. Data Analysis and Experiments

In this section, we first investigate the relationship between two central contexts (i.e., visit frequency and geographical distance) and users' preferences on real-world LBSNs and then conduct

several experiments to compare the recommendation performance of our approach with other collaborative filtering methods and the state-of-the-art POI recommendation methods.

#### 4.1. Datasets

We make use of two real-world datasets—Gowalla [27] and Brightkite [28]—as the data source in our experiments [29]. Gowalla is a location-based social network launched in 2007, where users are allowed to check into locations that they have visited, and users with similar interests can make friends with each other. The Gowalla dataset was collected by using their public API over the period of February 2009–October 2010. The total number of the crawled check-ins for Gowalla is 6.4 million from 196,591 users. Brightkite is another location-based social networking website, where users are able to check-in at places by using text message or one of the mobile applications. Brightkite also allowed registered users to connect with their existing friends and meet new people based on the places that they have gone. Once a user has checked in at a place, they could post notes and photos to a location, and other users could comment on those posts. The Brightkite dataset was collected by using their public API over the period of April 2008–October 2010 and resulted in 4.5 million check-ins from 58,228 users. To simplify, we only consider check-in data for POI recommendation in these two datasets. To alleviate the data sparsity, for Gowalla, we only keep users that have checked in at least five different POIs and POIs that have been checked in at least 30 times, resulting in a user-POI matrix with 575,323 nonzero entries. For Brightkite data, we only keep users that have checked in at least three different POIs, and POIs that have been checked in at least 20 times, resulting in a user-POI matrix with 100,069 nonzero entries. More details of our dataset can be found in Table 1.

**Table 1.** Statistics of user-POI check-in matrix.

Statistics	Gowalla	Brightkite
number of users	32,134	11,142
number of POIs	8867	4369
number of check-ins	575,323	100,069
Min. number of POIs per user	5	3
Min. number of check-ins per POI	1	1
Check-in sparsity	99.838	99.833

#### 4.2. Empirical Data Analysis

To gain a better understanding of users' check-in behaviors, in this section, we investigate the existence of a POI difference with different visit frequency and geographical distance and try to answer the following two questions: (1) Do POIs with similar visit frequency tend to share similar user preferences? (2) Do POIs located in neighborhood places tend to be preferred by similar users? To answer the first question, we need to define how to measure the similarity between a pair of POIs from user check-in behaviors.

Let  $A(i)$  be the set of users that have visited POI  $i$  and  $A(j)$  be the set of users that have visited POI  $j$ . The similarity between POI  $i$  and  $j$  can be computed by the cosine similarity:

$$Sim_{ij} = \frac{|A(i) \cap A(j)|}{\sqrt{|A(i)||A(j)|}} \quad (12)$$

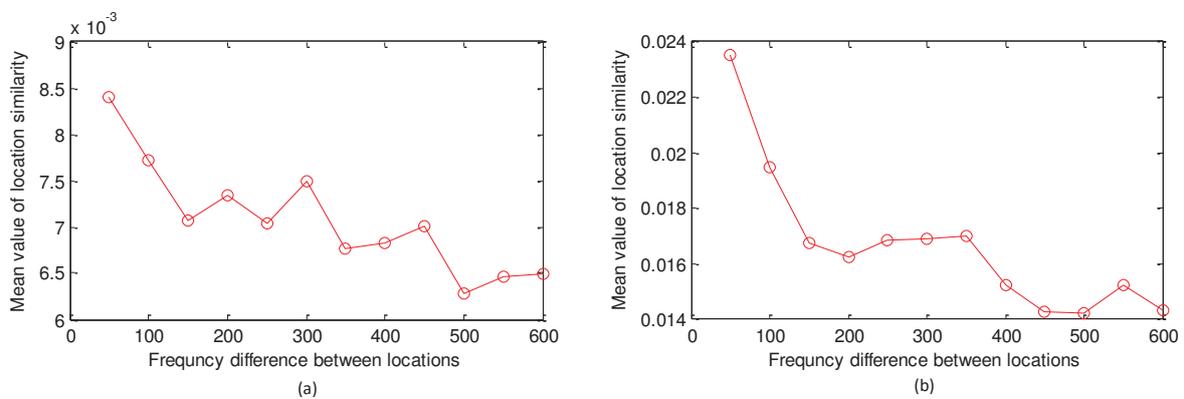
where  $|A(i)|$  denotes the length of set  $A(i)$ . However, this cosine similarity function does not consider the influence of popular users. Intuitively, active users usually visit a wide range of POIs (she/he may check in all of the POIs that she/he has ever gone), but many of them are not in her/his interests. However, when we know that two POIs are often visited by inactive common users, we can say

that these two POIs are similar with high confidence. Based on the above intuition, Breese et al. [30] proposed the adjusted cosine similarity function:

$$Sim_{ij} = \frac{\sum_{u \in A(i) \cap A(j)} \frac{1}{\log(1 + |A(u)|)}}{\sqrt{|A(i)| |A(j)|}} \tag{13}$$

