CoRe: Exploiting the Personalized Influence of Two-dimensional Geographic Coordinates for Location Recommendations

Jia-Dong Zhang\textsuperscript{a}, Chi-Yin Chow\textsuperscript{a,}\textsuperscript{*}

\textsuperscript{a}Department of Computer Science, City University of Hong Kong, Hong Kong

Abstract

With the rapid growth of location-based social networks (LBSNs), location recommendations play an important role in shaping the life of individuals. Fortunately, a variety of community-contributed data, such as geographical information, social friendships and residence information, enable us to mine users’ reality and infer their preferences on locations. In this paper, we propose an effective and efficient location recommendation framework called CoRe. CoRe achieves three key goals in this work. (1) We model a personalized check-in probability density over the two-dimensional geographic coordinates for each user. (2) We propose an efficient approximation approach to predict the probability of a user visiting a new location using her personalized check-in probability density. (3) We develop a new method to measure the similarity between users based on their social friendship and residence information, and then devise a fusion rule to integrate the geographical influence with the social influence so as to improve the user preference mod-
el on location recommendations. Finally, we conduct extensive experiments to evaluate the recommendation accuracy, recommendation efficiency and approximation error of CoRe using two large-scale real data sets collected from two popular LBSNs: Foursquare and Gowalla. Experimental results show that CoRe achieves significantly superior performance compared to other state-of-the-art geo-social recommendation techniques.

**Keywords:** Location-based social networks, Location recommendations, Personalization, Geographical influence, Social influence

### 1. Introduction

In a location-based social network (LBSN) depicted in Figure 1, such as Foursquare and Gowalla, users can establish social links with others, indicate their residence, check in some points-of-interest (POIs), e.g., restaurants, stores, and museums, and share their experiences of visiting specific locations by leaving some tips or comments [12]. These community-contributed data enable us to mine users’ reality [40, 42, 46]. For example, in LBSNs it is prevalent to recommend locations for users based on their preferences inferred from the community-contributed data [3, 44], which not only helps users explore new places and enrich their life but also enables companies to launch advertisements to potential customers and improve their profits.

The geographical feature of POIs distinguishes them from other non-spatial items, such as books, music and movies in conventional recommender systems [5], because physical interactions are required for users to visit POIs. Thus, the geographical information (geographic coordinates) of locations plays a significant influence on users’ check-in behaviors.
known as \textit{geographical influence} for short, which gives new opportunities and challenges to infer users’ preferences for making location recommendations for them.

\textbf{Limitations in existing methods.} There are a few studies that learn users’ preferences through exploiting the geographical influence for location recommendations \cite{9, 29, 36, 56}. Unfortunately, these studies use the one-dimensional geographical distance between locations for location recommendations. For example, the literatures \cite{29, 36} simply assume that the propensity of a user for a POI is inversely proportional to the distance between the POI and her visited locations; more sophisticatedly, two techniques \cite{9, 56} model the distance between locations visited by the same user as a power-law distribution or multi-center Gaussian distribution. In general, these studies suffer from two major limitations. (1) \textbf{One-dimensional geographical distance influence.} Existing work simplifies the geographical influence as a one-dimensional and monotonous distance distribution, but it is more reasonable and intuitive to model the geographical influence as a two-dimensional and multimodal check-in probability density. The reason is twofold. (a) The probability of a user visiting a location is not simply monotonous respect-
ing their distance, because the visiting probability is not only affected by
the distance but also the location’s intrinsic characteristics. For example,
in reality the check-in locations of a user are usually distributed in several
areas. (b) It is hard to compute a visiting probability for a location based
on a distance distribution, since it needs to find a reference location to de-
rive a reasonable distance for the location in the first place. Conversely, it
is considerably intuitive to employ a two-dimensional check-in probability
density to compute a visiting probability for any location with latitude and

Figure 2: Personal check-in probability densities
longitude. In addition, it is rational to use the two-dimensional density on the spherical dimensions because the altitude is a dependent variable (i.e., the altitude is usually determined based on a pair of latitude and longitude) and the earth is almost a sphere (i.e., the altitude of a location is negligible).

(2) Non-personalized geographical influence. They apply the same inversely proportional relation between the propensity and the distance for all users [29, 36] or universally estimate a common distance distribution for all users [9, 56], but in practice the geographical influence of locations should be personalized since the user’s check-in behavior is unique [62].

Real-world motivating examples. To observe users’ unique check-in behaviors and multimodal check-in probability densities over the two-dimensional geographic coordinates, a spatial analysis is conducted on two publicly available real data sets collected from Foursquare [15] and Gowalla [11], which are two popular LBSNs. Specifically, we focus on three users who are randomly chosen from each data set. Figure 2 depicts their individual check-in probability density over the two-dimensional geographic coordinates. The probability density is estimated based on the kernel density estimation (KDE) technique [49]. We have the following two findings. (1) The geographical influence of locations on these three users’ check-in behaviors is unique since their check-in probability densities are distinct from each other. (2) These check-in probability densities are usually multimodal rather than unimodal or monotonous.

