Privacy-preserving Point-of-Interest Recommendation Based on Geographical and Social Influence

Yongfeng Huo\textsuperscript{a}, Bilian Chen\textsuperscript{a,}\textsuperscript{*}, Jing Tang\textsuperscript{b}, Yifeng Zeng\textsuperscript{b,}\textsuperscript{*}

\textsuperscript{a}Department of Automation, Xiamen University, Xiamen 361005, China. 
\textsuperscript{b}School of Computing, Teesside University, UK.

Abstract

We investigate the privacy-preserving problem for point-of-interest (POI) recommendation system for the rapidly growing location-based social network (LBSN). The LBSN-based recommendation algorithms usually consider three factors: user similarity, social influence between friends and geographical location. The LBSN-based recommendation system first needs to collect relevant information of users and then provide them with potentially interesting contents. However, sensitive information of users may be leaked when the recommendation is provided. In this article, we focus on preventing user’s privacy from disclosure upon geographical location and friend relationship factors. We propose a geographical location privacy-preserving algorithm (GLP) that achieves \((r,h)\)-privacy and present a friend relationship privacy-preserving algorithm (FRP) through adding Laplacian distributed noise for fusing the user trusts. Subsequently, we integrate the GLP and FRP algorithms into a general recommendation system to build the privacy-preserving recommendation system. The novel system enjoys the privacy guarantee under the metric differential entropy through theoretical analysis. Experimental results demonstrate a good trade-off between privacy and accuracy of the proposed recommendation system.

Keywords: POI Recommendation, Privacy Preservation, Fuzzy Location, Differential Privacy

1. Introduction

1.1. Overview

With the rapid development of social network sites (SNSs), web 4.0 and mobile devices, a number of location-based social networking services, such as Facebook, Microblog, Foursquare, Whrrl, etc., have attracted millions of users many of whom even integrate social networks into their daily lives. The LBSNs allow users to establish online links with their friends or other users, and to share tips and experiences of their visits to plentiful point-of-interests (POIs) \cite{?}, e.g., restaurants, stores, cinemas and so on. It is important to enhance the effectiveness of the POI recommendation and recommend next new POIs to users so that
the users could explore new places and know their cities better through LSBNs. By doing this, the system needs to gather lots of important and valuable data, such as the connection between users, the relationship between the POIs and users and so on. The information may also be too sensitive to the user since a potential attacker will effectively exploit the user’s privacy information from the recommended results. Hence, we need to take the user’s privacy information into account while improving the accuracy of the POI recommendation system.

Although it is difficult to define the privacy in a precise way, it is not hard to understand the definition of privacy disclosure. If any user’s part of the privacy information is exposed to an attacker, the privacy disclosure of the user occurs. In general, there are four types of privacy disclosure: disclosure of identifiers, attributes, social relations and contact information [? ]. The focus of this paper is on designing algorithm, theoretical analysis and experimental verification of a recommendation system with a built-in privacy guarantee. We resort to the technique of differential privacy that is a mathematically rigorous definition of privacy and is suited to analysis of large datasets and equipped with a formal measurement of privacy loss [? ? ? ? ? ?]. Moreover, differentially private algorithms will be taken by inputting a privacy parameter, which indicates the permitted privacy loss in any execution of the algorithm and offers a concrete trade-off between privacy and accuracy.

1.2. Motivation

As the POI recommendation system may use a user’s sensitive information to make recommendations, the user may not want to accept such a recommendation system. The incorporation of privacy-preserving methods in a traditional POI recommendation system has been studied in many literatures [? ? ? ? ? ? ? ?]. Most of them hide the user’s personal records from the recommendation system, while providing the POI results as appropriate as possible.

Existing location privacy-preserving techniques exhibit two significant limitations. First, some require a trusted third-party anonymizer that maintains information of all user locations. Such an action may not always be available, and it could cause security/privacy problems by itself. Second, the underlying $k$-anonymity techniques are generally not adequate enough for location privacy, e.g. the privacy-area aware dummy generation algorithms for $(k,s)$-privacy [? ]. They do not consider population densities thereby being inapplicable for all regions. For example, there may exist a large population density in a shopping street and a small one in a flat countryside. A static setting of privacy parameters can not adapt the two cases and even cause a problem of data availability, i.e., to obtain a larger/smaller privacy region. Intuitively, we need to use different parameter values $(k,s)$; however, the existing algorithms do not support to tune a parameter of the population density.

Another challenge is to protect private cyber links information from disclosure in a concise way. There are usually two extensively studied buddy relationship attacking models [? ], which are illustrated in Section 3.3 in detail. Daniele et al. [? ] built a POI-Ti-Dico framework by classifying user roles and cutting space area with different marks. However, the weakness of the POI-Ti-Dico lies in a completely new model of a real-world case including a new division of space, a new classification of user roles and so on, i.e., this framework has a great difference with the existing systems. If applied to practice, it requires a major transformation for a general recommendation system and results in a relatively high cost.
1.3. Contributions

The main contribution of this work is to design and analyze a realistic recommendation system built to provide privacy guarantees. The task is non-trivial because the previous recommendation systems are not designed towards a new way of privacy protection, and the previous privacy research has focused on more modest algorithms without attempting at a practical validation. Our findings are that the privacy does not need to come at a substantial expense in accuracy. For the approaches we considered, a privacy-preserving algorithm may start from a geographical location and friend relationships of people who use the recommendation system. Our new algorithms are as follows.

**Geographical location privacy-preserving algorithm (GLP).** In the POI recommendation system, there is an attack mode of location privacy information for users. As shown in Fig. 1, inspired by \((k, s)\)-privacy, we present GLP to protect the user’s location information from disclosure. GLP is a controllable fuzzy geographical location algorithm to make a user’s position into a virtual circle with a dynamic radius based on the population density, but it does not significantly reduce the recommendation accuracy.

**Friend relationship privacy-preserving algorithm (FRP).** We introduce the attacking methods by using the friend relationship in the recommendation system. Contrary to the POI-Ti-Dico framework, we propose a more lightweight and effective controllable algorithm which adopts Laplacian differential privacy to fuse friend relationship by adding enough noise.

As an additional contribution of this work, we demonstrate the integration of differential privacy technology into practical systems. We adopt a novel evaluation approach of differential entropy. Moreover, we give an important formula and quantification standards, which allow users to control their privacy-preserving levels by choosing suitable private parameters. Experimental results show that our two privacy-preserving algorithms are useful and effective.

1.4. Outline

The rest of the paper is organized as follows. Section 2 discusses related works and Section 3 reviews the POI recommendation system, differential privacy and the private attacking models. Section 4 presents our privacy-preserving algorithms with respect to the factors of geographical location and friend relationship, and subsequently proposes theoretical analysis.
on them, respectively. In Section ??, extensive experiments are elaborated to demonstrate
the utility of the proposed methods. Section ?? concludes the work and discusses the future
works.