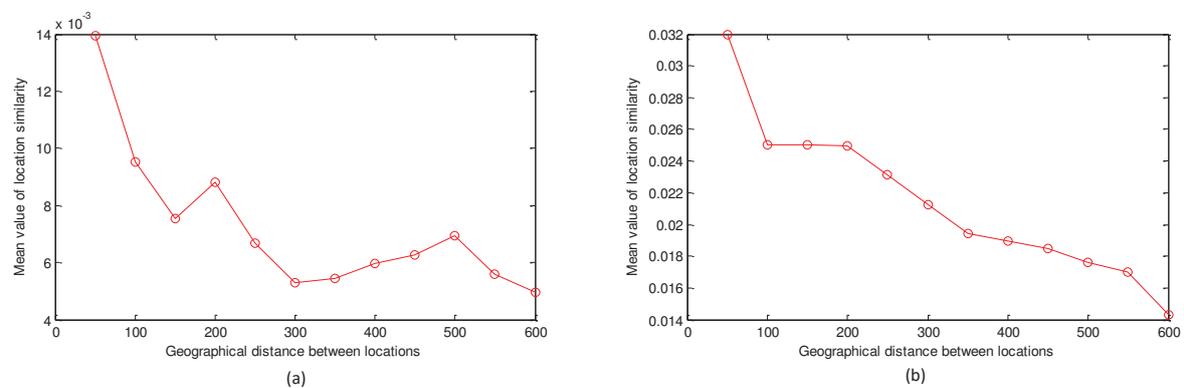
where  $A(u)$  is the set of POIs visited by user  $u$ . The term  $1/\log(1 + |A(u)|)$  punishes the influence of active users.

Figure 3 plots the relationship between the POI similarity and the frequency difference of randomly-chosen POI pairs. For both Gowalla and Brightkite datasets, we can observe that the mean value of POI similarities decreases with the increasing of frequency difference. This evidence suggests a positive answer to our first question: POIs with similar visit frequency tend to share more similar user preferences.



**Figure 3.** Relationship between frequency difference and POI similarity in the (a) Gowalla and (b) Brightkite datasets.

To answer the second question, we first randomly choose POI pairs from a POI dataset and then compute the similarities of locations by using Equation (13). The geographical distance between every POI pair is computed by the Haversine formula. Figure 4 plots the relationship between POI similarity and the geographical distance of randomly-chosen POI pairs. From Figure 4, we can observe that POI similarity decreases with increasing of geographical distance, which indicates that geographically-adjacent POIs tend to be visited by the same set of users. This evidence suggests a positive answer to our second question: neighborhood POIs are more likely to share similar user preferences.



**Figure 4.** Relationship between geographical distance and POI similarity in the (a) Gowalla and (b) Brightkite datasets.

### 4.3. Evaluation Metrics

We use three widely-used metrics [22] *Precision@k*, *Recall@k* and normalized discounted cumulative gain (*NDCG@k*) to measure the personalized ranking performance of our proposed approaches. *Precision@k* measures how many previously-labeled POIs are recommended to the users among the total number of recommended POIs, and *Recall@k* measures how many previously-labeled POIs are recommended to the users among the total number of labeled POIs. The formulations of *Precision@k* and *Recall@k* are defined as follows:

$$Precision@k = \frac{1}{M} \sum_{u=1}^M \frac{|L^k(u) \cap L^T(u)|}{k}$$

$$Recall@k = \frac{1}{M} \sum_{u=1}^M \frac{|L^k(u) \cap L^T(u)|}{|L^T(u)|}$$

Given an individual user  $u$ ,  $L^T(u)$  denotes the set of corresponding visited locations in the testing data, and  $L^k(u)$  denotes the top  $k$  recommended POIs by a method.  $k$  is the number of the recommendation list.  $M$  is the number of users. Specifically, we choose *Precision@5* and *Recall@5* as evaluation metrics in our experiments.

Normalized discounted cumulative gain (NDCG) measures the ranking quality of a recommendation algorithm based on the graded relevance of the recommended POIs. It varies from 0–1, with one representing the ideal ranking of the POIs. The NDCG at position  $k$  (*NDCG@k*) of a ranked POI list for a given user is defined as follows [31,32]:

$$NDCG@k = \frac{DCG@k}{IDCG@k}$$

where *DCG@k* is the discounted cumulative gain accumulated at a particular rank position  $k$  and *IDCG@k* is the ideal DCG through that position  $k$ :

$$DCG@k = \sum_{i=1}^k \frac{rel_i}{\log_2(i+1)} \quad \text{and} \quad IDCG@k = \sum_{i=1}^{|REL|} \frac{rel_i}{\log_2(i+1)}$$

where  $rel_i$  is the graded relevance (denoted by visit frequency in this work) of the result at position  $i$  and  $|REL|$  is the list of relevant POIs (ordered by their relevance) in the corpus up to position  $k$ .