Our approach. In this paper, we propose a new location recommendation framework, called CoRe, that achieves three key goals. (1) We infer users’ preferences by exploiting the personalized two-dimensional geographi-
cal influence. Specifically, we estimate a personalized two-dimensional check-in probability density over the latitude and longitude coordinates for each user rather than using a common one-dimensional distance distribution for all users. In addition, since the check-in probability densities of users are diverse, we cannot assume their forms. We thus use a nonparametric density estimation method, i.e., the popular kernel density estimation [49]. (2) We can use a personalized check-in probability density to predict the probabilities of each user visiting each of all new locations (that have not been visited by the user). In particular, we contrive an efficient approximation approach to derive these visiting probabilities based on the fast Gauss transform [20], in order to improve system efficiency and scalability. (3) We integrate the geographical influence with the social influence in the conventional social networks to enhance the user preference model and obtain better quality of location recommendations. First of all, to consider the social influence, we develop a new method to measure the similarity between users based on their social friendships and residence information, because nearby friends share more commonly visited locations than others [11]; the similarity measure between users is employed to estimate the rating of a user to a new location. We then design a new fusion rule, called the union rule, to integrate the geographical probability and social rating of a user to a new location into a unified score, instead of applying the simple product or sum rule. Eventually, we can make a location recommendation for each user by top-k locations with the highest scores to her.

This study is significant different from our previous work [62]. The main differences and contributions of this paper can be summarized as follows:
• Although both studies utilize the same kernel density estimation technique to model a geographical check-in probability density for each user, this study exploits the two-dimensional geographic coordinates of users’ check-in locations whereas the previous work [62] only considers the distance between users’ check-in locations. Actually, this is the first study to learn users’ preferences for location recommendations by utilizing the personalized two-dimensional geographical influence. Moreover, to improve the computational complexity of location recommendations, this study proposes an efficient approximation approach to predict the probability of a user visiting any new location rather than using the exact computation method as in the previous work [62]. (Section 2)

• Both studies employ a fusion rule to integrate the geographical influence with the social influence in order to improve the quality of location recommendations. However, this study develops a more sophisticated fusion rule whereas the previous work [62] applies the simple product or sum rule. In addition, this work devises a new method to measure the similarity between users based on their social friendships and residence information. (Section 3)

• We conduct extensive experiments to evaluate not only the recommendation accuracy of CoRe as in the previous work [62], but also its recommendation efficiency and approximation error using two large-scale real data sets collected from Foursquare and Gowalla. Experimental results show two key facts. (1) CoRe outperforms the state-of-the-art geo-
social recommendation techniques including the multi-center Gaussian model (MGM) [9], power-law distribution (PD) [56] and iGSLR [62] in terms of recommendation accuracy and efficiency. (2) CoRe is much more efficient than the exact method, and it only leads to small approximation errors. (Sections 4 and 5)

The remainder of this paper is organized as follows. We firstly model the personalized two-dimensional geographical influence for each user and propose an efficient approach to predict the probabilities of the user visiting new locations in Section 2. We then integrate geographical influence with social influence through a new user similarity measuring method and a new fusion rule in Section 3. In Sections 4 and 5, we present our experiment settings and analyze the performance of CoRe, respectively. Section 6 highlights related work on location recommendations. Finally, we conclude this paper in Section 7.

2. Modeling Personalized Two-dimensional Geographical Influence

In this section, we propose a kernel density estimation (KDE) approach to model the personalized two-dimensional geographical influence in Section 2.1 and develop an efficient approximation approach to derive users’ preferences in Section 2.2. Table 1 summarizes the key notations used in this paper.

2.1. A KDE-Based Approach to Model the Personalized Two-dimensional Geographical Influence

The experimental results in Section 1 inspired us to learn the users’ preferences by utilizing the personalized two-dimensional geographical influence.
<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>Set of all users in an LBSN</td>
</tr>
<tr>
<td>$u$</td>
<td>Some user and $u \in U$</td>
</tr>
<tr>
<td>$L$</td>
<td>Set of all locations (or POIs) in an LBSN</td>
</tr>
<tr>
<td>$l$</td>
<td>Some location with the latitude and longitude coordinates and $l = (lat, lon)^T \in L$</td>
</tr>
<tr>
<td>$L_u$</td>
<td>Set of locations visited by user $u$ (i.e., $u$'s check-in locations) and $L_u = {l_1, l_2, \ldots, l_n} \subset L$</td>
</tr>
<tr>
<td>$p(l</td>
<td>L_u)$</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>A set of cells that constitute a partition of the geographical space</td>
</tr>
<tr>
<td>$B$</td>
<td>Some cell and $B \in \Omega$</td>
</tr>
<tr>
<td>$l_B$</td>
<td>Center of $B$</td>
</tr>
<tr>
<td>$F_u$</td>
<td>Set of users having social links with $u$ and $F_u \subset U$</td>
</tr>
<tr>
<td>$r_{u',l}$</td>
<td>Actual rating of user $u'$ to \textbf{visited} location $l$</td>
</tr>
<tr>
<td>$\hat{r}_{u,l}$</td>
<td>Estimated rating of user $u$ to \textbf{unvisited} location $l$</td>
</tr>
<tr>
<td>$\hat{s}_{u,l}$</td>
<td>Unified score by fusing $p(l</td>
</tr>
</tbody>
</table>

To this end, we estimate a personalized two-dimensional check-in probability density for each user. Specifically, we apply a general nonparametric probability density estimation technique, known as the kernel density estimation [49] (KDE). Opposed to parametric estimation methods, a nonpara-
metric estimation technique does not assume a fixed distribution form in advance, but instead learn the distribution form from data. Thus, KDE can be used with arbitrary distributions because it does not have any assumption on the form of the underlying distribution.

