iGeoRec: A Personalized and Efficient Geographical Location Recommendation Framework

Jia-Dong Zhang, Chi-Yin Chow, *Member, IEEE*, and Yanhua Li, *Member, IEEE*

Abstract—Geographical influence has been intensively exploited for location recommendations in location-based social networks (LBSNs) due to the fact that geographical proximity significantly affects users' check-in behaviors. However, current studies only model the geographical influence on all users' check-in behaviors as a *universal* way. We argue that the geographical influence on users' check-in behaviors should be *personalized*. In this paper, we propose a personalized and efficient geographical location recommendation framework called iGeoRec to take full advantage of the geographical influence on location recommendations. In iGeoRec, there are mainly two challenges: (1) personalizing the geographical influence to accurately predict the probability of a user visiting a new location, and (2) efficiently computing the probability of each user to all new locations. To address these two challenges, (1) we propose a probabilistic approach to personalize the geographical influence as a personal distribution for each user and predict the probability of a user visiting any new location using her personal distribution. Furthermore, (2) we develop an efficient approximation method to compute the probability of any user to all new locations; the proposed method reduces the computational complexity of the exact computation method from $O(|L|n^3)$ to $O(|L|n)$ (where $|L|$ is the total number of locations in an LBSN and n is the number of check-in locations of a user). Finally, we conduct extensive experiments to evaluate the recommendation *accuracy* and *efficiency* of iGeoRec using two large-scale real data sets collected from the two of the most popular LBSNs: Foursquare and Gowalla. Experimental results show that iGeoRec provides significantly superior performance compared to other state-of-the-art geographical recommendation techniques.

Index Terms—Location-based social networks, location recommendations, probabilistic approach, personalized geographical influence, efficient approximation

1 Introduction

Recently with the emergence of location-based social networks (LBSNs) as shown in Fig. 1, like Foursquare and Gowalla, it is prevalent to recommend some specific locations for users (e.g., [1], [2], [3], [4], [5], [6], [7], [8], [9]), which not only helps users explore new places but also makes LBSNs more attractive to users. These spatial locations are also known as *points-of-interest* (POIs), e.g., restaurants, stores, and museums, and are distinct from other non-spatial items, such as books, music and movies in conventional recommendation systems [10], [11], because physical interactions are required for users to visit or check in locations [8]. Thus, **the geographical information of users and locations plays a significant influence on users' check-in behaviors** [3], [8], known as ***geographical influence*** for short, which has been intensively exploited to make location recommendations for users.

A simple way is to utilize **the geographical influence of users**, i.e., the distance between the residences of users, to recommend locations visited by nearby users, because they share more commonly check-in locations than others living far away [5], [6], [7], [9]. However, users often travel from one place to another and hence their static residences may not reflect their actual geographical positions. As a result, the improvement on the quality of location recommendations is relatively limited by only considering the geographical information of users' residences.

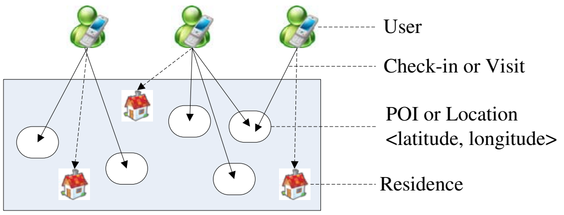


Fig. 1. A location-based social network

A better way is to exploit **the geographical influence of locations**, i.e., the distance between every pair of locations visited by the same user, to model all users' check-in behaviors. The major research direction assumes that the distance of visited locations follows a power-law distribution (PD) [8], [12], [13], [14], [15], [16], where the model parameters are derived from the whole check-in history of all users. Another research direction is to cluster the whole check-in history of all users to find the most popular locations as centers and assume that the distance

- *J.-D. Zhang and C.-Y. Chow are with Department of Computer Science, City University of Hong Kong, Hong Kong. Y. Li is with HUAWEI Noah's Ark Lab, Hong Kong. E-mail: jzhang26-c@my.cityu.edu.hk, chiychow@cityu.edu.hk, Li.Yanhua1@huawei.com. J.-D. Zhang and C.-Y. Chow were partially supported by a research grant (CityU Project No. 9231131). Y. Li was supported by National Grant Fundamental Research (973 Program) of China under Grant 2014CB340304.*

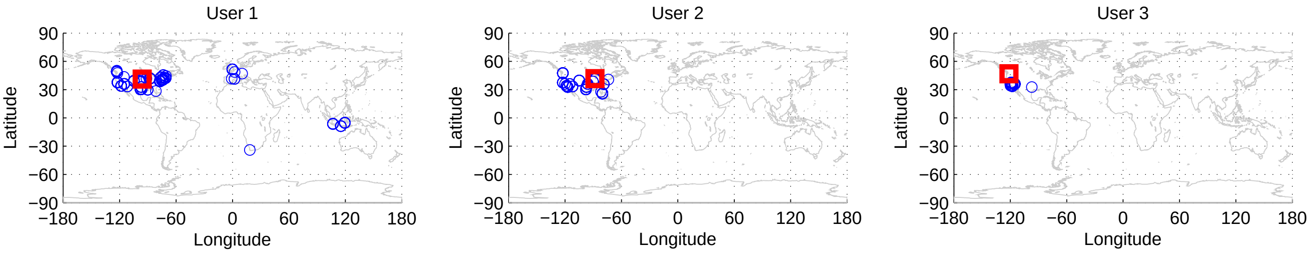


Fig. 2. Distributions of personal check-in locations

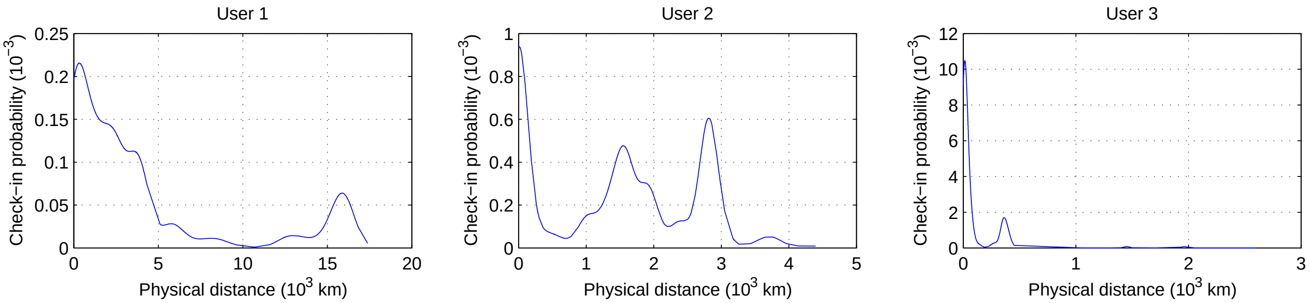


Fig. 3. Personal check-in probabilities over geographical distances

between visited locations and their centers follows a multi-center Gaussian model (MGM) [3]. These two directions use the obtained distribution to deduce the probability of a user visiting a new location that benefits the quality of location recommendations to some extent.

Nonetheless, the geographical influence is universally modeled as a common distribution for all users in [3], [8], [12], [13], [14], [15], [16]. On the contrary, in reality the geographical influence on users' check-in behaviors is unique. For instance, indoorsy persons like visiting locations around their living areas while outdoorsy persons prefer traveling around the world to explore new locations. Therefore, we argue that the influence of geographical information on individual users' check-in behaviors should be personalized when recommending locations for users. In fact, personalization is one of the most essential requirements of recommendation that can help alleviate the problem of information overload and is an important enabler of the success of e-business [17].

We use real-world examples (Fig. 2) to show that a user's check-in behavior is unique. For example, some people like visiting locations around their living areas while others may prefer traveling around the world to explore new locations. To observe these differences, a spatial analysis is conducted on two publicly available real data sets collected from Foursquare [5] and Gowalla [18], that are the two of the most popular LBSNs. Specifically, we focus on three users with the largest number of visited locations of each data set, i.e., the three users have about 300 locations in the Foursquare data set and 1,000 locations in the Gowalla data set. Due to similar result and space limitation, Fig. 2 only shows these three users' check-in locations (represented by blue circles) and residence locations (represented by red squares) in the Foursquare data set. The geographical influence on these three users' check-in behaviors is unique according to Fig. 2: User 1 travels around the world, e.g., North America, Europe, South Africa, and South Asia; User 2 moves around in the United States of America; and User 3 usually visits locations around her living area, i.e., Los Angeles. To further understand the geographical influence on the three users' check-in behaviors, Fig. 3 depicts their individual check-in distance distribution over the distance between every pair of locations visited by the same user or between the user's residence location and her visited locations. Their distance distributions are also unique, so it is undesirable to model them as a universal distribution, e.g., PD [8], [13], [14], [16] and MGM [3].