### 4.4. Performance Comparison

In order to evaluate the recommendation performance of our proposed approaches, we compare the recommendation results with the following methods:

**Random:** This method provides the basic recommendation result in our experiments, which ranks the POIs randomly for the users in the test set.

**MostPopular:** This method weights the POIs by how often they have been visited in the past; that is, the order of the recommend POIs is determined by their popularity. This simple method is supposed to have reasonable performance, since many people tend to visit the popular locations.

**WRMF:** The weighted matrix factorization method for one-class rating (WRMF) was proposed by Pan et al. [33] and Hu et al. [34] for item prediction with only positive implicit feedback, which extends the matrix factorization method and adds weights in the error function to decrease the impact of negative samples.

**GeoMF:** This method was proposed by Lian et al. [1] and is the state-of-the-art method for POI recommendation.

**BPR-POI:** This is our proposed method that introduces the Bayesian personalized ranking criterion [9] for POI recommendation.

**WBPR-R:** This is the POI recommendation method that weights each POI pair with randomly-generated values (bounded in [0.5, 1]).

WBPR-F: This is our frequency-based weighted ranking method proposed for POI recommendation by giving different weights to each POI pair.

WBPR-D: This is our geographically-based method that weights each POI pair with different geographical distances.

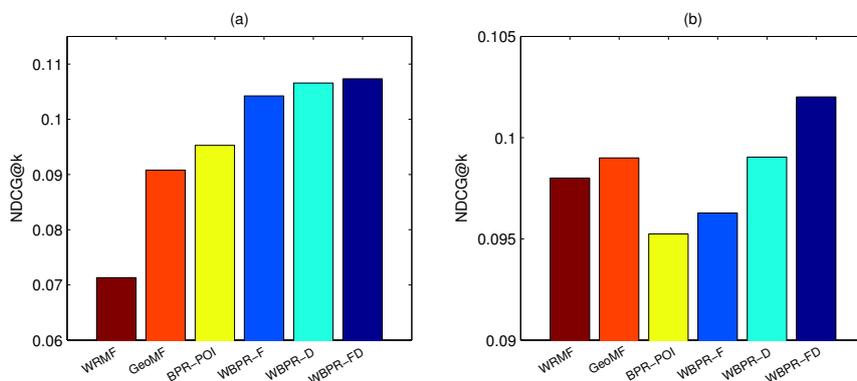
WBPR-FD: This is our fused weighted ranking method that can consider the frequency difference and geographical distance simultaneously.

In our experiments, we split the training data with different ratios to test the above algorithms. For example, training data 80% means we randomly select 80% actions (80% user-item pairs) of each user for training and predict the remaining 20% actions. The action selection is conducted five times independently. The parameter settings of our approaches are: for Gowalla, we set the parameter  $\alpha = 0.8$ , and for Brightkite, we set  $\alpha = 0.9$ . The regularization parameters of latent factors are set as  $\lambda_U = \lambda_{V_i} = 0.0025$ ,  $\lambda_{V_j} = 0.00025$ . The latent factor dimension  $l$  is set as 10.

Table 2 shows the comparison results for two real-world datasets in the measure of *Precision@k* and *Recall@k*. Although the MostPopular method only ranks the POIs based on their popularity, we can observe that this simple method has reasonable performance, since many people tend to focus on popular POIs. WRMF, as the state-of-the-art matrix factorization method proposed for item recommendation with implicit feedback, achieves a better precision than the MostPopular method, but does worse than the BPR-POI method. BPR-POI achieves a substantial improvement over WRMF, which indicates that optimizing the pairwise rank criteria directly for POI recommendation is more reasonable. In this work, we also include comparisons with the state-of-the-art POI recommendation method GeoMF. We find that our weighted ranking method WBPR-FD outperforms GeoMF in both Gowalla and Brightkite datasets. This result indicates that treating each POI pair in a weighted way can utilize the check-in data more effectively, and our method can model the users' ranking preference more accurately.

**Table 2.** Performance comparisons with different training datasets in the metrics of *Precision@k* and *Recall@k* (training = 80%,  $k = 5$ ). WRMF, weighted matrix factorization method for one-class rating; BPR, Bayesian personalized ranking; WBPR-F, weighted Bayesian personalized ranking model with visit frequency; WBPR-D, weighted Bayesian personalized ranking model with geographical distance.