**Personalized two-dimensional check-in probability density estimation.** Let \( L_u = \{l_1, l_2, \ldots, l_n\} \) be the set of locations visited by the user \( u \) that are drawn from some distribution with an unknown density \( f \). Its kernel density estimator \( \hat{f} \) using \( L_u \) is given by:

\[
\hat{f}(x) = \frac{1}{n\sigma^2} \sum_{i=1}^{n} K\left( \frac{x - l_i}{\sigma} \right),
\]

where each location \( l_i = (\text{lat}_i, \text{lon}_i)^T \) is a two-dimensional column vector with the latitude (\( \text{lat}_i \)) and longitude (\( \text{lon}_i \)), \( K(\cdot) \) is the kernel function and \( \sigma \) is a smoothing parameter, called the bandwidth. In this paper we apply the widely used standard two-dimensional normal kernel [49]:

\[
K(x) = \frac{1}{2\pi} \exp\left( -\frac{1}{2} x^T x \right),
\]

and the optimal bandwidth [49]:

\[
\sigma = n^{-\frac{1}{6}} \sqrt{\frac{1}{2} \hat{\sigma}^T \hat{\sigma}},
\]

where \( \hat{\sigma} \) is the marginal standard deviation vector of \( L_u \).

Thus, given \( u \)'s set of visited locations \( L_u = \{l_1, l_2, \ldots, l_n\} \), the probability of user \( u \) visiting a new location \( l \) can be computed by:

\[
p(l|L_u) = \frac{1}{2\pi n\sigma^2} \sum_{i=1}^{n} \exp\left( -\frac{1}{2\sigma^2} (l - l_i)^T (l - l_i) \right). \]

**Computational complexity.** Algorithm 1 outlines the process for computing \( p(l|L_u) \) through Equation (4). Algorithm 1 calculates a probability
Algorithm 1 The exact computation of $p(l|L_u)$

**Input:** $u$’s set of visited locations $L_u = \{l_1, l_2, \ldots, l_n\}$.

**Output:** $p(l|L_u)$ for each location $l \in L - L_u$.

1: Compute the bandwidth $\sigma$ using Equation (3)
2: for each unvisited location $l \in L - L_u$ do
3: \quad $z \leftarrow 0$ // Initializing auxiliary variable $z$
4: \quad for each $l_i \in L_u$ do
5: \quad \quad $z \leftarrow z + \exp(- (1 - l_i)^T (1 - l_i)/(2\sigma^2))$
6: \quad end for
7: \quad $p(l|L_u) \leftarrow z/(2\pi n\sigma^2)$
8: end for

for each unvisited location $l \in L - L_u$ in order to make location recommendations for $u$ by returning the top-$k$ locations with the highest probability. It is required to evaluate the sum of $n$ Gaussian $\exp(-(1 - l_i)^T (1 - l_i)/(2\sigma^2))$ at each target point $l \in L - L_u$, the computational complexity of which is $O(|L - L_u|n) = O(|L|n)$ in which $|L| \gg |L_u| = n$ since users only checked in a small fraction of locations.

2.2. An Efficient Approximation Approach to Infer Users’ Preferences

In Algorithm 1 with the computational complexity $O(|L|n)$, when users have checked in many POIs (i.e., with a large value of $n$), the calculation on recommending top-$k$ POIs to these users is prohibitively expensive. In this section, we approximately compute $p(l|L_u)$ through the fast Gauss transform [20] to reduce its complexity to be independent of $n$, i.e., $O(|L|)$. The fast Gauss transform is an optimized variant of the fast multipole method.
for Gaussian distributions. As for another kernel functions, we can employ the more general fast multipole method [19].

**Fast Gauss transform.** The fast Gauss transform shifts a Gaussian $\exp(-\frac{1}{2\sigma^2}(1-1_i)^T(1-1_i)/(2\sigma^2))$ centered at $1_i = (lat_i, lon_i)^T$ to a sum of Hermite polynomials centered at $1_0 = (lat_0, lon_0)^T$ by the Hermite expansion, given by:

$$
\exp\left(-\frac{1}{2\sigma^2}(1-1_i)^T(1-1_i)\right) = \sum_{q=0}^{c-1} \frac{1}{(q!)^2} \left(\frac{1_i - 1_0}{\sqrt{2\sigma}}\right)^q h_q\left(\frac{1-1_0}{\sqrt{2\sigma}}\right) + \varepsilon(c), \tag{5}
$$

where

$$
\left(\frac{1_i - 1_0}{\sqrt{2\sigma}}\right)^q = \left(\frac{lat_i - lat_0}{\sqrt{2\sigma}}\right)^q \left(\frac{lon_i - lon_0}{\sqrt{2\sigma}}\right)^q, \tag{6}
$$

$$
h_q\left(\frac{1-1_0}{\sqrt{2\sigma}}\right) = h_q\left(\frac{lat_i - lat_0}{\sqrt{2\sigma}}\right) h_q\left(\frac{lon_i - lon_0}{\sqrt{2\sigma}}\right), \tag{7}
$$

the Hermite functions $h_q(x)$ are defined by:

$$
h_q(x) = (-1)^q \frac{d^q}{dx^q} e^{-x^2}, \tag{8}
$$

and $\varepsilon$ is the error due to truncating the infinite series after $c$ terms. When $l_i = (lat_i, lon_i)^T$ closes to $l_0 = (lat_0, lon_0)^T$, a small value of $c$ is enough to guarantee that the error $\varepsilon$ is negligible as these terms converge to zero quickly.