In this paper, we propose a personalized and efficient geographical location recommendation framework called iGeoRec. In iGeoRec, there are mainly two challenges. The first challenge is to model **the personalized geographical influence** on users' check-in behaviors in order to accurately predict the probability of a user visiting a new location. To this end, rather than deriving a common distribution for all users [3], [8], [13], [14], [16], we model the geographical influence as a personalized distance distribution for each user based on a nonparametric method, i.e., the popular kernel density estimation (KDE) [19]. KDE does not have any assumption about the form of a distance distribution and hence can be used with arbitrary distributions. Then using the personalized distance distributions of users, we develop a probabilistic approach to accurately derive the probability of users to new locations. Eventually, we can make a personalized location recommendation for each user by returning her top-k locations with the highest visiting probability.

The other challenge of iGeoRec is to efficiently compute the probability of each user to all new locations. The exact computation method is to evaluate each location individually, which costs $O(|L - n|n^3) = O(|L|n^3)$ work in all, where $|L|$ is the total number of locations in an LBSN and n is the number of locations that the user has visited; note that $|L| \gg n$ since users only check in a little fraction of locations. Unfortunately, the computational requirement of the exact method grows rapidly with the increase of n, which makes large-scale calculations prohibitively expensive for location recommendations in an LBSN. Therefore, we propose an efficient approximation method to compute the visiting probability of a certain user to all new locations based on the *fast Gauss transform* (FGT) [20], *clustering* [21], and *three-sigma rule of Gaussian distribution* [22]. Our proposed method reduces the computational complexity to $O(|L|n)$.

This study is a significant extension to our previous work [23] by proposing a new probabilistic approach for predicting the visiting probability of a user to a new location, developing a new efficient approximation method for personalizing the geographical influence, and conducting the extensive experiments for all new algorithms. The main contributions of this paper can be summarized as follows:

- We personalize *the geographical influence* on a user's check-in behavior through learning an individual distance distribution from the user's geographical information including her check-in history and her residence. Accordingly, we propose a probabilistic approach to predict the probability of the user visiting any new location based on her personalized distance distribution. (Section 3)
- We develop an efficient approximation method for personalizing the geographical influence. Our efficient approximation algorithm decreases the computational complexity from $O(|L|n^3)$ to $O(|L|n)$ guaranteed by Theorem 1, and has a low upper bound of approximation error as shown in Theorem 2. (Section 4)
- We conduct extensive experiments to evaluate the recommendation *accuracy* and *efficiency* of iGeoRec using two large-scale real data sets collected from Foursquare and Gowalla. Experimental results show that (a) iGeoRec outperforms the state-of-the-art geographical recommendation techniques including PD [8], [13], [14], [16] and MGM [3] in terms of recommendation accuracy, and (b) iGeoRec achieves small approximation errors, but it is significantly faster than the exact method. (Sections 5 and 6)

The remainder of this paper is organized as follows. Section 2 highlights related work. Section 3 describes the proposed probabilistic approach to personalize geographical influence on users' check-in behaviors for predicting the probability of a user visiting a new location. We then present the efficient approximation method for the proposed approach in Section 4. In Sections 5 and 6, we present our experiment settings and analyze the performance of iGeoRec, respectively. Finally, we conclude this paper in Section 7.

2 Related Work

In this section, we highlight related work about location recommendations in LBSNs.

Location recommendations in LBSNs. Some recent studies provide POI recommendations by using the conventional collaborative filtering techniques on users' check-in data [24], [25], GPS trajectory data [26], [27], [28], [29], [30], [31], or text data [32]. 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 [3], [8], [12], [13], [14], [15], [16], 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 [8].

Location recommendations using geographical influence. To exploit geographical influence for improving the quality of location recommendations, some techniques [5], [6], [7], [9] employ the geographical influence of users to derive their similarity weights as an input of the conventional collaborative filtering techniques [11], [33], [34]. 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], [4] 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 literature [35] presents a geo-topic model 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 sophistically, the works [3], [8], [13], [14], [16] model the distance between two locations visited by the same user as a common distribution for all users, e.g., a power-law distribution or a multi-center Gaussian distribution. Nonetheless, in practice geographical influence of locations should be unique for each user.

To this end, we consider that the geographical influence on users' check-in behaviors should be personalized during the recommendation process. In this paper, we are motivated to model the personalized geographical influence of users and locations as a personalized distance distribution for each user based on the kernel density estimation. (Section 3)

TABLE 1
Key notations in this paper

Symbol	Meaning
U	Set of all users in an LBSN
u	Some user and $u \in U$
L	Set of all locations (or POIs) in an LBSN
l	Some location and $l \in L$
L_u	Set of locations visited by user u (i.e., u's check-in locations) and $L_u = \{l_1, l_2, \ldots, l_n\} \subset L$
h_u	Home residence location of user u, also denoted by l_{n+1} for presentation
X_u	Sample of distances between every pair of locations in $h_u \cup L_u$
y_i	Distance between $l_i \in h_u \cup L_u$ and unvisited location l
B	Some element of a partition of X_u
μ_B	Center of B
$p(l\|h_u, L_u)$	Predicted probability of u visiting l given h_u, L_u

Kernel density estimation (KDE) and fast Gaussian transform (FGT). As one of nonparametric methods for estimating probability distributions, KDE has two advantages: (1) it is generally applicable to arbitrary distributions and (2) it requires relatively fewer samples to give a good density estimation than the nonparametric methods based on histograms [19], [36]. The FGT is an important variant of the more general fast multipole method [20] and is successfully applied to accelerate KDE for many applications of pattern recognition [37], [38]. The literature [39] improves the original FGT by alleviating its two serious defects in higher dimensional spaces: the exponential growth of complexity with dimensionality and the uniform gird structure of samples.

In this paper, we further exploit the FGT to efficiently approximate the personalized geographical influence. iGeoRec is different from the previous works [20], [39], since it not only uses the clustering approach [21] to group the samples into boxes, but it also utilizes the three-sigma rule of Gaussian distribution [22] to reduce the number of boxes that are needed to be evaluated. (Section 4)

3 Modeling Geographical Influence

In this section, we define the research problem (Section 3.1), propose a KDE-based approach to personalize geographical influence (Section 3.2), and develop a probabilistic approach to derive the probability of a user visiting a new location (Section 3.3).

3.1 Notations and Problem Statement

TABLE 1 summarizes the key symbols used in this paper. In the problem of location recommendations with the geographical influence, given a user u's home residence location h_u and set of visited locations $L_u = \{l_1, l_2, \ldots, l_n\}$, the goal is to predict the probability of u visiting a new location l, denoted by $p(l|h_u, L_u)$ and then return the top-k locations with the highest visiting probability $p(l|h_u, L_u)$ for u.

3.2 Personalizing Geographical Influence

The experimental results in Section 1 inspire us to study the *personalized geographical influence of users and locations* on an individual user's check-in behavior. In addition, to relax the assumption about the universal form of the distance distribution for all users made in [3] and [8], [13], [14], [16], we apply a general nonparametric technique, known as the kernel density estimation [19] (KDE), which can be used with arbitrary distributions and without the assumption on the form of the underlying distribution. To this end, we model the personalized distribution of the distance between any pair of locations including the user's check-in locations and home residence location using KDE. This process consists of two steps: *distance sample collection* and *distance distribution estimation*.

Step 1: Distance sample collection. This step collects a sample for a user by computing the distance between every pair of locations including the user's check-in locations and home residence location, because each location in LBSNs is associated with its user's identity and position (i.e., latitude and longitude coordinates). Formally, given a user u's home residence location h_u and set of visited locations $L_u = \{l_1, l_2, \ldots, l_n\}$, the sample of distances between every pair of locations in $h_u \cup L_u$, denoted X_u, can be obtained by:

$$X_u = \{x = distance(l_i, l_j) | \forall l_i, l_j \in h_u \cup L_u\}, \tag{1}$$

where $distance(l_i, l_j)$ represents the geographical distance between locations l_i and l_j rather than the actual travel distance of user u from l_i to l_j, since the sampling rate of users' check-in locations is pretty low, from several times a day to one time in several months. Thus, it is meaningless to compute the actual travel distance between two consecutive check-in locations due to the large time gap between them.

Step 2: Distance distribution estimation. The task of kernel density estimation is to estimate a probability density function of an unknown variable based on a known sample. In our case, X_u is the known distance sample and y denotes the unknown distance variable. Then, the probability density function f of distance variable y using sample X_u is given by:

$$f(y) = \frac{1}{|X_u|\sigma} \sum_{x \in X_u} K\left(\frac{y - x}{\sigma}\right), \tag{2}$$

where $|X_u|$ is the number of sample points in X_u and equal to $n(n+1)/2$ according to Equation (1), $K(\cdot)$ is the kernel function and σ is a smoothing parameter, called the bandwidth. In this paper we apply the most popular normal kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \tag{3}$$

and the optimal and small bandwidth [19]:

$$\sigma = \left(\frac{4\hat{\sigma}^5}{3|X_u|}\right)^{1/5} \approx 1.06\hat{\sigma}|X_u|^{-1/5}, \quad (4)$$

where $\hat{\sigma}$ is the standard deviation of the sample X_u.