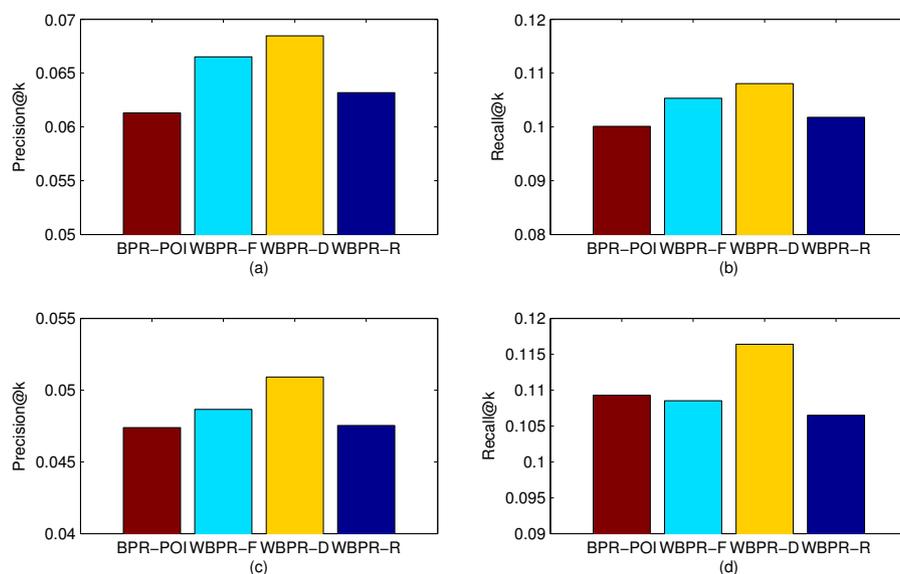
Metric	Dataset	Random	MostPopular	WRMF	GeoMF	BPR-POI	WBPR-F	WBPR-D	WBPR-FD
<i>Precision@k</i>	Gowalla	0.00046	0.01562	0.0500	0.0602	0.0613	0.0665	0.06846	0.06952
	Brightkite	0.00058	0.01811	0.04828	0.04987	0.04739	0.04866	0.05091	0.05227
<i>Recall@k</i>	Gowalla	0.00055	0.02236	0.0667	0.1004	0.1001	0.10531	0.10803	0.11140
	Brightkite	0.00104	0.04146	0.10484	0.10943	0.10929	0.10852	0.1164	0.11794



**Figure 5.** Performance comparisons with different training datasets in the metric of normalized discounted cumulative gain (*NDCG*)@ $k$  (Dimensionality = 10,  $k = 5$ ). (a) *NDCG*@ $k$  in Gowalla; (b) *NDCG*@ $k$  in Brightkite.

Figure 5 shows the comparison results in the measure of  $NDCG@k$ , from which similar results can be observed, that is our weighted ranking method can also do well in the metric that considers the positions of the result list.

To explore the effectiveness of different ways of deducing the weight factor, we conduct experiments on three methods by utilizing visit frequency, geographical distance and fused information separately, where the WBPR-FD method outperforms WBPR-F and WBPR-D and reaches the best performance. This result demonstrates that only using the frequency or distance context to weight POI pairs will not get satisfactory results, and it is beneficial to fuse them together. To further explore the impact of weight factors on the recommendation results, we conduct experiments on the method that generates the weight values randomly. As shown in Figure 6, although WBPR-R generates the weight values randomly, it slightly does better than the method BPR-POI that treats each POI pair equally, but does worse than WBPR-F and WBPR-D. This result also indicates that treating each POI pair in a weighted way is helpful, and the weight factors we learned are effective for POI recommendation. Note that with a small value of latent factor dimensions, the results of MF-based methods BPR-POI, WRMF, WBPR-F, WBPR-D and WBPR-FD can achieve a reasonable performance, which can significantly reduce the computational complexity. Hence, in our following experiments, we set the latent factor dimension to 10.