2. Related work

Dalenius [?] first proposed private data protection and introduced the purpose of private
database protection explicitly. On the one hand, the attacker can not get any information
in the database if no data is accessed. On the other hand, even if the attacker gets all the
entries except for a particular entry, he can not get any information of this particular one.
Although the definition of privacy was still too vague and he did not provide any accurate
or quantifiable indicators, he provided a general direction for the later study.

Can et al. [?] gathered 21 online social network problems and presented a review of them
with related applications. A crucial problem is the privacy leak of user social relationship.
Sweeney [?] proposed the $k$-anonymity method to solve the problem that even if the explicit
identifier of each entry is deleted, attackers can still infer the entry’s privacy information by
multiple attribute values of the entry with high probability. In a $k$-anonymous database,
for a given Quasi-Identifier (QID), there are at least $k$ records with the same value, so the
probability of deducing a target record by QID is at most $1/k$. However, the assumption of
$k$-anonymity is that each record in the database corresponds uniquely to an entity. Wong
et al. [?] proposed the $(X,Y)$-anonymous method, where $X$ and $Y$ are joint attributes of
records. However, both $k$-anonymity and subsequent extension methods have a weakness
of “joint attribute attacks” which is that with high probability the attacker can infer the
recorded private information if he cross-matches the data in other public databases or his
other background knowledge with the records in a database that satisfies the $k$-anonymity.
Machanavajjhala [?] proposed a diversity principle, also known as $l$-diversity, to prevent
this type of attack. $l$-diversity requires that each group of QIDs contains at least $l$ different
values in a privacy attribute. $l$-diversity is definitely satisfied with $k$-anonymity if $k \leq l$
because at least $l$ records are included in each QID group. However, if the distribution of
sensitive data and global data in some QID groups differ greatly, the attacker may still infer
the private information of the target record with high probability. To response this attack
mode, Li et al. [?] proposed the $t$-closeness method which considers the distance between
the privacy data and the overall data for each QID group.

Moreover, Lu et al. [?] proposed a general preserving method called $\langle k,s \rangle$-privacy
which is basing on the generator of virtual nodes to blur geographical location. $\langle k,s \rangle$-
privacy is blurring the target position into $k$ private locations and their area is no smaller
than $s$. Specifically, they proposed two dummy methods called CirDummy and GridDummy
to realize $\langle k,s \rangle$-privacy, where CirDummy is dedicated to generating $k - 1$ additional nodes
in a virtual circle which contains the user’s real position with a random center and an area
of $k \cdot s$ and GridDummy is meant to produce a big virtual square which consists of $k$ small
squares with an area of $s$ and make the user’s real position be a point of any small square.
However, this algorithm has many deficiencies. Firstly, $\langle k,s \rangle$-privacy algorithm needs two
parameters $k$ and $s$, while fixed $k$ and $s$ can not be adapted to all regions since population
densities vary widely from place to place. Secondly, in the design of $\langle k,s \rangle$-privacy, there
is no quantification of the actual degree of privacy preservation, so that the user can not
understand how much private it is if he/she took different privacy parameters. Last but not least, the privacy parameters may be too large for many of the service providers, so it has basically lost the value of the data because these large data can not be effectively used. Niu et al. [?] proposed dummy-location selection and its enhanced algorithms based on the entropy metric to ensure that all the regions are far away enough, which achieved \( (k, s) \)-privacy but suffered from lack of dynamics.

In addition, no matter whether it is \( k \)-anonymity, \( l \)-diversity or \( t \)-closeness, they all have corresponding attacking modes that invalidate their privacy algorithms. The primary reason is that there is no rigorous mathematical definition of the attack model and no quantitative indicators of the background knowledge of the attacker. Dwork [?] first presented the differential privacy method. The first survey summarized by Dwork et al. [?] repeated the definition of differential privacy and one of its implementation mechanisms aimed at exhibiting how to apply these techniques to data publishing and they used the difficulties encountered in the data publishing process to reflect forward-looking solutions in statistical analysis [?]. Followed by the review [?], they outlined the main incentive scenarios and summarized future research directions. Task et al. [?] applied differential privacy to social network analysis based on graph theory. A book written by Dwork et al. [?] provided an accessible starting point for anyone who wanted to study the theory of differential privacy. Furthermore, Dwork et al. [?] introduced the concentrated differential privacy which was a relaxation of differential privacy enjoying better accuracy.

In addition to differential privacy, cryptographic models have also been applied to protect user privacy. Liu et al. [?] proposed two privacy-preserving query schemes to protect location services in Internet of vehicles by taking the advance of fog computing and by applying oblivious transfer and ciphertext-policy attribute based encryption. Wang et al. [?] presented the first protocol to protect private data by computing over encrypted data using Paillier’s homomorphic property and they also provided two optimization methods to improve their proposed protocol.

In this article, instead of regularly using the static \( (k, s) \)-privacy algorithm, we exploit a new virtual circle technique to directly fuzz the geographical location of a POI which achieves our novel \( (r, h) \)-privacy that leads to dynamic privacy guarantees, i.e., it supports user-defined degree of geographical privacy preservation. We not only consider user’s location privacy but also take user’s social network privacy into account. Therefore, we present a FRP method to concern user’s cyber links that may be revealed any time. Contrary to Daniele’s complicated POI-Ti-Dico method, we resort to differential privacy to implement a more lightweight friend relationship privacy-preserving framework. We make a further step to focus on differential privacy with its Laplacian mechanism because we have to calculate the social relationship factor which is a numeric value between users in the recommendation system. To the best of our knowledge, we are the first to add Laplacian noise during the process of calculating social relationship factor but not the final recommendation result in order to smooth the weights among friend users to avoid social relationship attacks. Through this idea, we realize and prove our privacy-preserving FRP algorithm enjoying differentially private guarantees in recommendation servers.
3. Background

In this section, we first investigate the LBSN recommendation system that simultaneously considers the similarity between users, the relationship between the user and his friends, and the user’s geographical location, see e.g., [? ? ? ? ? ], which serve as the building blocks in our privacy-preserving approaches to exploit fuzzy location and friend relationship influence. Next we review the concept of differential privacy that will be applied to protect friend relationship privacy in Section 4.2. Finally, we give some examples of geographical and friend relationship privacy attacks to explain why we aim to propose privacy-preserving approaches.