It is worth emphasizing that the probability density function in Equation (2) based on KDE can fit any real distance distributions (e.g., the power-law distribution) only if X_u contains an adequate number of distance samples. Moreover, KDE requires relatively few distance samples to achieve a good density estimation [19], [36]. In contrast, the parametric methods for estimating the multi-center Gaussian distribution or power-law distribution require much more distance samples. Thus, the multi-center Gaussian model or power-law distribution is not appropriately applied to personalize the geographical influence, i.e., it is hard to accurately estimate the personal multi-center Gaussian distribution or power-law distribution just based on an individual user's distance samples.

3.3 Predicting Probabilities of Users to Locations

In this section based on the obtained distance distribution, we design a method to derive the probability of a user u visiting a new location l given u's home residence location h_u and set of check-in locations $L_u = \{l_1, l_2, \ldots, l_n\}$, denoted as $p(l|h_u, L_u)$.

First, the event of u visiting l consists of all individual events of u visiting l after l_i ($l_i \in h_u \cup L_u$), denoted as $l_i \rightarrow l$, since the check-in behavior of u to l is affected by the geographical influence of all locations in $h_u \cup L_u$. Moreover, the geographical influence of l_i to l is measured by the distance between them. To this end, we compute the distance of every pair of l and the locations in $h_u \cup L_u$:

$$y_i = distance(l_i, l), \forall l_i \in h_u \cup L_u, \quad (5)$$

where $i = 1, 2, \ldots, n+1$ and h_u is denoted as l_{n+1} for simplicity. Then, each y_i can be used to derive the probability of user u visiting location l based on the obtained distance distribution. However, in terms of the probability definition for the continuous distance variable y in Equation (2), the probability of y taking on any single value y_i is always zero, even though it need not be zero. Instead, the probability is computed by the integral of $f(y)$ in an interval centered at y_i based on probability theory. Specifically the probability of user u visiting location l caused by the geographical influence of location l_i, i.e., the probability that the event $l_i \rightarrow l$ occurs, is defined by

$$\begin{aligned} p(l_i \rightarrow l) &= \int_{y_i-\sigma/2}^{y_i+\sigma/2} f(y)dy \approx f(y_i)\sigma \\ &= \frac{1}{|X_u|}\sum_{x\in X_u} K\left(\frac{y_i - x}{\sigma}\right), \end{aligned} \quad (6)$$

where we apply the user-specific interval width σ rather than a fixed interval width for all users, since for any fixed width it is possible to result in the probability larger than one when σ is enough small.

Second, in reality users tend to visit locations close to their homes and also may be interested in exploring the nearby places of their visited locations [8], [18]. This means: a user visiting a new location can be resulting from the geographical influence of only one of her visited locations, if the new location is close enough to the visited one. Actually, it is usually impossible to require that the new location is close to all visited locations, especially when the visited locations are far away from each other. Thus, we compute the probability of u visiting location l caused by the geographical influence of all locations in $h_u \cup L_u$ based on the OR model, given as:

$$p(l|h_u, L_u) = p\left(\bigcup_{i=1}^{n+1}(l_i \rightarrow l)\right) = 1 - p\left(\bigcap_{i=1}^{n+1}\overline{l_i \rightarrow l}\right). \quad (7)$$

Third, in terms of Equation (4), σ is usually small enough to make the correlation between two close locations negligible, so we assume that the events $\overline{l_i \rightarrow l}$ are independent of each other in Equation (7):

$$p(l|h_u, L_u) = 1 - \prod_{i=1}^{n+1} p\left(\overline{l_i \rightarrow l}\right). \quad (8)$$

Finally, based on Equations (3), (6) and (8), we obtain the probability of a user u visiting a new location l given u's home residence location h_u and set of check-in locations $L_u = \{l_1, l_2, \ldots, l_n\}$:

$$\begin{aligned} p(l|h_u, L_u) &= 1 - \prod_{i=1}^{n+1}\left(1 - p(l_i \rightarrow l)\right) \\ &= 1 - \prod_{i=1}^{n+1}\left(1 - \frac{1}{|X_u|\sqrt{2\pi}}\sum_{x\in X_u} e^{-\frac{(y_i-x)^2}{2\sigma^2}}\right). \end{aligned} \quad (9)$$

In Equation (9), the residence location h_u is treated the same as the other n visited locations. Thus, the residence location only accounts for $1/(n+1)$ weight of the total geographical influence that obviously decreases with the increase of n. This self-adjusting weight reflects the fact that as a user checks in more and more locations, the influence of her residence on her check-in behavior gradually decreases, since a user with many check-in locations usually means she has rich travel experiences and her interested locations are more independent of her residence.

Computational complexity. Algorithm 1 outlines the process for computing $p(l|h_u, L_u)$ through Equation (9). Algorithm 1 calculates a probability 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. To compute $p(l|h_u, L_u)$ for a certain location $l \in L - L_u$, it is required to evaluate the sum of $|X_u|$ Gaussians $e^{-(y_i-x)^2/2\sigma^2}$ at $n+1$ target points y_i (Lines 5 to 12), the computational complexity of which is $O(|X_u|n) =$

Algorithm 1 The exact computation of $p(l|h_u, L_u)$

```
Input: u's home residence location $h_u$ and set of visited locations
    $L_u = \{l_1, l_2, \dots, l_n\}$.
Output: $p(l|h_u, L_u)$ for each location $l \in L - L_u$.
Collect the sample $X_u$ using Equation (1)
Compute the bandwidth $\sigma$ using Equation (4)
for each unvisited location $l \in L - L_u$ do
    $z \leftarrow 1$ // Initializing auxiliary variable $z$
    for each $l_i \in h_u \cup L_u$ do
        $y_i \leftarrow distance(l_i, l)$
        $v \leftarrow 0$ // Initializing auxiliary variable $v$
        for each $x \in X_u$ do
            $v \leftarrow v + e^{-(y_i - x)^2/2\sigma^2}$
        end for
        $z \leftarrow z\left[1 - v/(|X_u|\sqrt{2\pi})\right]$
    end for
    $p(l|h_u, L_u) \leftarrow 1 - z$
end for
```

$O(n^3)$ (Note that $O(|X_u|) = O(n^2)$ according to Equation (1)). Thus, the computational complexity of Algorithm 1 is $O(|L - L_u|n^3) = O(|L|n^3)$ in which $|L| \gg |L_u| = n$ since users only check in a small fraction of locations.

4 Efficient Approximation of Personalized Geographical Influence

The computational complexity $O(|L|n^3)$ of Algorithm 1 grows rapidly as n increases that makes large-scale calculations prohibitively expensive. In this section, we approximately compute $p(l|h_u, L_u)$ through the *fast Gauss transform*, *clustering* and *three-sigma rule of Gaussian distribution* to reduce its complexity to $O(|L|n)$.

4.1 Fast Gauss Transform with Clustering

The fast Gauss transform [20] shifts a Gaussian $e^{-(y_i-x)^2/2\sigma^2}$ centered at x to a sum of Hermite polynomials centered at x_0 by the Hermite expansion, given by

$$e^{-\frac{(y_i-x)^2}{2\sigma^2}} = \sum_{q=0}^{c-1} \frac{1}{q!}\left(\frac{x - x_0}{\sqrt{2}\sigma}\right)^q h_q\left(\frac{y_i - x_0}{\sqrt{2}\sigma}\right) + \varepsilon(c), \tag{10}$$

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

$$h_q(x) = (-1)^q \frac{d^q}{dx^q} e^{-x^2}, \tag{11}$$

and ε is the error due to truncating the infinite series after c terms. When x closes to x_0, a small value of c is enough to guarantee that the error ε is negligible as these terms converge to zero quickly.

The formation of boxes B using clustering. To ensure x being close to x_0, we cannot shift all Gaussians $e^{-(y_i-x)^2/2\sigma^2}$ for any $x \in X_u$ to the same center x_0. Instead, the original fast Gauss transform groups X_u into boxes using a uniform grid and shifts $x \in X_u$ to the center of the box that x belongs to. Nevertheless, such a uniform division scheme is independent of divided data and not appropriate to the distance sample X_u, since the sample is often unevenly distributed, especially in location recommendations in which the distance distributions of users are unique, as shown in Fig. 3. In this paper, to adaptively divide the space based on the sample of distances X_u, we group X_u through clustering to find a set of cluster centers. Specifically, we apply the farthest-point clustering algorithm [21], [39] because of its efficiency.

The primitive farthest-point clustering algorithm discovers a predefined number of clusters. We utilize this method to find an adaptive number of clusters as follows: (1) The algorithm initially selects an arbitrary point $x_0 \in X_u$ as the center of the first cluster and adds it to the cluster center set C. (2) In the i-th iteration, (a) for each point $x_j \in (X_u - C)$, the algorithm computes its nearest distance to the set C:

$$dist_{min}(x_j, C) = \min_{x' \in C} |x_j - x'|. \tag{12}$$

(b) The algorithm next determines the maximum-minimum distance, such that

$$dist_{max-min}(x_i, C) = \max_{x_j} dist_{min}(x_j, C). \tag{13}$$

(c) If

$$dist_{max-min}(x_i, C) > \sigma/2, \tag{14}$$

the algorithm creates a new cluster for the farthest-point x_i, adds x_i into C and continues the iteration process; otherwise, it terminates. It is important to note that the adaptive threshold $\sigma/2$ is carefully defined in this work to ensure a low upper bound of error $\varepsilon(c)$, as shown in Theorem 2 in Section 4.3.