**Efficient approximation algorithm.** At first, to ensure $l_i$ being close to $l_0$, the geographical space is subdivided into a set of cells $\Omega$ using a uniform grid, and each cell $B \in \Omega$ with center $l_B$ and size $\sigma$, i.e., the optimal bandwidth in Equation (3), as depicted in Figure 3. Accordingly, each location $l_i \in L_u$ is shifted into the center $l_B$ of the cell $B$ in which $l_i$ lies.
Thus, the approximation of \( p(l|L_u) \) in Equation (4) is given by:

\[
p(l|L_u) = \frac{1}{2\pi n\sigma^2} \sum_{i=1}^{n} \exp\left(-\frac{1}{2\sigma^2}(1 - l_i)^T(1 - l_i)\right)
\]

\[
\approx \frac{1}{2\pi n\sigma^2} \sum_{B \in \Omega} \sum_{l_i \in B} \sum_{q=0}^{c-1} \frac{1}{(q!)^2} \left(\frac{l_i - l_B}{\sqrt{2\sigma}}\right)^q h_q \left(\frac{1 - l_B}{\sqrt{2\sigma}}\right) h_q \left(\frac{1 - l_B}{\sqrt{2\sigma}}\right)
\]

\[
= \frac{1}{2\pi n\sigma^2} \sum_{B \in \Omega} \sum_{q=0}^{c-1} \frac{1}{(q!)^2} \sum_{l_i \in B} \left(\frac{l_i - l_B}{\sqrt{2\sigma}}\right)^q h_q \left(\frac{1 - l_B}{\sqrt{2\sigma}}\right)
\]

\[
= \frac{1}{2\pi n\sigma^2} \sum_{B \in \Omega} \sum_{q=0}^{c-1} A_q(B) h_q \left(\frac{1 - l_B}{\sqrt{2\sigma}}\right)
\]

(9)

together with

\[
A_q(B) = \frac{1}{(q!)^2} \sum_{l_i \in B} \left(\frac{l_i - l_B}{\sqrt{2\sigma}}\right)^q.
\]

(10)

**Computational complexity.** Algorithm 2 summarizes the overall process for approximating \( p(l|L_u) \) through Equation (9). \( A_q(B) \) is independent of any target point \( l \in L - L_u \) and only required to be computed once for
Algorithm 2 The efficient approximation of $p(l|L_u)$

**Input:** $u$’s set of visited locations $L_u = \{l_1, l_2, \ldots, l_n\}$ and a constant $c$.

**Output:** $p(l|L_u)$ for each location $l \in L - L_u$.

1: Compute the bandwidth $\sigma$ using Equation (3)
2: Subdivide the geographical space into a uniform grid
3: Compute common items $A_q(B)$ for each combination of $q$ and cell $B \in \Omega$ in terms of Equation (10)
4: for each unvisited location $l \in L - L_u$ do
5: $z \leftarrow 0$ // Initializing auxiliary variable $z$
6: for each cell $B \in \Omega$ that contains $l$ or is immediately adjacent to $l$ do
7: for $q = 0$ to $c - 1$ do
8: $z \leftarrow z + A_q(B)h_q((l - l_B)/(\sqrt{2}\sigma))$
9: end for
10: end for
11: $p(l|L_u) \leftarrow z/(2\pi n\sigma^2)$
12: end for

each $l_i \in L_u$ in advance, which avoids to separately calculate the Gaussian $\exp(-(l - l_i)^T(l - l_i)/(2\sigma^2))$ for each combination of $l \in L - L_u$ and $l_i \in L_u$ in Algorithm 1. The pre-computation step requires $O(n)$ work. The approximation step computes $p(l|L_u)$ using $A_q(B)$ (Lines 4 to 12) and it needs $O(c|L - L_u||\Omega|) = O(|L - L_u|)$ work, in which $c$ and $|\Omega|$ are a small constant relative to $|L - L_u|$. Therefore, the computational complexity of Algorithm 2 is $O(n) + O(|L - L_u|) = O(|L|)$. In addition, to accelerate the evaluation of Equation (9), we also can cut off the sum over the set of all cells $\Omega$ by only considering the nearest cells of $l$ (Line 6), since the cells far away from
I contribute a little to $p(l|L_u)$. For example, consider Figure 3 again. Given $l$ lying in the red cell, we may just consider the red cell and the nearest blue cells around it when estimating $p(l|L_u)$ based on Equation (9).

3. Integrating Geographical and Social Influences

In this section, we integrate the geographical influence with the social influence in the conventional social networks in order to gain better user preference model and quality of location recommendations.

3.1. Using Social Influence

To use the social influence, we design a new method to measure the similarity between users based on their social friendships and residence distance, because nearby friends share more commonly visited locations than others [11]. Specifically, we transform the residence distance of users with social friendships into a normalized similarity through a sigmoid function that is widely used to map an output into a probability. Formally, let $F(u)$ be a set of users having social friendships with $u$ and $distance(u, u')$ be the geographical distance between the residences of users $u$ and $u'$. If $u' \in F(u)$, the similarity between users $u$ and $u'$ is defined by the sigmoid function as follows:

$$sim(u, u') = \frac{2}{1 + \exp(distance(u, u'))}.$$  \hspace{1cm} (11)

Otherwise, $sim(u, u') = 0$. Obviously, we have $sim(u, u') \in [0, 1]$.

Based on the similarity, we can predict the rating $\hat{r}_{u,l}$ of a user $u$ to a new location $l$ via the conventional user-based collaborative filtering technique:

$$\hat{r}_{u,l} = \frac{\sum_{u' \in U} sim(u, u') \cdot r_{u', l}}{\sum_{u' \in U} sim(u, u')}.$$  \hspace{1cm} (12)
where \( r_{u',l} \) denotes the frequency of user \( u' \) visiting location \( l \).