3.1. Collaborative POI recommendation with social and geographical influence

User based collaborative filtering. Let \( U \) and \( L \) denote the user set and POI set, \( c_{i,j} = 1 \) indicates the user \( u_i \in U \) visits POI \( l_j \in L \) and \( c_{i,j} = 0 \) means there is no record of \( u_i \) visiting \( l_j \). We record the probability that the user \( u_i \) will visit \( l_j \) as \( \Pr[c_{i,j}] \), and it can be calculated by

\[
\Pr[c_{i,j}] = \frac{\sum_{u_k \in U} \omega_{i,k} \cdot c_{k,j}}{\sum_{u_k} \omega_{i,k}}, \tag{1}
\]

where \( \omega_{i,k} \) represents the similarity between user \( u_i \) and user \( u_k \). We have some ways to calculate \( \omega_{i,k} \) such as cosine similarity, pearson correlation coefficients, etc. We select the cosine similarity which is commonly used in most of the POI recommendation work, i.e.,

\[
\omega_{i,k} = \frac{\sum_{l_j \in L} c_{i,j} \cdot c_{k,j}}{\sqrt{\sum_{l_j \in L} c_{i,j}^2} \sqrt{\sum_{l_j \in L} c_{k,j}^2}}.
\]

Friend relationship based collaborative filtering. POI recommendation based on social network can be realized by collaborative filtering based on friend relationship which is similar to Eq.(1). It is defined as

\[
\Pr[c_{i,j}] = \frac{\sum_{u_k \in F_i} SI_{k,i} \cdot c_{k,j}}{\sum_{u_k \in F_i} SI_{k,i}}, \tag{2}
\]

where \( F_i \) denotes the user \( u_i \)'s friend set and \( SI_{k,i} \) represents directional social influence weight \( u_k \) has on \( u_i \). \( SI_{k,i} \) is not equal to \( SI_{i,k} \) all the time. One way to compute the social influence between two friends is based on both of their social links and similarity of their check-in behaviors.

\[
SI_{k,i} = \gamma \cdot \frac{|F_k \cap F_i|}{|F_k \cup F_i|} + (1 - \gamma) \cdot \frac{|L_k \cap L_i|}{|L_k \cup L_i|}, \tag{3}
\]

where \( \gamma (\gamma \in [0, 1]) \) is the tuning parameter and \( L_k \) denotes the POIs that \( u_k \) has visited.

Geographical location influence. Analogous to [? ? ], we use the exponential distribution and choose the naive Bayesian method to calculate the geographic factor values. For a user \( u_i \) and its visited POI set \( L_i \), the probability that the user \( u_i \) will visit a new \( POI_i \) is defined as

\[
\Pr[L_i] = \prod_{l_m, l_n \in L_i \wedge m \neq n} \Pr[d(l_m, l_n)],
\]
where $d(l_m, l_n)$ represents the distance between POI $l_m$ and $l_n$. $Pr[d(l_m, l_n)]$ follows the exponential distribution and $Pr[d(l_m, l_n)] = a \times d(l_m, l_n)^b$ \[72x130\]. There is a hypothesis that the distances of all POI pairs are independent of each other. Thus, for a given POI $l_j$, the probability of the user visiting it can be obtained by using

$$Pr[l_j | L_i] = \frac{Pr[l_j \cup L_i]}{Pr[L_i]} = \frac{Pr[L_i] \times \prod_{l_y \in L_i} Pr[d(l_j, l_y)]}{\prod_{l_y \in L_i} Pr[d(l_j, l_y)]}. \quad (4)$$

Hence, we can calculate the probability $Pr[l_j | L_i] (l_j \in L - L_i)$ of the POI that the user does not check in yet and sort it in descending order, and then recommend top-$K$ POIs to the user.

As discussed, we can integrate the user similarity factor, the friend relationship factor and the geographical location factor into a linear function, then calculate the probability that the user will check in a new POI. Let $S_{i,j}$ be the probability that user $u_i$ will check in the POI $l_j$ and let $S^a_{i,j}, S^s_{i,j}, S^g_{i,j}$ be the same probability, corresponding to the factors of user similarity, the friend relationship, and geographical location respectively. Then, $S_{i,j}$ is defined as

$$S_{i,j} = (1 - \alpha - \beta)S^a_{i,j} + \alpha S^s_{i,j} + \beta S^g_{i,j}, \quad (5)$$

where $\alpha$ and $\beta$ are the two weighting parameters satisfying $0 < \alpha + \beta \leq 1$. In order to obtain these three probabilities, we could estimate the check-in probabilities $p^u_{i,j}, p^s_{i,j}$ and $p^g_{i,j}$ for a user $u_i$ to visit a POI $l_j$, respectively. Here, we compute them according to Eq.(1), Eq.(2) and Eq.(4), respectively. And it is necessary to normalize the probabilities:

$$S^a_{i,j} = \frac{p^u_{i,j}}{Z^u_i}, \quad where \quad Z^u_i = \max_{l_j \in L - L_i} \{p^u_{i,j}\},$$

$$S^s_{i,j} = \frac{p^s_{i,j}}{Z^s_i}, \quad where \quad Z^s_i = \max_{l_j \in L - L_i} \{p^s_{i,j}\},$$

$$S^g_{i,j} = \frac{p^g_{i,j}}{Z^g_i}, \quad where \quad Z^g_i = \max_{l_j \in L - L_i} \{p^g_{i,j}\},$$

where $Z^u_i, Z^s_i, Z^g_i$ are the normalization terms.

3.2. Differential privacy for preserving friend relationship

**Definition 3.1 ((\(\epsilon, \delta\))-differential privacy).** \[72x130\] A randomized mechanism $M_{priv}(\cdot) : D \rightarrow S$ gives $(\epsilon, \delta)$-differential privacy if for every set of outputs $\Omega \subseteq S$ and for two adjacent datasets of $D$ and $D'$, $M_{priv}$ satisfies

$$Pr[M_{priv}(D) \in \Omega] \leq e^\epsilon \cdot Pr[M_{priv}(D') \in \Omega] + \delta, \quad (6)$$

where datasets $D$ and $D'$ are adjacent if they differ in at most one record. Its strictest definition does not include the additive term $\delta$, i.e., if $\delta = 0$, the randomized mechanism $M_{priv}(\cdot)$ gives $\epsilon$-differential privacy. $(\epsilon, \delta)$-differential privacy provides freedom to strict differential privacy for some low probability events. $\epsilon$-differential privacy is usually called pure differential privacy, while $(\epsilon, \delta)$-differential privacy with $\delta > 0$ is called approximate differential privacy \[72x130\].
In Definition 3.1, the private parameter $\epsilon$ indicates the privacy budget which gives strong privacy guarantees with a small $\epsilon$. Differential privacy has some particularly useful properties in our works such as composability and robustness of auxiliary information. Composability refers that if all the mechanisms are differentially private, then so is their composition. Robustness means that auxiliary information of the adversary can not break the privacy guarantee.

**Definition 3.2 (Sensitivity).** For a function $f : D \rightarrow \mathbb{R}^k$, and two adjacent datasets $D$ and $D'$, the sensitivity $\Delta f$ of $f$ is defined as

$$\Delta f = \max_{D,D'} \| f(D) - f(D') \|_1.$$  \hspace{1cm} (7)

Sensitivity $\Delta f$ is only related to the type of function $f$. Intuitively, when $k = 1$ the sensitivity of $f$ is the maximum difference in the values that function $f$ may take on a pair of databases that differ in only one record.

There are two basic mechanisms meeting Definition 3.1, which are widely used currently to realize differential privacy. One is the Laplace mechanism and the other is the exponential mechanism. Laplacian mechanism uses the sensitivity as a parameter and adds Laplacian noise to the output of the function. But for non-numeric queries, differential privacy uses an exponential mechanism for noisy results because Laplace mechanism failed to solve this problem. We will use Laplacian mechanism in our FRP algorithm (i.e., Algorithm 4.8) since our dataset is numeric.