After discovering the set of cluster centers C, each $x \in X_u$ is assigned to its nearest cluster center $x' \in C$. That is, each cluster B (box or group) is implicitly determined by a center $x' \in C$:

$$B = \{x \in X_u |\ |x - x'| \le |x - x''| \text{ for } \forall x'' \in C\}. \tag{15}$$

Hereafter, we denote μ_B as the cluster center corresponding to box B for the sake of presentation.

4.2 Efficient Approximation Algorithm

This section presents an efficient approximation algorithm of $p(l|h_u, L_u)$. Firstly, based on the obtained box B and its center μ_B through clustering, we have

$$\begin{aligned}
\sum_{x \in X_u} e^{-\frac{(y_i-x)^2}{2\sigma^2}} &\approx \sum_B \sum_{x \in B} \sum_{q=0}^{c-1} \frac{1}{q!}\left(\frac{x - \mu_B}{\sqrt{2}\sigma}\right)^q h_q\left(\frac{y_i - \mu_B}{\sqrt{2}\sigma}\right) \\
&= \sum_B \sum_{q=0}^{c-1} \frac{1}{q!} \sum_{x \in B}\left(\frac{x - \mu_B}{\sqrt{2}\sigma}\right)^q h_q\left(\frac{y_i - \mu_B}{\sqrt{2}\sigma}\right) \\
&= \sum_B \sum_{q=0}^{c-1} A_q(B) h_q\left(\frac{y_i - \mu_B}{\sqrt{2}\sigma}\right)
\end{aligned} \tag{16}$$

together with

$$A_q(B) = \frac{1}{q!} \sum_{x \in B}\left(\frac{x - \mu_B}{\sqrt{2}\sigma}\right)^q. \tag{17}$$

Algorithm 2 iGeoRec: The efficient approximation of $p(l|h_u, L_u)$

```
Input: Residence location $h_u$, a set of visited locations $L_u = \{l_1, l_2, \ldots, l_n\}$ and a constant $c$.
Output: $p(l|h_u, L_u)$ for each location $l \in L - L_u$.
 1: // The initialization step
 2: Collect the sample $X_u$ using Equation (1)
 3: Compute the bandwidth $\sigma$ using Equation (4)
 4: Group the sample $X_u$ into boxes $B$ with the center $\mu_B$ based
    on the farthest-point clustering algorithm in Section 4.1
 5: // The pre-computation step of common items $A_q(B)$ :
    $A(q, x)$ is a two-dimension array with two indices $q$ and $x$ to
    store $((x - \mu_B)/\sqrt{2}\sigma)^q/q!$ for $A_q(B)$ in Equation (17)
 6: $A_q(B) \leftarrow 0$
 7: for each $x \in X_u$ do
 8:     Find box $B$ with $\mu_B$ that $x$ belongs to
 9:     $a \leftarrow (x - \mu_B)/\sqrt{2}\sigma$
10:     for $q$ = 0 to $c - 1$ do
11:         if $q = 0$ then
12:             $A(0, x) \leftarrow 1$
13:         else
14:             $A(q, x) \leftarrow A(q - 1, x)a/q$
15:         end if
16:         $A_q(B) \leftarrow A_q(B) + A(q, x)$
17:     end for
18: end for
19: // The approximation computation of $p(l|h_u, L_u)$
20: for each unvisited location $l \in L - L_u$ do
21:     $z \leftarrow 1$ // Initializing auxiliary variable $z$
22:     for each $l_i \in h_u \cup L_u$ do
23:         $y_i \leftarrow distance(l_i, l)$
24:         $v \leftarrow 0$ // Initializing auxiliary variable $v$
25:         for each $B$ such that $|y_i - \mu_B| < 3\sigma$ do
26:             $b \leftarrow (y_i - \mu_B)/\sqrt{2}\sigma$
27:             for $q$ = 0 to $c - 1$ do
28:                 $v \leftarrow v + A_q(B)h_q(b)$
29:             end for
30:         end for
31:         $z \leftarrow z\left[1 - v/(|X_u|\sqrt{2\pi})\right]$
32:     end for
33:     $p(l|h_u, L_u) \leftarrow 1 - z$
34: end for
```

Further, in a Gaussian $e^{-(y_i - x)^2/2\sigma^2}$, the three-sigma rule [22] states that nearly all values of x lie within three standard deviations of y_i. Hence, it is desirable to cut off the sum over all boxes B in Equation (16) by only including the nearest boxes within three standard deviations away from y_i, given by

$$\sum_{x \in X_u} e^{-\frac{(y_i - x)^2}{2\sigma^2}} \approx \sum_{B:|y_i - \mu_B| < 3\sigma} \sum_{q=0}^{c-1} A_q(B) h_q\left(\frac{y_i - \mu_B}{\sqrt{2}\sigma}\right). \tag{18}$$

Finally, in terms of Equations (9) and (18) the approximate $p(l|h_u, L_u)$ is given by

$$p(l|h_u, L_u) \approx 1 - \prod_{i=1}^{n+1}\left(1 - \frac{1}{|X_u|\sqrt{2\pi}} \sum_{B:|y_i - \mu_B| < 3\sigma} \sum_{q=0}^{c-1} A_q(B) h_q\left(\frac{y_i - \mu_B}{\sqrt{2}\sigma}\right)\right). \tag{19}$$

Computational complexity. Algorithm 2 summarizes the overall process for approximating $p(l|h_u, L_u)$ through Equation (19). (1) Algorithm 2 first computes the sample X_u and clusters it into boxes (Lines 2 to 4), which needs $O(n^2)$ work. (2) The key idea of Algorithm 2 is to pre-compute the common items $A_q(B)$ used for all y_i and l (Lines 7 to 18) rather than calculating each combination of x and y_i for each l separately, as done in Algorithm 1. The pre-computation step requires $O(c|X_u|) = O(n^2)$ work. (3) The approximation computation step of Algorithm 2 is similar to Algorithm 1, the only one difference is that Algorithm 2 computes $p(l|h_u, L_u)$ using $A_q(B)$ instead of the Gaussians (Lines 20 to 34). In particular, for any evaluated point y_i, there exist 12 boxes at most that meet $|y_i - \mu_B| < 3\sigma$, as shown in Theorem 1 in Section 4.3. Thus, the approximation computation step needs $O(12c|L - L_u|n) = O(|L|n)$ work, in which c is a small constant since the upper bound of the error ε exponentially decreases as c increases, as shown in Theorem 2. (4) Therefore, the computational complexity of Algorithm 2 is $O(n^2) + O(n^2) + O(|L|n) = O(|L|n)$ (where $|L| \gg |L_u| = n$) that significantly reduces the complexity of Algorithm 1 (i.e., $O(|L|n^3)$).

4.3 Theoretic Analysis

Here, we show two nice properties that enable the efficiency and accuracy of Algorithm 2.

Theorem 1. The maximum number of boxes needed to be considered. In Algorithm 2, for any evaluated point y_i, there exist 12 boxes at most that satisfy $|y_i - \mu_B| < 3\sigma$.

Proof: Give an evaluated point y_i, assume there are more than 12 boxes, say 13 boxes, that meet $|y_i - \mu_B| < 3\sigma$. Let $\mu_{B_1}, \mu_{B_2}, \ldots, \mu_{B_{13}}$ be the centers of the 13 boxes. Note that these centers are discovered through the farthest-point clustering algorithm in Section 4.1. In terms of Inequation (14), only when the distance of a new farthest-point with the maximum-minimum distance (some μ_B) to all the found cluster centers is larger than $\sigma/2$, the farthest-point has the chance to be added into the set of cluster centers. Thus, we have: for $\forall i \neq j$,

$$|\mu_{B_i} - \mu_{B_j}| > \sigma/2.$$

Without loss of generality, assume

$$\mu_{B_1} < \mu_{B_2} < \cdots < \mu_{B_{13}}.$$

Whence,

$$\mu_{B_{13}} - \mu_{B_1} = (\mu_{B_{13}} - \mu_{B_{12}}) + \cdots + (\mu_{B_2} - \mu_{B_1}) > 6\sigma.$$

On the other hand, since μ_{B_1} and $\mu_{B_{13}}$ meet $|y_i - \mu_{B_1}| < 3\sigma$ and $|y_i - \mu_{B_{13}}| < 3\sigma$:

$$\begin{aligned} \mu_{B_{13}} - \mu_{B_1} &= |\mu_{B_{13}} - y_i + y_i - \mu_{B_1}| \\ &\leq |\mu_{B_{13}} - y_i| + |y_i - \mu_{B_1}| < 6\sigma, \end{aligned}$$

which contradicts $\mu_{B_{13}} - \mu_{B_1} > 6\sigma$. Therefore, Theorem 1 holds. □

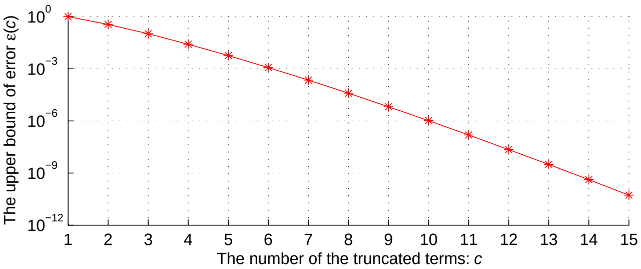


Fig. 4. The upper bound of the error $\varepsilon(c)$

Theorem 2. Relationship between constant c and error ε. By truncating the infinite series after c terms in Equation (10), Algorithm 2 guarantees that the error ε satisfies $|\varepsilon(c)| < 2^{1-c}/\sqrt{c!}$.