### 3.2. The Union Fusion Rule

Here we consider the integration of \textit{the geographical influence} with \textit{the social influence} to enhance user preference model and improve the quality of location recommendations. An easy way is to apply the simple product or sum rule to combine the visiting probability in Equation (9) and rating in Equation (12) of a user regarding a location into a unified score. However, the unified score from the product rule is dominated by a smaller value of either the visiting probability \( p(l|L_u) \) or the estimated rating \( \hat{r}_{u,l} \). As an example, supposing \( p(l|L_u) = 0.9 \) and \( \hat{r}_{u,l} = 0.1 \), the unified score given by the product rule is 0.09 that is nearly independent of \( p(l|L_u) \) with a large value but strongly dominated by \( \hat{r}_{u,l} \) with a small value. On the other hand, it is considerably hard to determine the relative weights of \( p(l|L_u) \) and \( \hat{r}_{u,l} \) for the sum rule to achieve the best performance. Although the weights can be learned from data, they cost much effort to find out their optimal settings and usually suffer from over-fitting the data.

To this end, we design a new fusion rule, called the union rule, instead of applying the simple product and sum rules. At first, the union rule values the geographical influence and social influence equally important under the condition of the lack of the prior knowledge about their effect. That is, the predicted probability in Equation (9) and rating in Equation (12) are transformed into a normalized probability with the same scale using

\[
\hat{p}(l|L_u) = \frac{p(l|L_u)}{\max_{l \in (L-L_u)} p(l|L_u)}, \tag{13}
\]
and
\[ \hat{p}_{u,1} = \frac{\hat{r}_{u,1}}{\max_{l \in (L - L_u)} \hat{r}_{u,l}}, \tag{14} \]
where \( \max_{l \in (L - L_u)} p(l|L_u) \) and \( \max_{l \in (L - L_u)} \hat{r}_{u,l} \) are the normalization constants for a given user \( u \). Further, the union rule views the difference between the probability and rating given by Equations (13) and (14) as an uncertain indicator and utilizes it to penalize the average of them in order to get a unified score. The underlying reason is that the difference between them usually implies the confidence on the average of them; the larger the difference is, the lower the confidence will be. Concretely, the union rule estimates the mean \( \hat{s}_{u,1}^\mu \) and standard deviation \( \hat{s}_{u,1}^\xi \) by:
\[
\hat{s}_{u,1}^\mu = \frac{\hat{p}(l|L_u) + \hat{p}_{u,1}}{2},
\tag{15}
\]
Together with
\[
\hat{s}_{u,1}^\xi = \sqrt{(\hat{p}(l|L_u) - \hat{s}_{u,1}^\mu)^2 + (\hat{p}_{u,1} - \hat{s}_{u,1}^\mu)^2}.
\tag{16}
\]
Then it penalizes the mean \( \hat{s}_{u,1}^\mu \) with the standard deviation \( \hat{s}_{u,1}^\xi \) to obtain the final score \( \hat{s}_{u,1} \) of user \( u \) visiting location \( l \), given by
\[
\hat{s}_{u,1} = \hat{s}_{u,1}^\mu - \hat{s}_{u,1}^\xi. \tag{17}
\]
It is important to note that: (1) The union rule in Equation (17) addresses the limitations of the simple product and sum rules, since in the union rule neither the probability nor the rating could dominate the final score and there is no need to derive their relative weights. (2) If the difference between the probability and rating determined by Equations (13) and (14), respectively, is large, the final score given by the union rule will be a little far from their
Algorithm 3: The fusion of the geographical and social influences

Input: $u$’s set of visited locations $L_u = \{l_1, l_2, \ldots, l_n\}$.

Output: $\hat{s}_{u,l}$ for each location $l \in L - L_u$.

1: for each unvisited location $l \in L - L_u$ do
2: Compute the visiting probability of $u$ to $l$ by Algorithm 2
3: Compute the similarity of $u$ with social friends by Equation (11)
4: Compute the rating of $u$ to $l$ by Equation (12)
5: Normalize the probability and rating by Equations (13) and (14)
6: Estimate the mean and standard deviation of the normalized probability and rating by Equations (15) and (16)
7: Compute the final score of $u$ to $l$ by Equation (17)
8: end for

average due to the penalty. (3) It is easy to show that the final score $\hat{s}_{u,l}$ lies in $[0,1]$. We summarize the fusion process of the geographical influence with the social influence in Algorithm 3.

4. Experimental Evaluation

In this section, we describe our experiment settings for evaluating the performance of CoRe against the state-of-the-art geo-social recommendation techniques. Specifically, our evaluation focuses on three aspects of CoRe: (1) its recommendation accuracy (i.e., precision and recall); (2) its approximation error in comparison to the exact method; and (3) its recommendation efficiency.
Table 2: Statistics of the two data sets

<table>
<thead>
<tr>
<th></th>
<th>Foursquare</th>
<th>Gowalla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of users</td>
<td>11,326</td>
<td>196,591</td>
</tr>
<tr>
<td>Number of locations (POIs)</td>
<td>182,968</td>
<td>1,280,969</td>
</tr>
<tr>
<td>Number of check-ins</td>
<td>1,385,223</td>
<td>6,442,890</td>
</tr>
<tr>
<td>Number of social links</td>
<td>47,164</td>
<td>950,327</td>
</tr>
<tr>
<td>User-location matrix density</td>
<td>$2.3 \times 10^{-4}$</td>
<td>$2.9 \times 10^{-5}$</td>
</tr>
<tr>
<td>Avg. No. of visited POIs per user</td>
<td>42.44</td>
<td>37.18</td>
</tr>
<tr>
<td>Avg. No. of check-ins per location</td>
<td>2.63</td>
<td>3.11</td>
</tr>
</tbody>
</table>

4.1. Data Sets

We use two publicly available large-scale real check-in data sets\(^1\) that were crawled from Foursquare [15] and Gowalla [11]. The statistics of the data sets are shown in Table 2. Figure 4 depicts the distribution of the locations in the data sets on a world map.