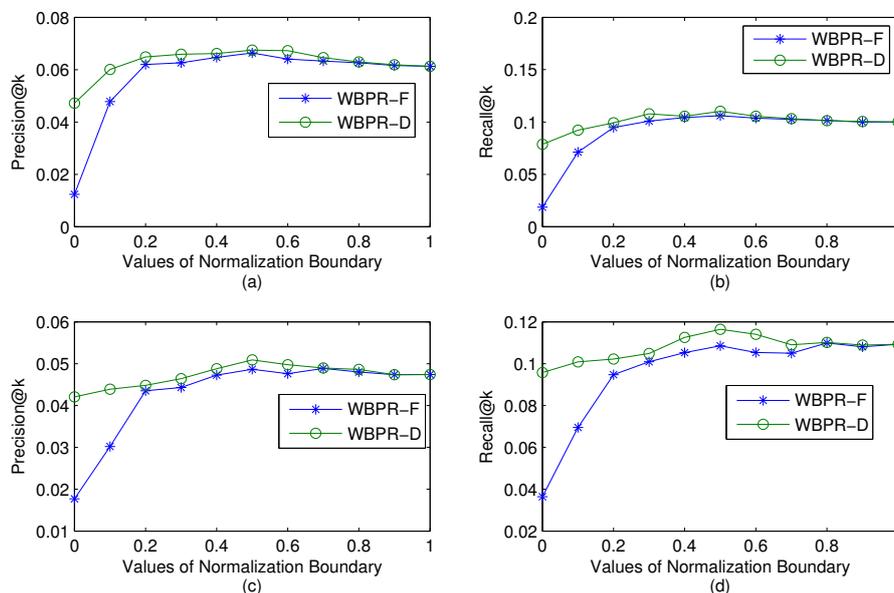
**Figure 6.** Impact of weight factors on the recommendation performance in Gowalla and Brightkite (Dimensionality = 10,  $k = 5$ ). (a)  $Precision@k$  in Gowalla; (b)  $Recall@k$  in Gowalla; (c)  $Precision@k$  in Brightkite; (d)  $Recall@k$  in Brightkite.

#### 4.5. Impact of Parameters

##### 4.5.1. Impact of the Normalization Boundary of the Weight Factors

In WBPR-F and WBPR-D, we use the normalized version of the weight factors ( $w_{uij}$  and  $d_{ij}$ ) to weight POI pairs and bound them into  $[0.5, 1]$ . To explore the impact of different normalization boundaries of the weight factors on the recommendation performance, we conduct experiments on a real-world dataset by varying the value of the normalization boundary from 0–1. Figure 7 shows the comparison results on Gowalla and Brightkite data. We observe that when the normalization boundary increases, the values of  $Precision@k$  and  $Recall@k$  increase (recommendation performance increase) at first, but when the boundary surpasses a certain threshold (0.5), the values of  $Precision@k$  and  $Recall@k$  decrease with the boundary further increasing. The reason that a small normalization boundary does not work is that in our data, many POI pairs are neighborhood places and have similar visit frequencies.

In these cases, when we update the latent factors (denoted in Equations (5) and (7)), only the latter information (i.e.,  $\lambda_U U_u$ ,  $\lambda_V V_i$  and  $\lambda_V V_j$ ) is used, and the former information of Equations (5) and (7) is lost. Hence, in the experiments, we choose 0.5 as the normalization boundary.

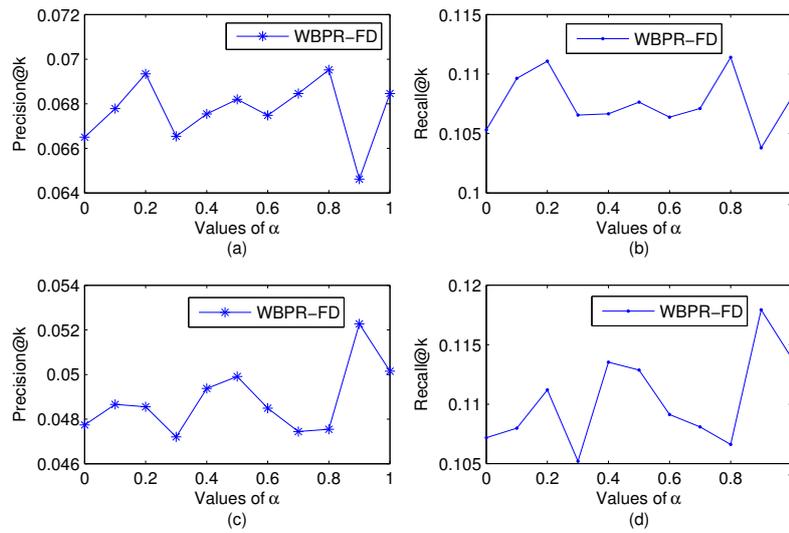


**Figure 7.** Impact of normalization boundary of the weight factors on the recommendation performance in Gowalla and Brightkite (dimensionality = 10,  $k = 5$ ). (a) *Precision@k* in Gowalla; (b) *Recall@k* in Gowalla; (c) *Precision@k* in Brightkite; (d) *Recall@k* in Brightkite.

#### 4.5.2. Impact of Parameter $\alpha$

In WBPR-FD, the parameter  $\alpha$  plays an important role and balances the information from visit frequency and geographical distance to weight POI pairs. It controls how much our weighted Bayesian personalized ranking method WBPR-FD should depend on the geographical distance. If  $\alpha = 0$ , we will only utilize the visit frequency to weight the difference of two POIs. If  $\alpha = 1$ , we will only utilize the geographical distance to weight POI differences. In other cases, we simultaneously weight the POI difference from the information of visit frequency and geographical distance.