Proof: At first, in terms of Cramér's inequality [40] $|h_q(x)| \leq 2^{q/2}(q!)^{1/2}e^{-x^2/2}$ and $e^{-x^2/2} \leq 1$, we have

$$|h_q(x)| \leq 2^{q/2}(q!)^{1/2}.$$

Hence, in Equation (10),

$$\begin{aligned}
|\varepsilon(c)| &\leq \sum_{q\geq c} \frac{1}{q!}\left|\frac{x-x_0}{\sqrt{2}\sigma}\right|^q \left|h_q\left(\frac{y_i-x_0}{\sqrt{2}\sigma}\right)\right| \\
&\leq \sum_{q\geq c} \frac{1}{q!}\left|\frac{x-x_0}{\sqrt{2}\sigma}\right|^q 2^{q/2}(q!)^{1/2} \\
&= \sum_{q\geq c} \frac{1}{\sqrt{q!}}\left|\frac{x-x_0}{\sigma}\right|^q < \frac{1}{\sqrt{c!}}\sum_{q\geq c}\left|\frac{x-x_0}{\sigma}\right|^q.
\end{aligned}$$

In Algorithm 2, each $x \in X_u$ is shifted to its box center (i.e., μ_B or x_0) and the distance between x and its center must be not larger than $\sigma/2$; otherwise, x itself will be selected as a center based on Inequation (14). Thus,

$$\left|\frac{x-x_0}{\sigma}\right| = \left|\frac{x-\mu_B}{\sigma}\right| \leq \frac{1}{2}.$$

Accordingly,

$$|\varepsilon(c)| < \frac{1}{\sqrt{c!}}\sum_{q\geq c} 2^{-q} = \frac{2^{1-c}}{\sqrt{c!}}.$$

□

According to Theorem 2, the upper bound of the error decreases even faster than the exponential decay as the number of the truncated terms increases, as shown in Fig. 4.

5 Experimental Evaluation

In this section, we describe our experiment settings for evaluating the performance of iGeoRec against the state-of-the-art location recommendation techniques.

5.1 Data Sets

We use two publicly available large-scale real check-in data sets[1] that were crawled from Foursquare [5] and Gowalla [18]. The statistics of the data sets are shown in TABLE 2.

TABLE 2
Statistical information of the two data sets

	Foursquare	Gowalla
Number of users	11,326	196,591
Number of locations	182,968	1,280,969
Number of check-ins	1,385,223	6,442,890
Number of social links	47,164	950,327
User-location matrix density	2.3×10^{-4}	2.9×10^{-5}
Avg. No. of visited locations per user	42.44	37.18
Avg. No. of check-ins per location	2.63	3.11

5.2 Evaluated Recommendation Techniques

The recommendation techniques implemented in our experiments are listed below.

- **Nonnegative Matrix Factorization** (NMF): In the conventional collaborative filtering techniques [11], [33], [34], matrix factorization models are superior to classic nearest-neighbor models for producing personalized recommendations [41], [42]. In particular, in our experiments, we use nonnegative matrix factorization as a baseline since it is comparable to or better than other matrix factorization techniques like singular value decomposition on effectiveness through respecting the nonnegativity [43], [44].
- **Multi-center Gaussian Model** (MGM) [3]: MGM uses the geographical influence by modeling the distance between visited locations and centers (i.e., the most popular locations) as a *universal* multi-center Gaussian distribution for all users.
- **Power-law Distribution** (PD) [8], [13], [14], [16]: PD uses the geographical influence by modeling the distance between every pair of locations visited by the same user as a *universal* power-law distribution for all users.
- **Algorithm 2** (iGeoRec): iGeoRec uses the geographical influence by modeling the distance between every pair of locations visited by the same user as a *personalized* nonparametric distribution for each user.
- **Algorithm 1** (Exact): The only one difference in Exact from iGeoRec is that it computes the exact probability of a user visiting a new location.

Note that: We also conduct experiments to investigate the performance of NMF, MGM, PD and iGeoRec when integrating with the widely used social influence (i.e., the social links between users established in LBSNs as depicted in TABLE 2) [3], [8], [23] in Section 6.2.

5.3 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,

[1] The check-in data sets used for our experiments can be downloaded from http://www.public.asu.edu/~hgao16/Publications.html and http://snap.stanford.edu/data/loc-gowalla.html.

it is important to find out how many locations actually visited by the target user in the testing data set are discovered by the recommendation technique. For this purpose, we employ two standard metrics: **precision** and **recall** [3], [8]:

- Precision defines the ratio of the number of discovered locations to the k recommended locations, i.e.,

$$\text{precision} = \frac{\text{the number of discovered locations}}{k}.$$

- Recall defines the ratio of the number of discovered locations to the number of **positive locations**, which have been visited by the target user in the testing set, i.e.,

$$\text{recall} = \frac{\text{the number of discovered locations}}{\text{the number of positive locations}}.$$

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

Recommendation efficiency. We compare the running time of iGeoRec 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.

5.4 Parameter Settings

Fixed testing set. 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. The half of check-in data with later timestamps is used as the fixed testing set for performance comparison of different recommendation methods or different parameter values of the same method.

Training set. By default the other half of check-in data with earlier timestamps is used as the training set, unless in Section 6.1.1 we use different percentages $x\%$ ($x = 10, 20, \ldots, 100$) of the half check-in data with the earliest timestamps as the training set to explore the effect of the size of training data.

Number of recommended locations. By default, the number of recommended locations k is set to 25, unless in Section 6.1.2 we vary k from 2 to 50 to investigate the effect of k.

Number of truncated terms. By default, in iGeoRec the number of truncated terms is set to a small value $c = 8$ based on Theorem 2, unless in Section 6.3 we study the approximation error with respect to the change of c from 1 to 15.

Note that: The numbers of *positive locations* and *check-in locations* of a user are not tunable but user-specific, in which the positive locations are the locations visited by the user in the testing data set while the check-in locations are the locations visited by the user in the training data set. Unless otherwise specified, the performance of evaluated recommendation techniques is averaged on all users with various numbers of *positive locations* and *check-in locations*.

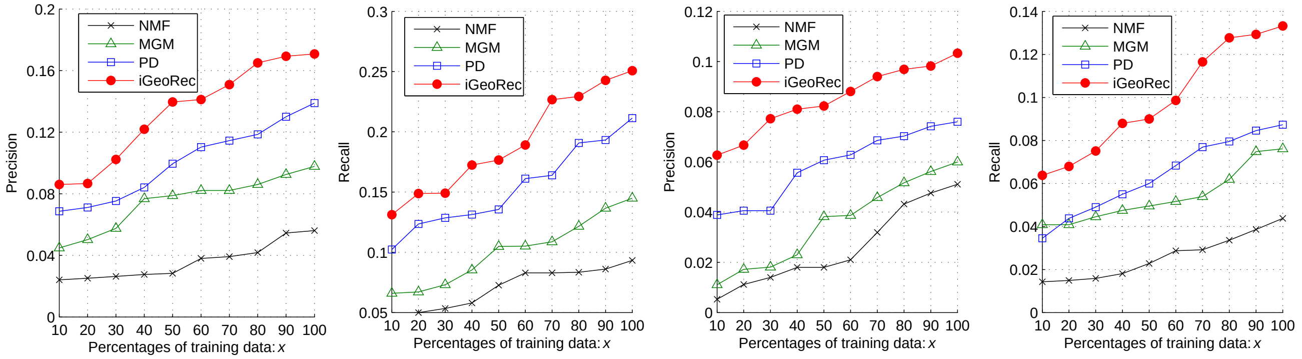
6 Experimental Results

This section analyzes our extensive experimental results. We first compare our iGeoRec against the state-of-the-art geographical recommendation techniques including MGM [3] and PD [8], [13], [14], [16] in terms of the *recommendation accuracy* (Sections 6.1 and 6.2). We then study the *approximation error* of iGeoRec in comparison to Exact, i.e., the exact method depicted in Algorithm 1 (Section 6.3). Finally, we evaluate the *recommendation efficiency* of iGeoRec (Section 6.4).