4.2. Evaluated Recommendation Techniques

The recommendation techniques implemented in our experiments are classified into three categories based on the information they use in the location recommendations.

\(^1\)The check-in data sets used for our experiments can be downloaded from [http://www.public.asu.edu/~hgao16/Publications.html](http://www.public.asu.edu/~hgao16/Publications.html) and [http://snap.stanford.edu/data/loc-gowalla.html](http://snap.stanford.edu/data/loc-gowalla.html).
Figure 4: Distribution of locations on a world map

(1) Geographical category:

- MGM: This technique models the distance between visited locations and centers (i.e., the most popular locations) as a multi-center Gaussian
distribution [9].

- **PD**: This technique models the distance between every pair of locations visited by the same user as a power-law distribution [56].

- **iGSLR**: This technique models the distance between every pair of locations visited by the same user as a nonparametric distribution [62].

- **CoRe**: Our technique models a personalized two-dimensional check-in probability density over the geographic coordinates for each user using Algorithm 2.

- **Exact**: The only difference between Exact and CoRe is that Exact computes the exact probability of a user visiting a new location using Algorithm 1, while CoRe uses the approximation method to compute the visiting probability.

(2) **Social category:**

- **SCF**: This technique measures the similarity between users by using their social friendships and then employs the social collaborative filtering technique [5].

- **CoRe**: Our technique measures the similarity between users based on their social friendships and residence distance in Equation (11) and also utilizes the collaborative filtering technique in Equation (12).

(3) **Fusion category:**

- **Prod**: This fusion technique applies the product rule to integrate the geographical and social influences.
• **Sum:** This technique uses the sum rule to integrate the geographical and social influences. It is worth mentioning that before applying the sum rule, the probability and rating have been normalized using Equations (13) and (14), respectively, which is different from our previous work [62] and improves the performance of the sum rule.

• **CoRe:** Our technique applies the union rule to integrate the geographical and social influences based on Algorithm 3.

Note that we implement a different version of CoRe for each category using the corresponding influence to achieve fair comparisons in Section 5.

### 4.3. Performance Metrics

**Recommendation accuracy.** In general, recommendation techniques compute a score for each candidate item (i.e., a location or POI in this paper) regarding a target user and return locations with the top-k highest scores as a recommendation result to the target user. To evaluate the quality of location recommendations, it is important to find out how many recommended locations are actually visited by the target user in the testing data set. Also, it is important to know how many of the locations actually visited were recommended by the evaluated technique. The former aspect is captured by precision and the latter by recall. Precision and recall are standard metrics used to evaluate the recommendation accuracy [9, 56]. We define a discovered location as a location that is both recommended and actually visited by the target user. Formally,
• Precision defines the ratio of the number of discovered locations to the total number \( k \) of recommended locations, i.e.,

\[
\text{precision} = \frac{\text{the number of discovered locations}}{k}. \tag{18}
\]

• Recall defines the ratio of the number of discovered locations to the total number of actually visited locations by the target user in the testing set, i.e.,

\[
\text{recall} = \frac{\text{the number of discovered locations}}{\text{the number of visited locations}}. \tag{19}
\]

**Approximation error.** We evaluate the approximation error of CoRe by comparing its recommendation accuracy with that of Exact (i.e., Algorithm 1). Note that we are more concerned about the effect of approximation error on the recommendation accuracy than the error itself.

**Recommendation efficiency.** We compare the running time of CoRe and Exact with respect to various numbers of check-in locations of users. All algorithms were implemented in Matlab and run on a machine with 3.4GHz Intel Core i7 Processor and 16GB RAM.

### 4.4. Experiment Settings

We split each data set into the training set and the testing set in terms of the check-in time rather than using a random partition method, because in practice we can only utilize the past check-in data to predict the future check-in events. A half of check-in data with earlier timestamps are used as the training set and the other half of check-in data are used as the testing set. In CoRe, unless otherwise specified, in Algorithm 2 the cell size is set to
the optimal bandwidth in Equation (3) and the number of truncated terms in Equation (5) is set to a small value: $c = 6$.

5. Experimental Results

This section analyzes our extensive experimental results. We first compare our CoRe against the state-of-the-art geo-social recommendation techniques in terms of the recommendation accuracy (Section 5.1). We then study the approximation error of CoRe (Section 5.2). Finally, we evaluate the recommendation efficiency of CoRe (Section 5.3).

5.1. Recommendation Accuracy

Here we separately investigate the performance of three new techniques proposed in this paper, including the efficient approximation approach to exploit the personalized two-dimensional geographical influence in Algorithm 2 (Section 5.1.1), the method to measure the similarity of users in Equation (11) (Section 5.1.2), and the union rule to integrate the geographical and social influences in Algorithm 3 (Section 5.1.3). Moreover, we discuss the effect of the number $k$ of recommended locations for users (Section 5.1.4), the number $n$ of check-in locations of users (Section 5.1.5), and the cell size on the recommendation accuracy (Section 5.1.6).

5.1.1. Geographical Recommendation Techniques

Figures 5 and 6 compare the recommendation accuracy of CoRe (i.e., Algorithm 2), iGSLR [62], PD [56] and MGM [9] with the effect of the number of recommended locations for users (top-$k$) and the number of check-in locations of users (given-$n$) using two large-scale real data sets collected from
Figure 5: Comparison of geo-techniques on the number of recommended locations $k$ (CoRe is Algorithm 2)
Figure 6: Comparison of geo-techniques on the number of check-in locations $n$ (CoRe is Algorithm 2)
Foursquare and Gowalla, respectively. Note that $k$ is set to a smaller number than $n$ because a large number of recommended locations may not be helpful for users. As a whole, our proposed method CoRe always exhibits the best precision and recall for all the values of $k$ and $n$. The details are demonstrated as follows.