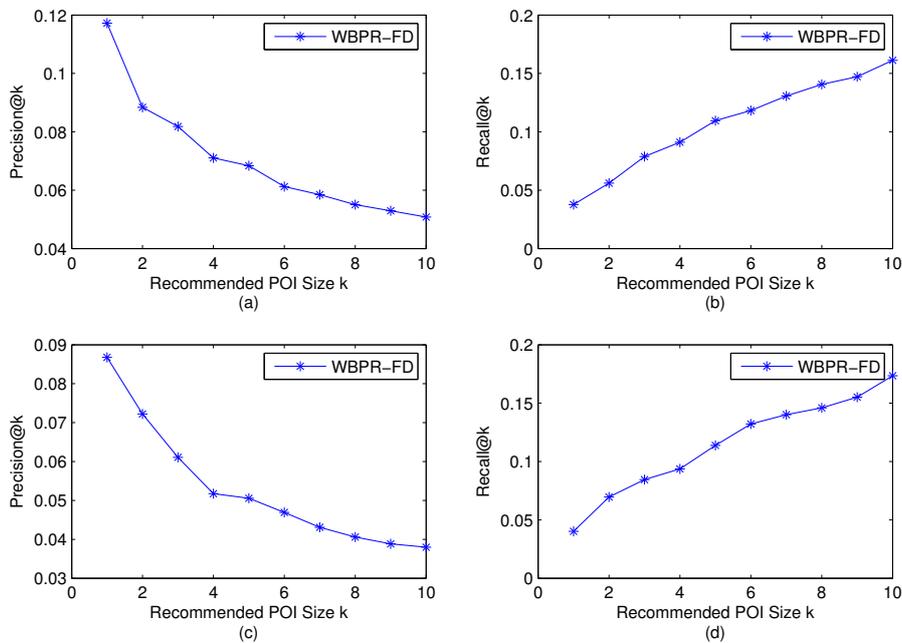
To get an appropriate  $\alpha$  value, we use a five-fold cross-validation to learn and test. We conduct each experiment five times and take the mean value as the final result. Figure 8 illustrates how the changes of  $\alpha$  affect the recommendation results on the measures *Precision@k* and *Recall@k*. We notice that although the metrics *Precision@k* and *Recall@k* jump all over the place by varying  $\alpha$  from 0–1, but in overview, purely utilizing visit frequency or geographical distance for recommendations cannot make better results than fusing them together. This result also indicates that although the geographical distance and visit frequency are both impact factors, it is difficult to distinguish which factor has a greater impact on the recommendation results. In experiments, we choose the settings that can reach the best results as the final  $\alpha$  value. For Gowalla, we set  $\alpha$  as 0.8. For Brightkite, we set  $\alpha$  as 0.9.



**Figure 8.** Impact of parameter  $\alpha$  on the recommendation performance in Gowalla and Brightkite (dimensionality = 10,  $k = 5$ ). (a) Precision@k in Gowalla; (b) Recall@k in Gowalla; (c) Precision@k in Brightkite; (d) Recall@k in Brightkite.

4.6. Impact of Recommended POI Sizes

In the top- $k$  POI recommendation systems, users can specify the number of most relevant POIs that the system shall return to her/him. Appropriate POI size helps to avoid overwhelming the user with a large number of POIs by returning only the number of the most relevant ones that she/he wishes. In our experiments, we varied the number of recommended POIs provided to the users, and set the POI size  $k$  from 1–10.

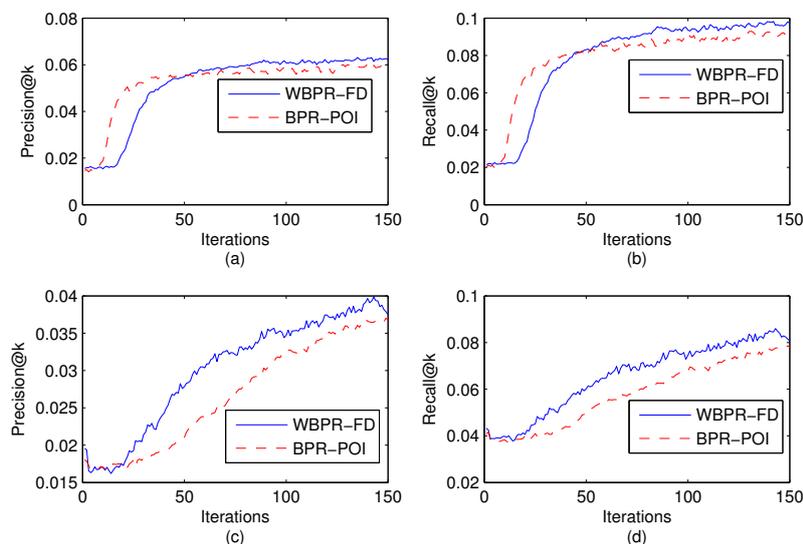


**Figure 9.** Impact of recommended POI size in the Gowalla and Brightkite data. (a) Precision@k in Gowalla. (b) Recall@k in Gowalla. (c) Precision@k in Brightkite. (d) Recall@k in Brightkite.