6.1 Recommendation Accuracy

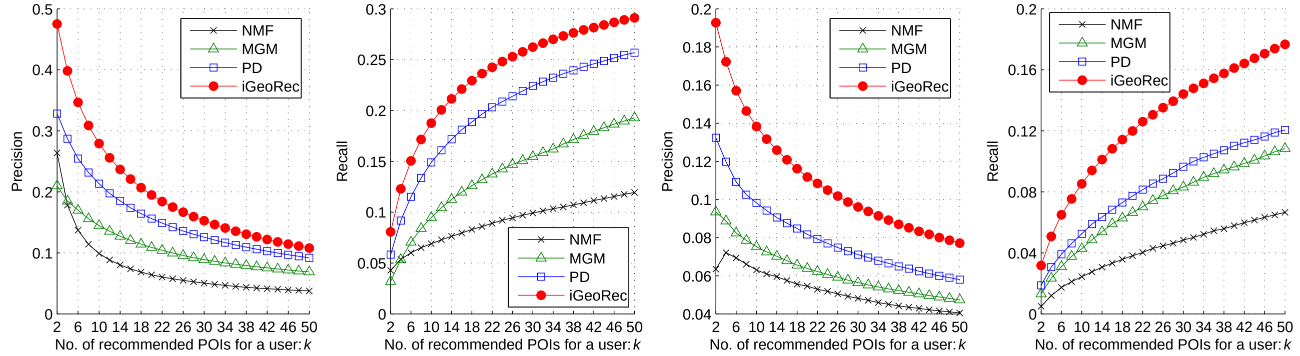
Here we compare the recommendation accuracy of iGeoRec, PD, MGM and NMF with the effect of percentages of training data (Fig. 5), numbers of recommended locations (Fig. 6), numbers of positive locations (Fig. 7), and numbers of check-ins of users (Fig. 8). At first, we conclude **the most important and general findings** in all experiments on two large-scale real data sets collected from Foursquare and Gowalla as follows.

1) NMF: As one of the most powerful collaborative filtering techniques, NMF is still inferior to iGeoRec, PD and MGM, since it ignores the geographical influence of users and locations on users' check-in behavior.
2) MGM [3]: By using the geographical influence, MGM improves the accuracy of location recommendations in comparison to NMF. However, the improvement is considerably limited, because it models the geographical influence as a universal distribution for all users and considers the distance between a location and a center instead of between every pair of locations visited by the same user. As a result, the distribution $p(l|h_u, L_u)$ obtained by MGM is actually independent of u's set of visited locations L_u.
3) PD [8], [13], [14], [16]: By modeling the distance between every pair of locations visited by the same user as a power-law distribution for all users, PD further enhances the performance of recommending locations, but it still inherits the limitation of the universal distance distribution for all users.
4) iGeoRec: By personalizing the geographical influence through modeling an individual distance distribution for each user, our iGeoRec



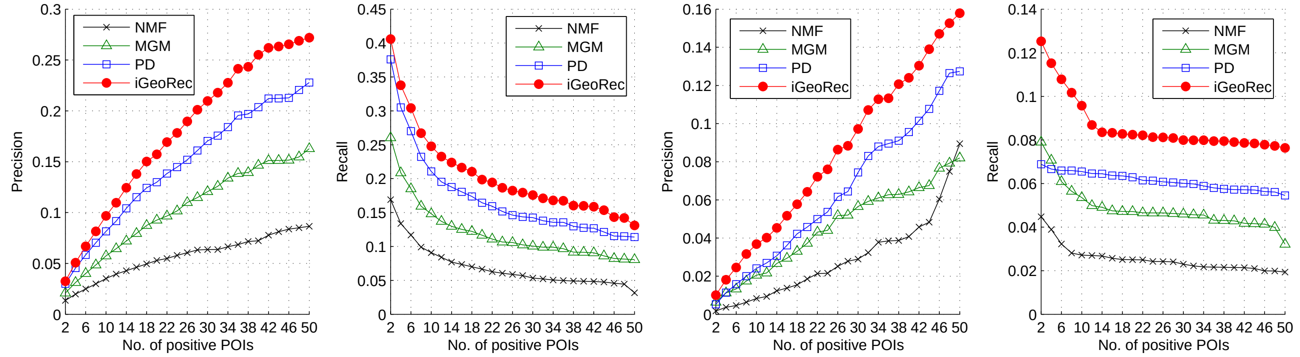
(a) Foursquare (b) Foursquare (c) Gowalla (d) Gowalla

Fig. 5. Effect of percentages of training data on recommendation accuracy



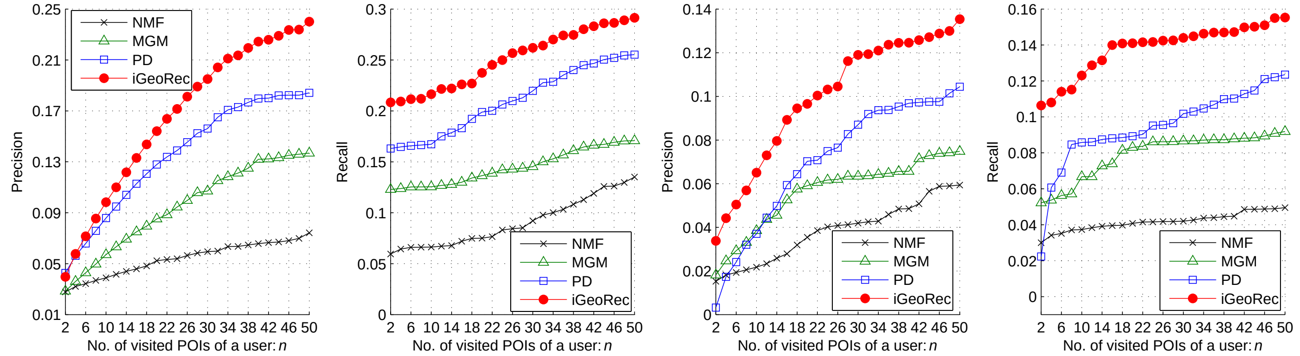
(a) Foursquare (b) Foursquare (c) Gowalla (d) Gowalla

Fig. 6. Effect of numbers of recommended locations for users on recommendation accuracy



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

Fig. 7. Effect of numbers of positive locations on recommendation accuracy



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

Fig. 8. Effect of numbers of check-in locations of users on recommendation accuracy

always exhibits the best recommendation quality in terms of precision and recall. These results verify the superiority of exploiting the personalized geographical influence for location recommendations proposed in this paper over the universalized geographical influence adopted by MGM and PD.

5) Foursquare vs. Gowalla: The performance of all evaluated methods in the Foursquare data set is always higher than that in the Gowalla 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. Promisingly, iGeoRec achieves the best recommendation accuracy in both the Foursquare and Gowalla data sets.

6.1.1 *Effect of Percentages of Training Data*

Fig. 5 depicts the recommendation accuracy of NMF, MGM, PD and iGeoRec with respect to varying the percentages of training data in the earlier half check-in data. As expected, the precision and recall steadily increase as the percentage of the training data rises. The reason is that, with the raise of the percentage of the training data, the training data set becomes denser, which is helpful for recommendation techniques to learn users' preferences on locations or POIs.

6.1.2 *Effect of Numbers of Recommended Locations*

Fig. 6 depicts the recommendation accuracy of the techniques with respect to varying the numbers of recommended locations, i.e., k from 2 to 50. With the increase of k, the recall gradually gets higher but the precision steadily becomes lower on the two data sets. The reason is that, in general, by returning more locations for users, it is able to discover more locations that users would like to check in. However, since the recommendation techniques return the locations with the top-k highest scores, e.g., rating for NMF or visiting probability for MGM, PD and iGeoRec, some recommended locations are less possible to be liked by users due to their lower visiting probabilities.

6.1.3 *Effect of Numbers of Positive Locations*

Fig. 7 depicts the recommendation accuracy with respect to the change of the numbers of **positive locations** from 2 to 50 on the two data sets. For example, a measure at "No. of positive POIs = 2" is averaged on all users who have checked in two locations in the testing data set. As the number of the positive locations of users gets larger, the precision generally increases but the recall usually decreases. Our explanation is that the raise of the number of the positive locations means that the recommendation techniques are more capable of discovering locations that users would like to visit, but it is hard to discover all of this kind of locations.

6.1.4 *Effect of Numbers of Check-in POIs of Users*

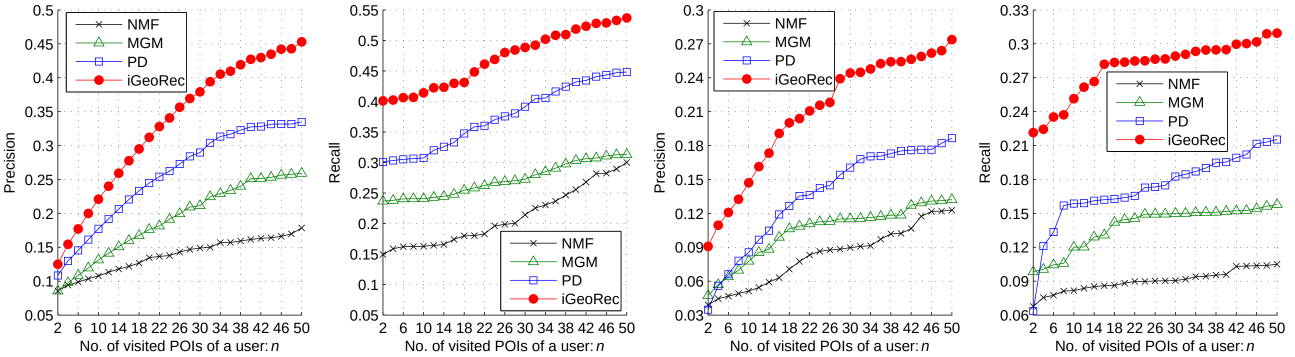
Fig. 8 depicts the recommendation accuracy with respect to the change of the numbers of check-in locations of users, i.e., n from 2 to 50, on the two data sets. For instance, a measure at "$n = 2$" is averaged on all users who have checked in two locations in the training data set. As users check in more locations, our iGeoRec can more accurately estimate the probability density and predict the visiting probability of new locations for these users through using more check-in data. As a result, their precision and recall incline accordingly.

6.1.5 *Discussion on Data Sparsity*

The accuracy of all recommendation techniques for LBSNs is usually not high, because they suffer from the data sparsity problem, i.e., the density of user-location check-in matrix is pretty low. For example, the reported maximum precision is 0.03 over the two data sets with 9.85×10^{-4} and 6.35×10^{-3} densities in [16]. Even worse, the two data sets used in our experiments have a lower density, 2.3×10^{-4} in the Foursquare data set and 2.9×10^{-5} in the Gowalla data set (TABLE 2), so the relatively low precision and recall values are common and reasonable in the experiments. Thus, we focus on the relative accuracy of iGeoRec compared to the state-of-the-art geographical recommendation techniques including MGM and PD, which we expect that iGeoRec can improve recommendation accuracy as more check-in activities are logged, for example, as shown in Figs. 5 and 8.

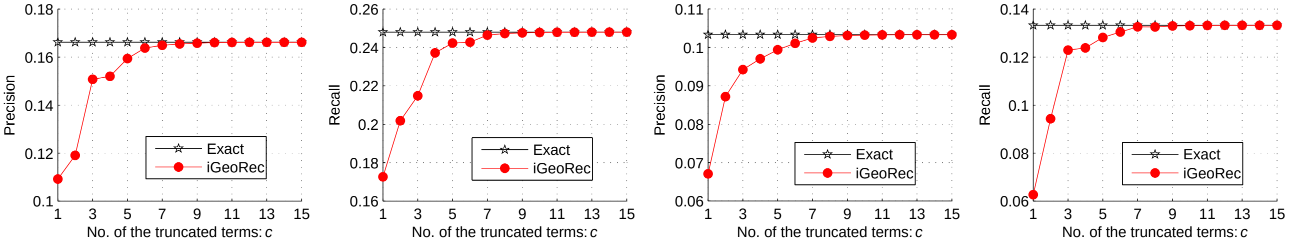
Following, we show how well iGeoRec deals with the data sparsity problem in three-fold. (1) By applying the Gowalla data set with one-order-of-magnitude sparser than Foursquare, compared to PD, MGM and NMF, iGeoRec accomplishes better improvement on recommendation accuracy in Gowalla than Foursquare according to Figs. 5, 6, 7 and 8. (2) By using the less check-in data as the training set depicted in Fig. 5, iGeoRec always shows much better precision and recall than the second best result given by PD when confronting more severe problems of data sparsity. (3) By observing the recommendation accuracy for cold-start users who have only few check-in POIs, for example, $n \leq 10$ in Fig. 8, iGeoRec generally maintains a remarkably higher level of precision and recall for the cold-start users in comparison to PD, MGM and NMF.

The reason why iGeoRec outperforms PD, MGM and NMF for the data sparsity problem is that: iGeoRec utilizes the personalized geographical influence to learn users' profiles for location recommendations. That is, if a user likes to travel around, iGeoRec estimates the locations far away with higher probabilities and recommends this kind of locations for the user. In contrast, if a user tends to stay in a single region, iGeoRec estimates the nearby locations with higher probabilities and recommends them to the



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

Fig. 9. Recommendation accuracy enhancement through social influence



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

Fig. 10. Approximation error of iGeoRec in comparison to Exact, i.e., the exact method (Algorithm 1)

user. Therefore, iGeoRec is actually a kind of contend-based recommendation models that are less sensitive to the data sparsity and are usually employed to relieve the data sparsity problem. Further, we can mitigate the data sparsity problem by integrating the geographical influence with the social influence which is discussed in Section 6.2.

6.2 Recommendation Accuracy Enhancement

To further enhance the recommendation accuracy, we integrate iGeoRec with the social influence (i.e., the social links between users established in LBSNs as depicted in TABLE 2) based on the fact that friends are more likely to share common interests. We employ the social location recommendation technique [23] to estimate the rating of a user visiting a location. In general, the integration has three main steps.

(1) The social links between users and distances between their residences are used to derive their similarities. Formally, let $F(u)$ be a set of users having social links with u and $distance(h_u, h_{u'})$ be the distance between the residences h_u and $h_{u'}$. If $u' \in F(u)$, the similarity between u and u' is calculated by $sim(u,u') = 1 - \frac{distance(h_u,h_{u'})}{\max\limits_{u'' \in F(u)} distance(h_u,h_{u''})}$. Otherwise, $sim(u,u') = 0$.

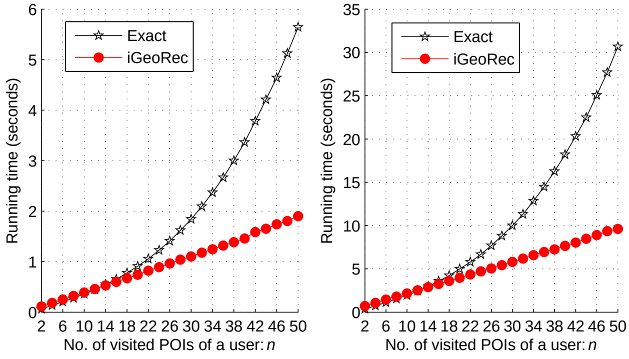
(2) The rating $\hat{r}_{u,l}$ of user u to new location l is estimated based on the social collaborative filtering method by $\hat{r}_{u,l} = \frac{\sum_{u' \in F(u)} sim(u,u') \cdot r_{u',l}}{\sum_{u' \in F(u)} sim(u,u')}$, where $r_{u',l}$ is the actual frequency of u' visiting l.

(3) The estimated rating $\hat{r}_{u,l}$ and visited probability $p(l|h_u, L_u)$ in Equation (19) are fused into a unified score $s_{u,l}$ by the product rule: $s_{u,l} = \hat{r}_{u,l} \cdot p(l|h_u, L_u)$. Finally, iGeoRec recommends the top-k locations with the highest score $s_{u,l}$ for user u.

To make fair comparison, NMF, MGM and PD are also integrated with the social influence in the same way. Due to similar result and space limitation, Fig. 9 only shows the recommendation accuracy enhancement regarding the various numbers of check-in locations of users. The precision and recall of all methods increase significantly and nearly achieve the double accuracy in comparison to Fig. 8. In particular, the recommendation accuracy of the cold-start users with few visited POIs ($n \leq 10$) records the highest growth rates. The reason is that these methods are able to infer users' preferences on locations more accurately by using the social influence. More importantly, the proposed iGeoRec still performs the best, since iGeoRec exploits the geographical influence more sophisticatedly and comprehensively, i.e., using the personalized distance distribution rather than the universal distribution.

6.3 Approximation Error

Fig. 10 shows the approximation error of iGeoRec with respect to varying the numbers of truncated terms in Equation (10), i.e., c from 1 to 15. Here we compare its recommendation accuracy with that of Exact, i.e., the exact method (Algorithm 1), instead of measuring the error itself, since the error itself is not meaningful for location recommendations and the effect of approximation error on the recommendation accuracy is more significant.



(a) Foursquare (b) Gowalla

Fig. 11. Recommendation efficiency of iGeoRec compared to Exact, i.e., the exact method (Algorithm 1)

In Fig. 10, with the increase of the number of truncated terms, the precision and recall of iGeoRec 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 as c ascends based on Theorem 2. In previous experiments, it is reasonable to set a relatively small default value: $c = 8$, since the resultant approximation error is negligible according to the experimental results depicted in Fig. 10.

6.4 Recommendation Efficiency

In this section, we focus on comparing the recommendation efficiency of iGeoRec with Exact, i.e., the exact method depicted in Algorithm 1. Note that iGeoRec and PD have the same computational complexity of $O(|L|n)$, and the complexity of MGM does not rely on the number of locations visited by users, but it depends on the number of centers (the most popular locations).