### 3.3. Attacking privacy model

It is more and more popular to use mobile APPs nowadays. An infrastructure is rapidly developing that encompasses a great number of users equipped with mobile terminals such as mobile phones that posses location-targeting capabilities, e.g., built-in GPS receivers, and datacom capabilities. At the same time, people like adding and making friends on many social networking platforms which always ask for their personal information. Recommendation systems also work like this so it always leads to privacy leaks. We consider two different private attacking models: one is a geographical attacking model and the other one is a friend relationship attacking model.

**Geographical attacking model.** Location-based services are increasingly becoming available that return results relative to the locations of their users. In recent works, the fuzzy information of the user’s real-time position was sent to the system in his client to obscure the current position. It is worth noting that the location of the check-in POI is accurate while the current real-time position is still obscure. The attacker can still obtain the user’s real-time geographic location from the relevant information of the POI, e.g., when user Anna chooses to record and share directly at a POI, such as the lake P of X University, her friends can immediately receive the check-in information. The potential attacker will find her current position easily basing on the POI’s information. Therefore, the privacy guarantee is not enough and the position which the user creates by himself/herself needs to be blurred.
Friend relationship attacking model. In the POI recommendation system, the user’s preference of POIs is personal privacy information. The attacker can easily derive the user’s private sensitive information, such as his political tendency, religious belief, and even sexual orientation and so on, from the user’s preferred POIs. We assume that each user could be a potential attacker, and they can rebuild the user’s preferred POIs with their background knowledge by submitting right queries [? ]. The details are illustrated in the following two examples.

![Fig. 2. Attacking mode of POI preference](image)

Example 3.3. Assuming that user Bron has been using the recommendation system service for some time, the system has also recorded his POI preference. And Anna, a curious friend of Bron, often goes out with Bron together, such as having dinner and signing in the restaurant, watching the movies and signing in the cinema and so on. Recently Anna learns that Bron likes to go to city Y alone every Saturday night, but she does not receive his check-in information from the system, i.e., Bron may select the private check-ins which are not instantly shared to his friends. Anna wants to know where Bron went every Saturday night. So Anna submits a query to the system. As shown in Fig. 2, the query: “What are the POIs nearby?” is firstly sent to the system server by Anna. The returned result would be top-K POIs which are recommended to Anna with highest probability. There would be an unusual POI returned such as “Hell Bar” and the unusual POI is not located in their city but in the city Y. According to Anna’s background knowledge, she can infer that “Hell bar” is the POI where Bron went every Saturday night with high probability.

In Example 3.3, Anna and Bron have high user similarity $S_{A,B}$, and they are friends so their friend relationship $S_{s,A,B}$ is also high. In the top-K results returned to Anna, some abnormal positions appear. These abnormal positions are far away from her current location, i.e, the geographical location value $S_{g,A,B}$ is very low. Anna understands the recommendation algorithm so she infers that such anomaly result is obtained mainly because her best friend Bron has a similar preference. The following example shows another probable attacking mode because of the similarity between users.

Example 3.4. It is still assumed that Bron has used the recommendation system service for some time, the system has also recorded his POI preferences, which include several unusual POIs: “Game Restaurant” (POI$_a$), “Wonder Clothing Store” (POI$_b$), “Homosexual Museum” (POI$_c$) and “Hell Bar” (POI$_d$). Bron selected to share the check-in information of the POI$_a$, POI$_b$ and POI$_c$ with his friends, but Bron would like to check in the POI$_d$
secretly. Anna, a curious friend of Bron, doubts that Bron often goes to a bar called “Hell Bar” (POI$_d$) which does not have a good reputation. So Anna sends a virtual information to the server to tell that the POI$_d$ is one of her preferred POIs. Then the server return Anna a message: “People who like POI$_d$ also often like POI$_a$, POI$_b$, POI$_c$.” Since these three POIs belong to unusual ones, and Bron also shared the three POIs publicly, Anna can infer that POI$_d$ is also Bron’s preferred POI with high probability.

In Examples 3.3 and 3.4, we have assumed that the attacker knows the recommendation algorithm, but not the value of each parameter, e.g., $\alpha$, $\beta$ in Eq.(5) yet. These examples show us two different attack modes based on friend relationship. We will solve these problems in Section 4.2.

4. Privacy-preserving approaches

We investigate two attacking models which may reveal the private information of the user’s location and friend ties in the previous sections so as to present the corresponding privacy-preserving algorithms for them and then evaluate the utility and performance of our approaches in detail. We also build the two methods into the recommendation framework as our novel idea to return recommendations enjoying privacy guarantees followed by strict theoretical analysis. To begin with, we introduce the following two definitions that will be used throughout this section.

Definition 4.1 (Differential entropy). [? ] A differential entropy $H(X)$ for a continuous random variable $X$ with its density function $f(x)$ is defined as

$$H(X) = -\int_{Q} f(x) \log f(x) dx,$$

where $Q$ is the support set of $X$, i.e., $Q = \{x | f(x) > 0\}$.

Analogous to the definition of privacy loss in [? ], we present privacy gain to meet our needs for our privacy-preserving algorithms.

Definition 4.2 (Privacy gain). Let $H(X)$ ($X \in D^n$) be the entropies of each record in the database $D^n$ and $H'(X)$ ($X \in D'^n$) be the entropies of each record after a statistical release. Therefore, privacy gain can be defined as the maximum information gain of a record in the database, i.e., $\max_{i \leq n}(H'(X_i) - H(X_i))$.

For a given database statistics release, the above definition links the differential privacy guarantee to the largest information gain of a single record in the database.

4.1. GLP: Geographical location privacy-preserving algorithm

In this part, we will present the GLP algorithm (i.e., $\langle r, h \rangle$-privacy algorithm) to preserve location privacy and deal with the shortcomings of $\langle k, s \rangle$-privacy [? ] as well for the geographical attacking model discussed in Section 3.3.
4.1.1. Enforcing \((r, h)\)-privacy

Our \((r, h)\)-privacy inherits the idea of \((k, s)\)-privacy but we do not generate virtual nodes any more. We directly make the user position blur into a virtual circle, and the precise position may exist anywhere in the circle. In this ambiguous mode, the probability of the attacker inferring the user’s real position approaches 0. It is noteworthy that the attacker may also have a kind of violent way to find the user, but the \((r, h)\)-privacy algorithm is able to adjust the radius of the circle based on the local population, allowing the user to have enough time to leave without being found. We also assume that an attacker can obtain all the information from the server, as the \((k, s)\)-privacy. Note that if a user wants to send a position to the server, a fuzzy one will simultaneously be sent in our assumption.

**Definition 4.3 ((r, h)-Privacy).** If a geographical location algorithm turns a user position to a larger virtual circle (i.e., privacy area) with radius \(r\) based on local population density \(h\) and the user can move anywhere in the circle, then the algorithm satisfies \((r, h)\)-privacy.