Figure 9 shows how the change of  $k$  affects the recommendation results of our WBPR-FD method on  $Precision@k$  and  $Recall@k$ , which only considers the top-most results returned by the system. We observe that when the number of recommended POIs increases, the hits of correct POIs increase, and this results in the increase of  $Recall@k$  values and the decrease of  $Precision@k$  values. The increase of the  $Recall@k$  denotes that the recommended POI list can cover more previous labeled POIs, which indicates that the ability of retrieving relevant POIs of our method is increasing. The decrease of  $Precision@k$  denotes that the recommended POI list can hit less correct POIs, which indicates that the ability of predicting correct POIs of our method is decreasing.

#### 4.7. Convergence Analysis

We further compare the convergence of the WBPR-FD and the BPR-POI method. Figure 10 shows the comparison results on the Gowalla and Brightkite data. For these two methods, in each iteration, we select the same number of instances for training and set the learning rate for both as 0.05. From the results, we see that both BPR-POI and WBPR-FD can converge within 50 iterations in Gowalla data and within 60 iterations in Brightkite data. Incorporating a weight factor does not slow down the convergence rate of WBPR-FD, but makes it achieve a higher performance than BPR-POI.



**Figure 10.** Convergence analysis on the Gowalla and Brightkite datasets. (a)  $Precision@k$  in Gowalla; (b)  $Recall@k$  in Gowalla; (c)  $Precision@k$  in Brightkite; (d)  $Recall@k$  in Brightkite.

## 5. Conclusions

With the rapid growth of LBSNs, how to effectively recommend POIs to users has become more and more important. In this work, we focused on the POI recommendation problem in the implicit feedback and proposed a novel weighted POI ranking method named WBPR-FD. We derived the optimization criterion of WBPR-FD from a Bayesian analysis of the problem, where we weighted each POI pair by utilizing visit frequency and geographical distance to improve the POI recommendation performance. Data analysis and experimental results on two real-world datasets demonstrated the existence of POI differences and the effectiveness of the weighted POI ranking method WBPR-FD.

In this work, we mainly focused on how to model the POI difference by giving different weights to each POI pair, but we did not consider the users' social interests. In fact, users who have similar social relations tend to visit similar POIs. As future work, we plan to develop new social interest-aware algorithms to further improve our weighted POI ranking method.

**Acknowledgments:** This work is supported by the Natural Science Foundation of China (61602282, 90612003, 61502284), the Postdoctoral Science Foundation of China (2016M602181) and the Higher Educational Science and Technology Program of Shandong Province (J15LN56).

**Author Contributions:** Lei Guo conceived and designed the algorithm and the experiments, and wrote the manuscript; Haoran Jiang performed the experiments and analyzed the experiments; Xinhua Wang and Fangai Liu provided the instructions during design the algorithm. All authors have read and approved the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lian, D.; Zhao, C.; Xie, X.; Sun, G.; Chen, E.; Rui, Y. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 831–840.
2. Li, X.; Cong, G.; Li, X.L.; Pham, T.A.N.; Krishnaswamy, S. Rank-GeoFM: A ranking based geographical factorization method for point of interest recommendation. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 433–442.
3. Ying, H.; Chen, L.; Xiong, Y.; Wu, J. PGRank: Personalized Geographical Ranking for Point-of-Interest Recommendation. In Proceedings of the 25th International Conference Companion on World Wide Web. International World Wide Web Conferences Steering Committee, Montreal, QC, Canada, 11–15 April 2016; pp. 137–138.
4. Tobler, W.R. A computer movie simulating urban growth in the Detroit region. *Econ. Geogr.* **1970**, *46*, 234–240.
5. Ye, M.; Yin, P.; Lee, W.C.; Lee, D.L. Exploiting geographical influence for collaborative point-of-interest recommendation. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 24–28 July 2011; pp. 325–334.
6. Cheng, C.; Yang, H.; King, I.; Lyu, M.R. Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; Volume 12, pp. 17–23.
7. Zhang, J.D.; Chow, C.Y. iGSLR: personalized geo-social location recommendation: A kernel density estimation approach. In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Orlando, FL, USA, 5–8 November 2013; pp. 334–343.
8. Liu, B.; Fu, Y.; Yao, Z.; Xiong, H. Learning geographical preferences for point-of-interest recommendation. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 1043–1051.
9. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009; pp. 452–461.
10. Zheng, V.W.; Zheng, Y.; Xie, X.; Yang, Q. Towards mobile intelligence: Learning from GPS history data for collaborative recommendation. *Artif. Intell.* **2012**, *184*, 17–37.
11. Cho, S.B. Exploiting machine learning techniques for location recognition and prediction with smartphone logs. *Neurocomputing* **2016**, *176*, 98–106.
12. Yin, H.; Cui, B.; Chen, L.; Hu, Z.; Zhang, C. Modeling location-based user rating profiles for personalized recommendation. *ACM Trans. Knowl. Discov. Data* **2015**, *9*, 19.
13. Gao, H.; Tang, J.; Liu, H. Personalized location recommendation on location-based social networks. In Proceedings of the 8th ACM Conference on Recommender Systems, Silicon Valley, CA, USA, 6–10 October 2014; pp. 399–400.
14. Yin, H.; Cui, B.; Huang, Z.; Wang, W.; Wu, X.; Zhou, X. Joint modeling of users' interests and mobility patterns for point-of-interest recommendation. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015; pp. 819–822.
15. Cheng, C.; Yang, H.; King, I.; Lyu, M.R. A Unified Point-of-Interest Recommendation Framework in Location-Based Social Networks. *ACM Trans. Intell. Syst. Technol.* **2016**, *8*, 10.

16. Zhao, S.; Zhao, T.; Yang, H.; Lyu, M.R.; King, I. STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
17. Ye, M.; Yin, P.; Lee, W.C. Location recommendation for location-based social networks. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 3–5 November 2010; pp. 458–461.
18. Yuan, Q.; Cong, G.; Ma, Z.; Sun, A.; Thalmann, N.M. Time-aware point-of-interest recommendation. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 28 July–1 August 2013; pp. 363–372.
19. Zhang, J.D.; Chow, C.Y.; Li, Y. iGeoRec: A personalized and efficient geographical location recommendation framework. *IEEE Trans. Serv. Comput.* **2015**, *8*, 701–714.
20. Gao, H.; Tang, J.; Hu, X.; Liu, H. Content-Aware Point of Interest Recommendation on Location-Based Social Networks. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 1721–1727.
21. Griesner, J.B.; Abdesslem, T.; Naacke, H. POI Recommendation: Towards Fused Matrix Factorization with Geographical and Temporal Influences. In Proceedings of the 9th ACM Conference on Recommender Systems, Vienna, Austria, 16–20 September 2015; pp. 301–304.
22. Gao, H.; Tang, J.; Hu, X.; Liu, H. Exploring temporal effects for location recommendation on location-based social networks. In Proceedings of the 7th ACM conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 93–100.
23. Ma, H.; Yang, H.; Lyu, M.R.; King, I. Sorec: Social recommendation using probabilistic matrix factorization. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; pp. 931–940.
24. Karatzoglou, A.; Baltrunas, L.; Shi, Y. Learning to rank for recommender systems. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 493–494.
25. Normalization (statistics). Available online: [https://en.wikipedia.org/wiki/Normalization\\_\(statistics\)](https://en.wikipedia.org/wiki/Normalization_(statistics)) (accessed on 4 February 2017).
26. Haversine formula. Available online: [https://en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula) (accessed on 4 February 2017).
27. Gowalla. Available online: <https://snap.stanford.edu/data/loc-gowalla.html> (accessed on 4 February 2017).
28. Brightkite. Available online: <https://snap.stanford.edu/data/loc-brightkite.html> (accessed on 4 February 2017).
29. Cho, E.; Myers, S.A.; Leskovec, J. Friendship and mobility: User movement in location-based social networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 1082–1090.
30. Breese, J.S.; Heckerman, D.; Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, USA, 24–26 July 1998; pp. 43–52.
31. Weimer, M.; Karatzoglou, A.; Le, Q.V.; Smola, A. Maximum margin matrix factorization for collaborative ranking. Available online: <http://www.markusweimer.com/files/pub/2007/2007-NIPS.pdf> (accessed on 4 February 2017).
32. Guo, L.; Ma, J.; Chen, Z.; Zhong, H. Learning to recommend with social contextual information from implicit feedback. *Soft Comput.* **2015**, *19*, 1351–1362.
33. Pan, R.; Zhou, Y.; Cao, B.; Liu, N.N.; Lukose, R.; Scholz, M.; Yang, Q. One-class collaborative filtering. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 502–511.
34. Hu, Y.; Koren, Y.; Volinsky, C. Collaborative filtering for implicit feedback datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 263–272.