**MGM** models the distance between a location and a center as a universal multi-center Gaussian distribution for all users. Actually, its estimated probability of $p(l|L_u)$ is independent of user $u$ or $L_u$, that is, it recommends the same set of locations for all users. As a result, it performs the worst.

**PD** models the distance between every pair of locations visited by the same user as a power-law distribution for all users. Although it enhances the performance of recommending locations in comparison to MGM, it still inherits the limitation of the universal distance distribution for all users.

**iGSLR** estimates an individual distance distribution for each user and then improves the recommendation accuracy compared to PD, but the improvement is very limited since it still simplifies the geographical influence as one-dimensional distance distributions just as in MGM and PD.

**CoRe** estimates a personalized two-dimensional check-in probability density over the geographic coordinates for each user and always shows the best recommendation quality in terms of precision and recall. These results verify the superiority of exploiting the personalized two-dimensional geographical influence for location recommendations.

### 5.1.2. Social Recommendation Techniques

Figures 7 and 8 contrast the recommendation accuracy of CoRe (i.e., Equation (12)) with that of SCF [5] on the Foursquare and Gowalla data.
Figure 7: Comparison of social techniques on the number of recommended locations $k$ ($\text{CoRe}$ is Equation (12))

sets. By transforming the residence distance and social friendships of users into a normalized similarity through a sigmoid function in Equation (11), $\text{CoRe}$ is able to reflect the fact that nearby friends share more commonly visited locations than others. As a result, $\text{CoRe}$ performs better than $\text{SCF}$ in terms of both precision and recall.

On the other hand, both $\text{CoRe}$ and $\text{SCF}$ suffer from the effect of data
Figure 8: Comparison of social techniques on the number of check-in locations $n$ (CoRe is Equation (12))

sparsity. For instance, their precision and recall in the Gowalla data set are obviously lower than those in the Foursquare data set, because the density of the Gowalla data set is one order-of-magnitude lower than that of the Foursquare data set as shown in Table 2.
5.1.3. Fusion Rules

Figures 9 and 10 depict the performance of three fusion rules for integrating geographical and social influences, including the product rule (Prod), the sum rule (Sum), and the union rule (CoRe, i.e., Algorithm 3).

Prod and Sum are comparable to each other on both Foursquare and Gowalla data sets and they improve the performance of the geo-social rec-
Figure 10: Comparison of fusion rules on the number of check-in locations $n$ (CoRe is Algorithm 3)

ommendation techniques only using either geographical information or social information in Sections 5.1.1 or 5.1.2, respectively. Promisingly, using the union rule that considers the difference of two scores in Equations (13) and (14) given by the geographical and social recommendation techniques respectively as an uncertain indicator to penalize the average of them, CoRe can further raise the precision and recall in comparison to Prod and Sum.
5.1.4. Effect of Number of Recommended Locations

Figures 5, 7 and 9 describe the recommendation accuracy of a variety of recommendation techniques with respect to the change of the number of recommended locations, i.e., $k$ from 5 to 50. As $k$ increases, the precision gradually gets lower but the recall steadily becomes higher on the two data sets. Our explanation is that, in general, by returning more locations for users, it is able to discover more locations that users would like to visit. However, the extra recommended locations are less possible to be liked by users due to the lower visiting probabilities of these locations.

5.1.5. Effect of Number of Check-in Locations

Figures 6, 8 and 10 depict the recommendation accuracy of a variety of recommendation techniques with regard to various numbers of check-in locations of users, i.e., $n$ from 10 to 100, on the two data sets. As users visit more locations, more check-in data are available for these recommendation techniques; hence, they can more accurately estimate the scores of these users for new locations. Accordingly, the precision inclines. Nonetheless, users who have checked in many locations in the training data set usually have visited many locations in the testing data set, so the recall fluctuates.

5.1.6. Effect of Cell Sizes

Heretofore, the cell size in CoRe is set to the optimal bandwidth $\sigma$ in Equation (3). Figure 11 depicts the effect of cell sizes on the recommendation accuracy of CoRe in the two data sets. (1) In general, CoRe achieves the best precision and recall when the cell size equals the optimal bandwidth $\sigma$. (2) With the decrease of the cell size, the recommendation accuracy
of CoRe gradually declines. The reason is that: For efficiency CoRe only takes into account the nearest cells of a location when computing its visiting probability according to Algorithm 2 and a smaller cell size leads to an end that more visited locations out of the nearest cells are ignored. (3) As the cell size increases, the recommendation accuracy of CoRe decreases as well. Our explanation is that: A larger cell size accordingly requires a larger number of truncated terms in Equation (5) in order to complete the same approximation.
5.2. Approximation Error

Figure 12 compares the approximation error of CoRe (Algorithm 2) with that of Exact (Algorithm 1) regarding the change of the number of truncated terms in Equation (5) in terms of the recommendation accuracy rather than the error itself, since the error itself is not meaningful for location recom-
mendations and the effect of approximation error on the recommendation accuracy is more significant.

As shown in Figure 12, with the increase of the number of truncated terms, the precision and recall of CoRe quickly rise and approach to the performance of Exact in both the Foursquare and Gowalla data sets. The reason is that the approximation error caused by truncating the infinite series with the first $c$ terms descends exponentially. For instance, in previous experiments, the default value of $c$ is set to 6, in which the caused approximation error is negligible according to the experimental results depicted in Figure 12.