Fig. 11 gives the running time of iGeoRec regarding the change of the number of check-in locations of users, i.e., n from 2 to 50. It is important to note that the running time is averaged on different users with the same number of check-in locations in order to obtain the smooth experimental results. For the small value of n, e.g., less than 10, iGeoRec needs more time than Exact, since the actual computational requirement of iGeoRec is $12c|L|n$ that is larger than $|L|n^3$ when $n < 10$ and $c = 8$ (default). Conversely, as the number of visited locations of users gets larger, the running time of iGeoRec linearly increases, but that of Exact dramatically increases. Subsequently, Exact costs much more time than iGeoRec for larger numbers of check-in locations. Therefore, iGeoRec is more scalable to the web-scale calculation in the process of location recommendations. In addition, both iGeoRec and Exact take more time to recommend locations in the Gowalla data set than the Foursquare data set, because the former contains more location candidates for recommendations than the latter, as shown in TABLE 2. In practice, we can prune the location candidates in the recommendation process through only considering the candidates nearby the visited locations of users, which can be implemented by the inverted-indexing technique in information retrieval and is out of the scope of this paper.

7 Conclusion and Future Work

In this paper, our objective is to overcome the limitation that the current graphical recommendation techniques merely model the geographical influence as a *universal* distance distribution for all users. We have proposed the *personalized* and *efficient* geographical location recommendation framework called iGeoRec. In iGeoRec, we have overcome two main challenges: (1) iGeoRec personalizes the geographical influence and computes the probability of users visiting new locations accurately. Specifically, we have developed the probabilistic approach to personalize the geographical influence as an individual distribution for each user and predict the probability of a user visiting a new location using the personal distribution. (2) We have developed the efficient approximation method to compute the probability of each user visiting all new locations; the efficient approximation method reduces the computational complexity of the exact computation method from $O(|L|n^3)$ to $O(|L|n)$ (where $|L|$ is the total number of locations and n is the number of check-in locations of a user). Finally, we have conducted extensive experiments to evaluate the recommendation accuracy, recommendation efficiency, and approximation errors of iGeoRec using the two data sets crawled from Foursquare and Gowalla. Experimental results show that iGeoRec provides significantly better performance than all other evaluated recommendation techniques.

In the future, we plan to study three directions of location recommendations to extend iGeoRec: (a) how to recommend a trip of a series of locations, (b) how to incorporate the category information of locations into our personalized geographical location recommendation framework, and (c) how to take temporal influence into account to capture the change of users' preferences.

References

[1] J. Bao, Y. Zheng, D. Wilkie, and M. F. Mokbel, "A survey on recommendations in location-based social networks," *ACM TIST, accepted to appear*, 2013.

[2] J. Bao, Y. Zheng, and M. F. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," in *ACM SIGSPATIAL*, 2012.

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

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

[5] H. Gao, J. Tang, and H. Liu, "gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks," in *ACM CIKM*, 2012.

[6] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "LARS: A location-aware recommender system," in *IEEE ICDE*, 2012.

[7] M. Ye, P. Yin, and W.-C. Lee, "Location recommendation for location-based social networks," in *ACM SIGSPATIAL*, 2010.

[8] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *ACM SIGIR*, 2011.

[9] J. J.-C. Ying, E. H.-C. Lu, W.-N. Kuo, and V. S. Tseng, "Urban point-of-interest recommendation by mining user check-in behaviors," in *ACM UrbComp*, 2012.

[10] G. Adomavicius and A. Tuzhilin, "Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE TKDE*, vol. 17, no. 6, pp. 734–749, 2005.

[11] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM TWEB*, vol. 5, no. 1, pp. 2:1–24:33, 2011.

[12] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[13] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," in *ACM KDD*, 2013.

[14] X. Liu, Y. Liu, K. Aberer, and C. Miao, "Personalized point-of-interest recommendation by mining users' preference transition," in *ACM CIKM*, 2013.

[15] C. Wang, M. Ye, and W.-C. Lee, "From face-to-face gathering to social structure," in *ACM CIKM*, 2012.

[16] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann, "Time-aware point-of-interest recommendation," in *ACM SIGIR*, 2013.

[17] J. Riedl, "Personalization and privacy," *IEEE Internet Computing*, vol. 5, no. 6, pp. 29–31, 2001.

[18] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *ACM KDD*, 2011.

[19] B. W. Silverman, *Density estimation for statistics and data analysis*. Chapman and Hall, 1986.

[20] L. Greengard and J. Strain, "The fast Gauss transform," *SIAM Journal on Scientific Computing*, vol. 12, no. 1, pp. 79–94, 1991.

[21] T. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.

[22] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[23] J.-D. Zhang and C.-Y. Chow, "iGSLR: Personalized geo-social location recommendation - A kernel density estimation approach," in *ACM SIGSPATIAL*, 2013.

[24] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "A random walk around the city: New venue recommendation in location-based social networks," in *IEEE SocialCom*, 2012.

[25] D. Zhou, B. Wang, S. M. Rahimi, and X. Wang, "A study of recommending locations on location-based social network by collaborative filtering," in *Canadian AI*, 2012.

[26] C.-C. Hung, W.-C. Peng, and W.-C. Lee, "Clustering and aggregating clues of trajectories for mining trajectory patterns and routes," *VLDB Journal, accepted to appear*, 2011.

[27] K. W.-T. Leung, D. L. Lee, and W.-C. Lee, "CLR: A collaborative location recommendation framework based on coclustering," in *ACM SIGIR*, 2011.

[28] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, "Collaborative filtering meets mobile recommendation: A user-centered approach," in *AAAI*, 2010.

[29] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Collaborative location and activity recommendations with GPS history data," in *WWW*, 2010.

[30] ——, "Towards mobile intelligence: Learning from gps history data for collaborative recommendation," *Artificial Intelligence*, vol. 184-185, pp. 17–37, 2012.

[31] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma, "Recommending friends and locations based on individual location history," *ACM TWEB*, vol. 5, no. 1, pp. 5:1–5:44, 2011.

[32] B. Hu and M. Ester, "Spatial topic modeling in online social media for location recommendation," in *ACM RecSys*, 2013.

[33] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized qos-aware web service recommendation and visualization," *IEEE TSC*, vol. 6, no. 1, pp. 35–47, 2013.

[34] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, "Personalized web service recommendation via normal recovery collaborative filtering," *IEEE TSC*, vol. 6, no. 4, pp. 573–579, 2013.

[35] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura, "Geo topic model: Joint modeling of user's activity area and interests for location recommendation," in *ACM WSDM*, 2013.

[36] N. Joshi, T. Kadir, and M. Brady, "Simplified computation for nonparametric windows method of probability density function estimation," *IEEE TPAMI*, vol. 33, no. 8, pp. 1673–1680, 2011.

[37] A. Elgammal, R. Duraiswami, and L. S. Davis, "Efficient kernel density estimation using the fast Gauss transform with applications to color modeling and tracking," *IEEE TPAMI*, vol. 25, no. 11, pp. 1499–1504, 2003.

[38] B. Jian and B. C. Vemuri, "Robust point set pegistration using gaussian mixture models," *IEEE TPAMI*, vol. 33, no. 8, pp. 1633–1645, 2011.

[39] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis, "Improved fast gauss transform and efficient kernel density estimation," in *IEEE ICCV*, 2003.

[40] J. Indritz, "An inequality for hermite polynomials," *Proceedings of the AMS*, vol. 12, no. 6, pp. 981–983, 1961.

[41] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[42] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative web service qos prediction via neighborhood integrated matrix factorization," *IEEE TSC*, vol. 6, no. 3, pp. 289–299, 2013.

[43] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[44] C. Liu, H.-C. Yang, J. Fan, L.-W. He, and Y.-M. Wang, "Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce," in *WWW*, 2010.

Jia-Dong Zhang received the M.Sc. degree in computer science and engineering from Yunnan University, China, in 2009. He is currently working toward the Ph.D. degree with the Department of Computer Science, City University of Hong Kong. His research work has been published in journals such as *IEEE TITS*, *Pattern Recognition*, and *Decision Support Systems*. His research interests include machine learning, data mining, and location-based services.

Chi-Yin Chow received the M.S. and Ph.D. degrees from the University of Minnesota-Twin Cities in 2008 and 2010, respectively. He is currently an assistant professor in Department of Computer Science, City University of Hong Kong. His research interests include spatio-temporal data management and analysis, GIS, mobile computing, and location-based services. He is the co-founder and co-organizer of ACM SIGSPATIAL MobiGIS 2012, 2013, and 2014.

Yanhua Li is a researcher with HUAWEI Noah's Ark LAB. He received two Ph.D. degrees in Computer Science from University of Minnesota, Twin Cities in 2013 and Electrical Engineering from Beijing University of Posts and Telecommunications in 2009. His broad research interests are in analyzing, understanding, and making sense of big data generated from various complex networks, including online social behavior modeling and analysis, large-scale complex network sampling, measurement, and mining, spatio-temporal data management and mining. He has interned in Bell Labs in New Jersey, Microsoft Research Asia, and HUAWEI research labs of America.