Our \((r, h)\)-privacy algorithm shown in Algorithm 4.4 takes both population density and private geo-location into account. In our calculation, the population density \(h_i\) of \(POI_i\) is determined by \(h(POI_i)\), i.e., its total number of historical check-ins (line 1). We calculate the radius of the virtual circle that needs to blur the current \(POI_i\) based on \(h_i\) according to Eq.(9) (line 2). The size of the radius is determined by whether the current \(POI_i\) is in a densely populated or sparsely populated place (lines 3-7). Then we calculate the coordinate of the new center \(o'_i\) after selecting a random angle \(\theta\) and appropriate distance \(l\) (lines 8-9). Finally, we generate a virtual circle as privacy area with center \(o'_i\) and radius \(r_i\) (line 10).

**Algorithm 4.4. \((r, h)\)-Privacy algorithm**

**Input:** \(POI_i\) with coordinate \((x_i, y_i)\)

**Output:** center \(o'_i\) and radius \(r_i\)

1: \(h_i \leftarrow\) total number of historical check-ins \(h(POI_i)\)
2: \(r_i \leftarrow R(h_i)\) according to Eq.(9)
3: if \(h_i < 3\) then
4: \(l \leftarrow\) random\((r_i/2, r_i)\)
5: else
6: \(l \leftarrow\) random\((0, r_i)\)
7: end if
8: \(\theta \leftarrow\) random\((0, 2\pi)\)
9: Determine center \(o'_i\) with coordinate \((x_i + l \cos \theta, y_i + l \sin \theta)\)
10: \(POI_i \leftarrow\) blur \(POI_i\) into a circle with center \(o'_i\) and radius \(r_i\)
11: **return** \(o'_i\) and \(r_i\)

There are several ways to obtain the virtual circle radius \(r_i\). Here we use a simple and effective method, i.e., a linear function of \(h_i\), to calculate:

\[
R(h_i) = -\frac{r_{\text{max}} - r_{\text{min}}}{h_{\text{max}}} \cdot h_i + r_{\text{max}},
\]  \(9\)
where $r_{\text{max}}$ and $r_{\text{min}}$ represent the upper and lower bounds of the virtual circle radius respectively (their computations will be discussed later in Section ??), and $h_{\text{max}}$ is the maximum number of historical check-in numbers in all POIs ($h_{\text{max}} = \max\{h(\text{POI}_t), t = 1, 2, \cdots, n\}$). $r_{\text{min}}$ is set to prevent some hot POIs from losing privacy guarantee since the virtual circle will be very small when $h_t$ increases.

We will also face some extreme situations when using $\langle r, h \rangle$-privacy. To exemplify, the historical check-in numbers of some POIs are only 1, and then the area of virtual circle reaches maximum. Under this situation, attackers can quickly locate the precise position of the POI and catch the target user. Hence, we let the distance between the user’s real position $p$ and the virtual center $o'$ satisfy $\text{dist}(p, o') \in [r/2, r]$ to ensure that the two points are not too close, where $r$ is the radius of the virtual circle.

4.1.2. Theoretical analysis of $\langle r, h \rangle$-privacy

We will in this section provide an attractive theoretical analysis of GLP algorithm by using differential entropy which supplies a quantitative indicator for the effectiveness of blurring, i.e., the privacy guarantee.

**Theorem 4.5 (Distance distribution).** In the $xOy$ cartesian coordinate system shown in Fig. 3, there is a point $C(c, 0)$ on the $x$-axis and a circle with center $O$ (the origin) and radius $r$ ($0 < r < c$). Let $P(x_0, y_0)$ be any point in the circle and $z$ be the distance between $C$ and $P$. Then the probability density function of this distance is $f(z) = 2z\psi/\pi r^2$.

![Fig. 3. Geometric schematic of distance distribution](image)

**Proof.** The two-dimensional random variable $(x_0, y_0)$ of point $P$ in the circle follows the uniform distribution, so it obeys $U(x_0^2 + y_0^2 \leq r^2)$ and their joint probability density function is $1/(\pi r^2)$.

To facilitate the calculation, we consider the polar coordinate system with pole $C$ and polar axis $x$. From Fig. 3 and some geometric knowledge, we know that the cumulative distribution function of distance $z$ is the area of the shadow region, and its probability density function $f(z)$ can be computed as the quotient of the length of the arc with angle $2\psi$ (marked as red in Fig. 3) and $\pi r^2$, which is $2z\psi/\pi r^2$. \hfill \IEEEQEDboxed

When the $\langle r, h \rangle$-privacy algorithm is added to the normal POI recommendation system, POIs are blurred into virtual circles, the distance between any two POIs is required when calculating the geographical location factor according to Eq.(4). We define the distance $z_{ij} = d(\text{POI}_i, \text{POI}_j)$ between two private fuzzy POIs as the distance between the center of
one virtual circle and any point of the other virtual one with radius \( r \) so as to simplify the calculation, i.e., the distance between \( C \) and a random point in the circle \( O \) in Fig. 3. In fact, the definition of this distance can be verified if we tell the random point in another virtual circle also follows the uniform distribution, hence, we simplify the calculation is reasonable.

For computing this distance, suppose \( c = t \cdot r \) \((t > 1)\), then \( z^2 = r^2 + c^2 - 2rc \cos \vartheta = r^2(1 + t^2 - 2t \cos \vartheta) \) and \( \frac{\sin \psi}{r} = \frac{\sin(\vartheta + \psi)}{r} \), which leads to \( \tan \psi = \frac{\sin \vartheta}{t - \cos \vartheta} \). Meanwhile, let \( x = \cos \vartheta \in [-1, 1] \) be the integral variable, therefore, the expectation of this distance is

\[
E(z) = \int_{c-r}^{c+r} \left| z \cdot \frac{2z\psi}{\pi r^2} \right| dz
\]

\[
= \int_{-1}^{1} \left| \frac{r \sqrt{1 + t^2 - 2tx}}{t - x} \right| \frac{d}{dx} \ln \left( \frac{t - x}{t} \right) \frac{d}{dx} \frac{\sqrt{1 - x^2}}{\pi r^2} dx
\]

\[
= \frac{2tr}{\pi} \int_{-1}^{1} \frac{1}{t - x} \ln \left( \frac{t - x}{t} \right) \frac{\sqrt{1 - x^2}}{\pi r^2} dx
\]

\[
\approx \frac{2tr}{\pi} \cdot (0.9e^{-1.5t} + 1.6).
\]

In addition, POIs blurring into circles result in the uncertainty of the distance, which brings information increment. Therefore, the difference of differential entropy for computing the private distance and normal distance is

\[
\Delta H(z) = H(f(z)) - H(1)
\]

\[
= - \int_{c-r}^{c+r} \left( \frac{2z\psi}{\pi r^2} \right) \ln \left( \frac{2z\psi}{\pi r^2} \right) dz
\]

\[
= \int_{-1}^{1} \frac{2t}{\pi r} \arctan \left( \frac{\sqrt{1 - x^2}}{t - x} \right) \ln \left( \frac{2r \sqrt{1 + t^2 - 2tx}}{t - x} \right) \arctan \left( \frac{\sqrt{1 - x^2}}{t - x} \right) dx
\]

\[
\approx \frac{2t}{\pi r} \cdot \ln 2 \cdot (0.3e^{-0.5t} + 0.02) \text{ (bits) > 0}.
\]

We conclude this section in the following theorem.

**Theorem 4.6 (Privacy gain of GLP).** *When adopting GLP algorithm to obtain the location factor, instead of the normal method by Eq.(4), we will get a reasonable privacy gain, which is positive because of Eq.(10).*

### 4.2. FRP: Friend relationship privacy-preserving algorithm

We will in this section present the FRP algorithm based on differential privacy for the friend relationship attacking models discussed in Section 3.3.

#### 4.2.1. Designing FRP algorithm

In the social networking recommendation system, each user’s friends are potential attackers. The attacker may infer the target user’s privacy information with high probability if the user is very close to his friends (i.e., the value of friend relationship is high) in the POI recommendation system. Our FRP algorithm tries to add enough noise to the factor of
friend relationship, so that the attacker will not infer that his friends would have any connection with the returned results from the system and will not obtain any relevant private information.

We choose $\epsilon$-differential privacy and Laplace mechanism to build our FRP algorithm. The following corollary will be directly used in the design of the FRP algorithm.

**Corollary 4.7 (Laplacian $\epsilon$-differential privacy).** [? ] For function $f : D \to \mathbb{R}^k$, suppose that its sensitivity is $\Delta f$, then $f(D) + \text{Lap}_k(\Delta f/\epsilon)$ satisfies $\epsilon$-differential privacy, where $\text{Lap}_k(\Delta f/\epsilon)$ is a $k$-dimensional vector obtained from the Laplacian distribution with a position parameter of 0 and a scale parameter of $\Delta f/\epsilon$.

The FRP algorithm is shown in Algorithm 4.8. We first replace one user $u_i$’s one friend relationship link from his friend set $F_i$ randomly, i.e., $u_i$ has a new unknown friend, and then generate a new neighbor set $F_i'$ (line 1). Then, we calculate the new social influence list $SI_i(F_i')$ of user $u_i$ basing on $F_i'$ (line 2). Thereby, we can obtain the global sensitivity $\Delta f$ of the query function according to Eq.(7) (line 3). Then we set position and scale parameters $(\mu, \lambda)$ of the Laplacian noise to be $(0, \Delta f/\epsilon)$ as to Corollary 4.7 (line 4). Therefore, we can get the Laplace distribution and add Laplacian noise to the original friend relationship factor and get the noisy social influence set $SI_i'$ of $u_i$ finally (lines 5-9).

**Algorithm 4.8. Friend relationship privacy-preserving algorithm**

**Input:** privacy parameter $\epsilon$, $u_i$’s friend set $F_i$ and his social influence list $SI_i(F_i)$ for all other users $u_k \in F_i$

**Output:** $u_i$’s noisy social influence list $SI_i'$

1: $F_i' \leftarrow$ Replace $u_i$’s friend $u_{\text{random}} \in F_i$ with $u_{\text{unknown}} \notin F_i$
2: Calculate $u_i$’s new $SI_i(F_i')$ according to Eq.(3)
3: Sensitivity $\Delta f \leftarrow \max_{F_i,F_i'} \|SI_i(F_i) - SI_i(F_i')\|_1$
4: Laplacian parameters $(\mu, \lambda) \leftarrow (0, \Delta f/\epsilon)$
5: for each $u_k$ in $F_i$ do
6: \[ SI_{i,k}' \leftarrow SI_{i,k} + \text{Lap}_1(\lambda) \]
7: Append $SI_{i,k}'$ to list $SI_i'$
8: end for
9: return $SI_i'$

We can see that the friend relationship factor become smoother and closer, and it will not exist the situation that the friend relationship of some users are particularly high from the follow-up experimental results after we implement FRP in the recommendation system. Furthermore, the problems in Examples 3.3 and 3.4 are solved because there is no particularly high friend relationship factor value for Anna, then she can not infer any private information on Bron.

### 4.2.2. Theoretical analysis of FRP

We still use the differential entropy to evaluate the FRP algorithm because it mainly adopts wiping out the friend relationship values with great differences for fuzzy implementation. There is a property that the bigger difference between users’ friend relationship
factor, the greater the information entropy, i.e., the greater uncertainty and the smaller the probability that the attacker derives the user’s private information from the recommended POIs.

**Theorem 4.9 (Privacy gain of FRP).** Friend relationship factor with differential privacy increases the information entropy and improves the uncertainty of the friend relationship distribution, i.e., privacy gain of FRP algorithm equals to $H(SI'_i) - H(SI_i) > 0$, which protects the user’s information privacy.

**Proof.** We calculate the difference of $SI'_i$ and $SI_i$’s differential entropy to get the information increment, i.e.,

$$H(SI'_i) - H(SI_i) = H(SI_i + \text{Lap}^1(\frac{\Delta f}{\epsilon})) - H(SI_i)$$

$$= H(\text{Lap}^1(\frac{\Delta f}{\epsilon}))$$

$$= - \int_{-\infty}^{+\infty} \text{Lap}^1(\frac{\Delta f}{\epsilon}) \log(\text{Lap}^1(\frac{\Delta f}{\epsilon})) dx$$

$$= \log(\frac{2\Delta f}{\epsilon}) \text{ (bits)}$$

where $\text{Lap}^1(\Delta f/\epsilon) = \frac{1}{2\Delta f/\epsilon} \exp(-\frac{|x|}{\Delta f/\epsilon})$ and the second equation is obtained since the Laplacian noise and $SI_i$ are independent. Then

$$2^{H(SI'_i) - H(SI_i)} = 2^{\log(\frac{2\Delta f}{\epsilon})} = \frac{2\Delta f}{\epsilon},$$

where $0 < \epsilon < 1$, and $\Delta f = 1$ if there exists a user having only one friend. Then, we have $2 \Delta f/\epsilon > 1$, which implies $H(SI'_i) - H(SI_i) > 0$. The proof is completed. □

### 4.3. Private POI recommendation method based on GLP and FRP

Algorithms 4.4 and 4.8 constitute our novel private recommendation method by adding the privacy-preserving information on the aspects of geographical location and friend relationship. We will again employ differential entropy to measure the degree of privacy guarantee.

**Lemma 4.10 (Functional entropy chain rule).** Let $X, Y, Z$ be three discrete random variables which are mutually independent and $M = X + Y + Z$. There exists $H(M) = H(X) + H(Y) + H(Z)$.

**Proof.** Since $M$ is a function of $(X, Y, Z)$ and according to the chain rule of entropy, we have

$$H(M) \leq H(X, Y, Z) = H(X) + H(\text{Y}|X) + H(\text{Z}|X, Y) \leq H(X) + H(Y) + H(Z).$$

Furthermore, $(X, Y, Z)$ is also a function of $M$ and the random variables $X, Y, Z$ are independent of each other, the two equalities hold. The proof is completed. □
Theorem 4.11 (Total privacy gain). Under the assumption that the three parameters ($\alpha$, $\beta$ and $1 - \alpha - \beta$) are pairwise independent, the entropy of the probability distribution (Eq.(5)) obtained by our private recommendation algorithm is greater than the normal one and the total privacy gain is $H(2z\psi/\pi r^2) + H(\text{Lap}^1(\Delta f/\epsilon))$ bits.

Proof. It has been proved that the information entropy of geographical location and friend relationship increased after adding privacy-preserving algorithms according to Theorems 4.6 and 4.9, where $H(f(z)) = H(2z\psi/\pi r^2) > 0$ and $H(SI_i) - H(SI_i) > 0$. Based on Eqs.(5), (10), (11) and Lemma 4.10, the total privacy gain is given by

$$\Delta H = H(S'_{i,j}) - H(S_{i,j})$$

$$= H((1 - \alpha - \beta) S^u_{i,j} + \alpha S^s_{i,j} + \beta S^g_{i,j}) - H((1 - \alpha - \beta) S^u_{i,j} + \alpha S^s_{i,j} + \beta S^g_{i,j})$$

$$= H(S^u_{i,j}) - H(S^u_{i,j}) + H(S^s_{i,j}) - H(S^s_{i,j}) + H(S^g_{i,j}) - H(S^g_{i,j})$$

$$= H(S^u_{i,j}) - H(S^u_{i,j}) + H(S^s_{i,j}) - H(S^s_{i,j})$$

$$= H(S^u_{i,j}) - H(S^u_{i,j}) + H(S'_{i,j}) - H(S_{i,j})$$

$$= H(2z\psi/\pi r^2) + H(\text{Lap}^1(\Delta f/\epsilon)) > 0,$$

where the third equation is obtained due to the fact that $H(\nu S^u_{i,j}) = H(S^u_{i,j}) + \log \nu (\nu \in (0,1), * \in \{u,s,g\})$ and the fifth equation is because of $H(S^s_{i,j}) = H(\varsigma SI_i) = H(SI_i) + \log |\varsigma| (\varsigma \in \mathbb{R})$. □

Remark that $\Delta H$ is the incremental differential entropy after adding the privacy-preserving algorithm, and it represents the degree of privacy guarantee. For the friend relationship factor, differential privacy parameter $\epsilon$ and the query function sensitivity $\Delta f$ can be set by users in advance to ensure a reasonable privacy guarantee. Regarding to the location factor, in addition to the default radius of the virtual circle given by the system, users can also customize the radius, since the radius $r$ is also a potential control for privacy guarantee.

5. Empirical study

In this section, we will test the accuracy and privacy of the private algorithms and find out all the optimal values of the parameters, e.g., $\alpha, \beta, r_{max}$, to build an effective recommendation system with privacy guarantee.

5.1. Experiment setup
5.1.1. Dataset description

We conduct experimental studies using data crawled from Foursquare\(^1\) which is one of the most representative location-based services sites. The dataset also contains a large number of friend ties while many other datasets do not have them. The data we selected is from March 2010 to December 2011, including 24,941 users, 43,593 POIs and 2403,909 check-in records and 120,883 friend ties. As shown in Fig. ??, after summarizing the check-in records,

\(^1\)https://foursquare.com
the user and POI check-in matrix is generated and its density is only about $2.41 \times 10^{-3}$. Due to the sparsity of the check-in matrix, the information we can get is really limited and the effectiveness of the recommendation is usually not high enough. Therefore, we do mark off $x\%$ ($x = 10, 30, 50$, generally taking 30 by default) of POIs visited by the user randomly for each user to facilitate the evaluation of our algorithms. In the experiments, we apply the recovered POIs to evaluate the performance of the recommendation algorithm.

5.1.2. Evaluation metrics

To evaluate the statistical accuracy of top-$K$ ranking POIs, we apply two widely used metrics, namely, recall@$K$ and precision@$K$ in which $K$ is the number of recommended POIs. Recall@$K$ represents the fraction of labeled POIs that have been returned in the dataset among top-$K$ POIs, while precision@$K$ is the fraction of labeled POIs among top-$K$ POIs. Finally, let F-value be the harmonic mean of recall@$K$ (R) and precision@$K$ (P) to become a comprehensive indicator. Therefore, we have

\[
\text{recall}@K = \frac{|A \cap B|}{|A|},
\]

\[
\text{precision}@K = \frac{|A \cap B|}{|B|},
\]

\[
F = \frac{2}{\frac{1}{R} + \frac{1}{P}} = \frac{2RP}{R + P},
\]

where $A$ represents the labeled POI set in the dataset and $B$ represents the top-$K$ POIs and $K$ will be 5, 10, 20 and 50. We know that the first two metrics mentioned take values from $[0, 1]$, and larger values indicate better quality of recommended POIs.

5.1.3. Comparative approaches

To evaluate the effectiveness of our proposal, we compare our method with three other state-of-the-art methods that involve a location-aware social POI recommendation model:

- **USG [? ]**: This method unifies a POI recommendation framework, which fuses user preference to a POI with social influence and geographical influence. Note that USG model has no data obfuscation.

- **PPTR [? ]**: The model is for privacy-preserving trust-oriented POI recommendation, which involves a partially homomorphic encryption model.
• **PLAS**: This method is a location-aware social POI recommendation model using Paillier’s homomorphic property to protect private data, in which Paillier is a probabilistic public key cryptosystem.

• **PMLS**: PMLS stands for privacy-preserving method based on location sensitivity. It exploits differential privacy for location recommendation based on location sensitivity division, which is similar to our FRP that adopts social sensitivity in differential privacy.

5.1.4. Impact of parameter settings

In this section, we discuss the impact of parameter settings to ranking recommendation accuracy. We mainly analyze the impact of both the maximum and minimum radii of the virtual circle in the proposed GLP algorithm and the weight coefficients $\alpha, \beta$ for friend relationship factor and geographical location factor, respectively.

**Determining the weights of three POI factors.** In order to recommend top-$K$ POIs to a user, we need to calculate the probability $S_{i,j}$ according to Eq.(5). Here, we employ our private methods (Algorithms 4.4 and 4.8) to calculate $S_{i,j}^g$ and $S_{i,j}^r$ respectively. More importantly, we need to determine their weights, that is, the values of $\alpha$ and $\beta$. Let $K = 5$, we use various combinations of $\alpha$ and $\beta$ to test precision and recall of POI recommendation method, and we choose the values when the best performance is achieved, see Section ???. We also use the obtained values of $\alpha$ and $\beta$ for other top-$K$ recommendation.

![Fig. 5. The check-in number of POI (logarithmic)](image)

**Determining the radii in the GLP algorithm.** There are several parameters not determined yet in Eq.(9) for obtaining the radius of the virtual circle in our GLP algorithm. As shown in Fig. ??, the user number of $x$-axis and the POI check-in number of $y$-axis are log-processed, and it can fit into an approximate straight line, which indicates that the original data are better fitted to the exponential mechanism. We can see the check-in number $h$ of POIs starts at about 3500 and then drops rapidly until 1. The maximum $h_{max}$ can reach more than 10000, and overview all the POI historical records, only a small amount of POI can reach 10000 or so while the vast majority of the POI history check-in number even less than 10, i.e., the check-in number which is less than 100 accounts for nearly 99% of the total. Hence, the virtual circle radius of most POIs is very close to $r_{max}$ and it is more important
to determine the upper bound of the radius than the lower bound. Due to this fact, we set \( h_{\text{max}} = 100 \) by trial and error approach. Then, about 500(1\%) of the hottest POIs’ virtual radii will achieve the minimum \( r_{\text{min}} \). On the contrary, the number of unpopular POIs, whose history check-in numbers are 3 or less, exceeds 7000. Therefore, about 18\% of the POIs’ virtual radii will achieve the maximum \( r_{\text{max}} \). The empirical value of \( r_{\text{max}} \) is 200 (meters) so that the maximum area of the virtual circle is \( S_{\text{circle}} = \pi \cdot r_{\text{max}}^2 \approx 12500m^2 \), only 1/8 of maximum private area of \( \langle k, s \rangle \)-privacy algorithm [? ].

5.2. Experimental results

5.2.1. Weights of three factors when \( K = 5 \)

![Recall@5 of normal algorithm](image1)

![Precision@5 of normal algorithm](image2)

Fig. 6. Recall@5 and Precision@5 of normal algorithm

In this part, we conduct experiments on normal and private algorithms to determine the weights of the three POI factors when \( K = 5 \), respectively. In Fig. ??, the best performance of normal algorithm is achieved when \( \alpha \) and \( \beta \) are both equal to 0.1. The result is probably due to the fact that the user similarity factor plays a decisive role among the three factors. The higher value of user similarity usually makes contribution to the higher performance of the recommendation algorithm. At the same time, the factors of friendship and geographical location are also non-negligible for the recommendation algorithm since they at least take an account for 20\%. On the other hand, considering the three factors alone, we know that the friend relationship factor has the minimum effect on the performance of recommendation among all the factor combinations, so the contribution of the friend relationship factor is less than the others.

As shown in Fig. ??, in the adjustment of the linear parameters of the private recommendation algorithm, it can be seen that the private algorithm does not reduce the validity of the recommendations significantly, and both \( \alpha \) and \( \beta \) are exactly equal to 0.1 when recall rate and precision rate get the optimal point. From Fig. ?? and Fig. ??, we see that the private algorithm does not have a large impact on the accuracy of recommendation result based on both recall rate and precision rate, so we conclude that it is reasonable to add our private algorithms 4.4 and 4.8 into normal recommendation system.
5.2.2. Effect of varying $\epsilon$

Evaluation of our private algorithms need to be carried out in two aspects: one is the degree of privacy protection, and the other one is the performance of the recommendation algorithm. The experiments above has explained that the effectiveness of the private algorithm is as good as the normal one. The smaller $\epsilon$, the greater noise added in the factor of friend relationship, and the higher degree of privacy protection. While the privacy parameter $\epsilon = 0$, the private algorithm reduces to the normal one. The experimental results are shown in Fig. ??, where the information entropy of the privacy-preserving algorithm is higher than that of the normal version, and this also confirms the theoretical analysis of private algorithm in Section 4.3. For the different privacy parameters $\epsilon (\epsilon = 0.1, 0.3, 0.5, 0.7, 0.9)$, when $\epsilon$ become larger, the more close to 1, the added noise is relatively smaller, and then the information entropy is gradually reduced. When $\epsilon = 0.1$, the information entropy is already 9 times that of the normal version. And when $\epsilon = 0.5$, the information entropy has dropped rapidly to $1/5$ of that when $\epsilon = 0.1$. However, the information entropy is still more than double of the normal version, so we conclude that the privacy guarantee of the private algorithm has been greatly improved compared to the normal one. Fig. ?? also shows that the private algorithm achieves a good balance between privacy and accuracy in terms of recall rate, precision rate and $F$-value of them when $K = 10$.

5.2.3. Effect of varying $K$

We will use the optimal linear parameters $\alpha = 0.1$ and $\beta = 0.1$ for the following experiments. We also set $\epsilon$ as 0.5 to guarantee a proper privacy gain according to the experimental results just above. The recall rate @$K$ and the precision rate @$K$ (where $K = 5, 10, 20, 50$) of the recommendation algorithm for the normal and private algorithms are shown in Fig. ??.

No matter the value $K$ is, the performance of the normal and the private recommendation algorithm is roughly the same, and the private algorithm is not worse than the normal version in terms of the recall rate and precision rate. Note that the user-POI check-in matrix of our data is very sparse. The precision rate is 0.2 with a matrix sparseness of $2.41 \times 10^{-3}$ in our experiment.
5.2.4. Comparison in precision and recall with comparative approaches

Fig. ?? depicts the performance of Top-K ($K = 5, 10, 20, 50$) recommendations of all approaches, where $\epsilon$ is set as 0.5 as above. Fig. ?? (a) shows the recall@K performance, and Figs. ?? (b) shows the precision@K performance. See that the precision of our method degrades with the increase of $K$, while the recall upgrades. From Fig. ??, we can observe that the proposal outperforms PPTR, PLAS and PMLS in both precision and recall rates. For example, the precision@20 and recall@20 of the proposal are respectively 8.2% and 2.6% higher than PLAS. Note that the PMLS and our proposal both involve a differential privacy technique while our proposal is superior to PMLS in terms of all evaluation indicators, regardless of the change of $K$. This is mainly because that PMLS has to manually set the privacy parameter $\epsilon$ based on the number of POI check-ins, while we can automatically determine the radius of virtual circle according to Eq.(9). Hence, our approach is not only more effective but also easier to implement without having to spend a lot of labor costs. Besides, we can find that when $\epsilon = 0.5$, the ranking prediction accuracy of our proposal is just a little lower than USG, e.g., the precision@5 and recall@5 of our method are only 3.5% and 1.4% lower than that of USG. The results, together with the experimental results in Section ??, present that our proposal could achieve decent accuracy on the premise of privacy preservation and depict that a good balance could be made between the prediction accuracy and the goal of privacy preservation based on our proposal.
6. Conclusion and future work

Exploiting privacy guarantee in a POI recommendation system is significative and feasible without taking visible hit in accuracy of the recommendation. Our idea is to incorporate the user interest, fuzzy social ties and fuzzy geographical location in the recommendation. We propose two privacy-preserving algorithms, namely GLP and FRP, to offer privacy guarantee of geographical location and social relationships independently and provide a differential entropy method for demonstrating their privacy gains. We conduct a comprehensive performance evaluation over a large-scale real dataset collected from Foursquare. In our experiments, we tune enough parameters (e.g., the weights $\alpha, \beta$ of POI factors, privacy parameter $\epsilon$) and we see that applying the two privacy-preserving algorithms into a normal recommendation system does not decrease the accuracy of the recommended POIs in terms of precision and recall. The experimental results also show the robustness and effectiveness of the GLP and FRP algorithms.

Direction for future work could include more effective methods for the privacy gain of the private algorithms which need to be defined concretely because the definition of privacy is still a conundrum.

References