5.3. Recommendation Efficiency

Figure 13 depicts the running time of CoRe (Algorithm 2), Exact (Algorithm 1), MGM, PD and iGSLR with respect to various numbers of check-in locations of users. (1) With the increase of the number of check-in locations of users, CoRe maintains a low consistent running time while Exact takes a linearly increasing running time, since CoRe has the computational complexity of $O(|L|)$ and Exact with $O(|L|n)$. (2) Although the running time of MGM also remains constant because of its computational complexity of $O(M|L|)$ in which $M$ is the number of centers (i.e., the most popular POIs), MGM requires more running time than CoRe. (3) As the number of check-in locations of users gets larger, the running time of both PD and iGSLR with $O(|L|n)$ also increases linearly, but increasing faster than that of Exact, not to mention CoRe.

Therefore, CoRe is more scalable to the web-scale calculation in the process of location recommendations. In addition, all evaluated techniques take more time to recommend locations in the Gowalla data set than the
Figure 13: Recommendation efficiency of CoRe (Algorithm 2) in comparison to Exact (Algorithm 1), MGM, PD and iGSLR.
Foursquare data set, because the former contains more location candidates for recommendations than the latter as shown in Table 2.

6. Related Work

In this section, we highlight related work about location recommendations in location-based social networks [12].

**Mobile context-aware recommender systems.** Location recommendations have been widely studied in mobile context-aware recommender systems for tourism [18]. In addition to user locations, potentially useful context information contains mobility trajectory, distance from POIs, budget, time, weak day, season, time available for sightseeing, transportation mode, weather condition, etc. For example, to make location recommendations, Barranco et al. [4] employed users’ mobility trajectories and speeds, Savage et al. [47] utilized users’ transportation modes and moods, and the works [1, 17] focus on using time of a day and weather conditions. However, most these studies do not take into account the social interactions between users.

**Social recommendations.** With the rapid growth of social networks, like Facebook and Twitter, social network information, e.g., personal profile content, tags and friendships, has been utilized to improve the quality of recommender systems. Such information can be considered as implicit user feedbacks to alleviate the data sparsity problem of explicit user feedbacks (i.e., the rating matrix) in traditional recommender systems [35, 50]. In particular, recommendation techniques integrating social friendships have been widely studied [5], including model-based methods [13, 25, 26, 27, 33, 35, 38, 41, 43, 48, 53, 54] and memory-based methods [24, 28]. The rationale behind these
methods is that friends are more likely to share common interests and thus the social influence should be considered when making recommendations.

**Location recommendations in LBSNs.** Some recent studies provide POI recommendations by using the conventional collaborative filtering techniques on users’ check-in data [14, 21, 45, 64, 65, 70], GPS trajectory data [8, 23, 30, 40, 58, 66, 67, 68, 69], or text data [22, 37, 51, 52]. However, these studies have not leveraged any geographical influence when generating recommendations. In reality, the geographical information of users and locations plays a significant influence on users’ check-in behaviors [9, 56], since physical interactions are required for users to visit locations that are totally different from other non-spatial items, e.g., books, music and movies [56].

**Location recommendations using geographical influence.** To exploit geographical influence for improving the quality of location recommendations, some techniques [15, 16, 31, 55, 59, 61] employ the geographical influence of users to derive their similarity weights as an input of the conventional collaborative filtering techniques [6, 7]. However, the performance is considerably limited due to no consideration for the geographical influence of locations. In contrast, other techniques explore the geographical influence of locations. For example, the studies [2, 10, 44, 57] view locations as ordinary non-spatial items and consider the geographical influence of locations by predefining a range; locations only within this range will be possibly recommended to users. The literatures [29, 32, 34] present a geo-topic model respectively by assuming that if a location is closer to the locations visited by a user or the current location of a user, it is more likely to be visited by the same user. More sophisticatedly, some works model the distance between
two locations visited by the same user as a common distribution for all users, e.g., a power-law distribution [36, 39, 56, 60] or a multi-center Gaussian distribution [9]. In particular, our previous papers [62, 63] personalize the geographical influence by modeling the distance between locations visited by the same user as a personalized distance distribution for each user.

7. Conclusion

In this paper, we have proposed an effective and efficient location recommendation framework called CoRe. In CoRe, we have modeled the personalized two-dimensional geographical influence and employed the kernel density estimation technique to derive the personalized two-dimensional check-in probability density for a user visiting a new location. To improve the system efficiency and scalability, we have used the fast Gauss transform to develop the approximation method to compute visiting probabilities. In addition, we have also designed a new method to measure the similarity between users based on their social friendships and residence information and a new fusion rule to combine the geographical and social influences into a unified score. Finally, we have conducted extensive experiments to evaluate the recommendation accuracy, recommendation efficiency, and approximation errors of CoRe using two data sets from Foursquare and Gowalla. Experimental results show that CoRe outperforms all other evaluated recommendation techniques.
References


[7] F. Cacheda, V. Carneiro, D. Fernández, V. Formoso, Comparison of collaborative filtering algorithms: Limitations of current techniques and


[13] P. Cui, F. Wang, S. Liu, M. Ou, S. Yang, L. Sun, Who should share what?: Item-level social influence prediction for users and posts ranking, in: Proceeding of the 34th International ACM SIGIR Conference on
Research and Development in Information Retrieval, Beijing, China, 2011, pp. 185–194.


[27] M. Jiang, P. Cui, F. Wang, W. Zhu, S. Yang, Recommendation with
social contextual information, IEEE Transactions on Knowledge and Data Engineering (2014) accepted to appear.


D. Yang, D. Zhang, Z. Yu, Z. Yu, Fine-grained preference-aware location search leveraging crowdsourced digital footprints from LBSNs, in:


