Differentially Private Point-of-Interest Recommendation

name[a], name[a], name[a,*], name[b], name[a,c]

[a] *Department of Automation, Xiamen University, Xiamen 361005, China.*
[b] *Department of Mathematics, Xiamen University, Xiamen 361005, China.*
[c] ********

Abstract

We investigate the privacy-preserving problem for point-of-interest(POI) recommendation system in a differentially private way for the rapid growing location-based social networks (LBSNs). The recommender system needs to collect the relevant information of users so that providing them with potentially interesting content. The recommender algorithm has three factors: the user preferences, the recommendation between friends and the geographical locations. However, the user's sensitive information may be leaked when collected. There are two ways of the potential risk for privacy disclosure now so we propose two privacy-preserving algorithms for them respectively, without causing serious data availability issues. Then we add them in the recommender system to build a both recommender and privacy-preserving framework and conduct theoretical analysis to derive the privacy-preserving formula which can be adjusted to obtain different levels of privacy guarantees. Extensive experimental results demonstrate a good trade-off between privacy and accuracy of the proposed algorithms.

Keywords:
POI Recommendation System, Differential Privacy, Privacy Preservation

1. Introduction

1.1. Overview

With the rapid development of social network sites(SNSs), wireless networks 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. These LBSNs allow users to establish online links to their friends or other users, and share tips and experiences of their visits to plentiful point-of-interests (POIs), e.g., restaurants, stores, cinema theaters. In LBSNs, enhancing the effect of POI recommendation, aiming at recommending new POIs to users in order to help them explore new places and know their cities better, is very important and interesting because from the system we can get lots of important and valuable data, such as the connection between users, the contact between the POIs and users and so on. However, these information may also be too sensitive to the user, which is that the potential attacker will effectively derive the user's privacy information from the recommended result. Therefore, we must not only

*Corresponding author
Email addresses: email1 (name), email2 (name), email3 (name), email4 (name), email5 (name)

improve the accuracy of the POI recommendation system, but also take the user's privacy information into account.

Although it is difficult to define the privacy accurately, the definition of privacy disclosure is relatively simple. Any user's part of the privacy information was exposed to the attacker; a privacy disclosure of the user occurred. Normally, there are four types of privacy disclosure [32]: disclosure of identifiers, attributes, social relations and contact information. The introduction of these four kinds of privacy disclosure can be seen in many existing literatures and we will not repeat them here.

1.2. Technical Motivation

A collaborative POI recommendation system based on location is a double-edged sword. By gathering and processing most users' preferences it could provide interesting contents, increasing a recommender mobile APP's revenue and enriching user experience. On the other hand, users always have to expose their privacy information to obtain the better recommended results, which leads to privacy disclosure. For example, in the social network-based recommendation system, if one user only has a friend and he wants to go shopping, he will get the recommended result which is certainly from his only friend's shopping record [22]. And this makes all the purchase history information of his friend be leaked, although his friend would not like to share to him. The focus of this paper is on design, analysis, and experimental verification of a recommender system with built-in privacy guarantees. Differential privacy is a mathematically rigorous definition of privacy suited to analysis of large datasets and equipped with a formal measurement of privacy loss [9]. Moreover, differentially private algorithms will be taken by inputting a parameter, typically called ϵ, that indicates the permitted privacy loss in any execution of the algorithm and offers a concrete trade-off of privacy and utility. As the POI recommendation system may use the 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 traditional POI recommendation systems has been studied in [2, 3, 13, 14, 15, 16, 31]. Most of them hide the user's personal records from the recommendation system, while providing the POI results as suited as possible.

Existing location privacy-preserving techniques exhibit two significant limitations. First, some require a trusted third-party anonymizer that maintenances information of all user locations. Such a action may not always be available, and it could itself present security/privacy problems. Second, the underlying k-anonymity techniques is generally not adequate enough for location privacy, e.g. the privacy-area aware dummy generation algorithms for $\langle k, s\rangle$-privacy in [20], where it do not consider population densities and is 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, but stable 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, we intuitively need to use different values for k and s but this algorithm do not support receiving a parameter of population density.

Another challenge is protecting private cyber links information from disclosure in a concise way. There are usually two extensively studied buddy relationship attacking models and Daniele etc. [26] built a POI-Ti-Dico framework to solve them friendly by classifying user roles and cutting space area with different marks. However, a inadequacy of the POI-Ti-Dico

algorithm is that it lies in a completely new model of the real world, including the new division of space, the 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 normal recommender systems and brings a relatively high cost. Our idea is giving up the new framework and adopting the differential privacy in the existing recommendation system, which also has a certain degree of versatility.

1.3. Contributions

The main contribution of this work is to design and analyze a realistic recommender system built to provide differential privacy guarantees. The task is non-trivial: prior recommender systems are not designed with an eye towards modern privacy, and prior privacy research has focused on more modest algorithms without attempts at practical validation.

We give the biggest shortcoming and deficiency of the privacy-preserving algorithm without differential privacy. Also, we propose a reasonable mathematical definition for differential privacy and we study the implementation mechanism of differential privacy to put into noise, including Laplacian mechanism and exponential mechanism.

Our findings are that privacy does not need to come at substantial expense in accuracy. For the approaches we consider, privacy-preserving algorithm may start from the geographical location and friend relationships of people who use the recommender system. While there is some specialized analysis required, the methodology itself is relatively straight forward and accessible. Our new algorithms are as follows.

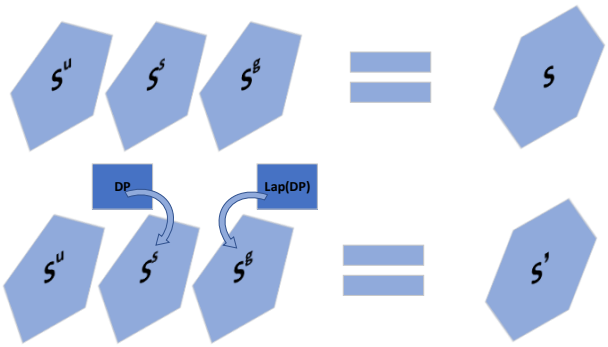


Fig. 1. Our contributions on differential privacy

Geographical Location Privacy-Preserving Algorithm(GLP). In the POI recommender system, there is an attack mode of the user's location privacy information. As shown in Fig.1, we add an special algorithm to protect the user's location information from disclosure and propose a controllable fuzzy geographical location algorithm in case.

Friend Relationship Privacy-Preserving Algorithm(FRP). We introduce the attacking methods of that attackers can derive the users privacy information by using the friend relationship in recommender system. And study the existing privacy-preserving algorithms for this kind of attacking model and point out their drawbacks. At last we propose a more lightweight and effective controllable algorithm which is shown in Fig. 1.

As an additional contribution of this note, we hope to demonstrate the integration of differential privacy technology to practical systems so we adopt a new evaluation method. We

evaluate the effectiveness of our algorithms theoretically with Shannon information entropy. And conduct enough experiments to prove that our two privacy-preserving algorithms are useful and effective. Finally we give an important formula and quantification standards which allow users to control their privacy-preserving levels by choosing suitable private parameters.

1.4. Outline

The rest of the paper is organized as follows. Section 2 gives some related works and Section 3 reviews the POI recommendation system, differential privacy and the private attacking models. Section 4 shows our two fuzzy privacy-preserving algorithms followed by theoretical analysis of them into private POI recommendation. Extensive experiments are conducted in Section 5 to demonstrate the effectiveness of the proposed methods. Section 6 concludes the work and give some advice on the future work.

2. Related Work

In 1977, Dalenius [5] first proposed private data protection and introduced the purpose of private database protection explicitly as follows:

- The attacker can not get any information in the database if no data is accessed.
- 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.

Sweeney [28] proposed the k-anonymity method in 2002 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. [30] proposed the (X, Y)-anonymous method in 2006, 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 [21] proposed a diversity principle, also known as l-diversity, to prevent this type of attack in 2006. 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. [19] proposed the t-closeness method which considers the distance between the privacy data and the overall data for each QID group in 2007.

Moreover, Lu et al. [20] proposed a general preserving method called $\langle k, s\rangle$-privacy which is basing on the generator of virtual nodes to blurred geographical location in 2008. $\langle k, s\rangle$-privacy is to blur the target position into k private locations and their area are 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 receive two parameters k and s and fixed k and s can not be adapted to all regions, i.e., it is not taken into account that 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 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. We improve the $\langle k, s\rangle$-privacy algorithm and propose our $\langle r, h\rangle$-privacy to avoid these deficiencies as much as possible.

On the other hand, no matter k-anonymity, l-diversity, 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 [9] first put forward the differential privacy method in 2006. Over the last decades, several surveys on differential privacy have been completed:

- The first survey summarized by Dwork et al. [6] repeated the definition of differential privacy and one of its implementation mechanisms aiming at exhibiting how to apply these techniques to data publishing;

- Dwork et al. [7] used the difficulties encountered in the data publishing process to reflect forward-looking solutions in statistical analysis. It identified several research issues in the analysis of data that had not yet been fully investigated at that time;

- In the review [8], Dwork et al. outlined the main incentive scenarios and summarized future research directions;

- Task et al. [29] applied differential privacy to social network analysis based on graph theory;

- Sarwate et al. [27] focused on maintaining the privacy protection of continuous data to solve the problems in signal processing;

- A book written by Dwork et al. [10] provided an accessible starting point for anyone who wanted to study the theory of differential privacy.

- Dwork et al. [11] introduced concentrated differential privacy which is a relaxation of differential privacy enjoying better accuracy.

We 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 recommender system. We first add Laplacian noise to the social relationship factor but not the final recommender result to smooth the weights between friend users to avoid social relationship attacks. Through this design we realize our original idea enjoying differentially private guarantees in public recommender servers.

3. Background

3.1. POI recommendation system

We investigate the location-based social network recommender system which considers the similarity between users, the relationship between the user and his friends, and the user's geographical location.

3.1.1. User similarity collaborative filtering

Let U and L denote the user set and POI set, $c_{i,j} = 1$ indicates the user $u_i \in U$ signed in 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

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

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.

$$\omega_{i,k} = \frac{\sum_{l_j \in L} c_{i,j} \cdot c_{k,j}}{\sqrt{\sum_{l_j \in L} c_{i,j}^2} \sqrt{\sum_{l_j \in L} c_{k,j}^2}} \tag{2}$$

3.1.2. Friend relationship collaborative filtering

POI recommendation based on social network can be realized by collaborative filtering based on friend relationship [17] which is similar to Eq.(1). It is defined as

$$Pr[c_{i,j}] = \frac{\sum_{u_k \in F_i} SI_{k,i} \cdot c_{k,j}}{\sum_{u_k \in F_i} SI_{k,i}} \tag{3}$$

where F_i is the user u_i's friend set. $SI_{k,i}$ represents the friendship degree of the user u_k to u_i, and $SI_{k,i}$ will not equals to $SI_{i,k}$ all the time. One way to compute the friendship degree between two friends is based on both of their cyber links and similarity of their check-in behaviors [18].

$$SI_{k,i} = \gamma \cdot \frac{|F_k \cap F_i|}{|F_k \cup F_i|} + (1 - \gamma) \cdot \frac{|L_k \cap L_i|}{|L_k \cup L_i|} \tag{4}$$

where $\gamma\,(\gamma \in [0, 1])$ is the adjustment parameter, F_k represents the user u_k's friend set and L_k is the POIs that u_k has visited.

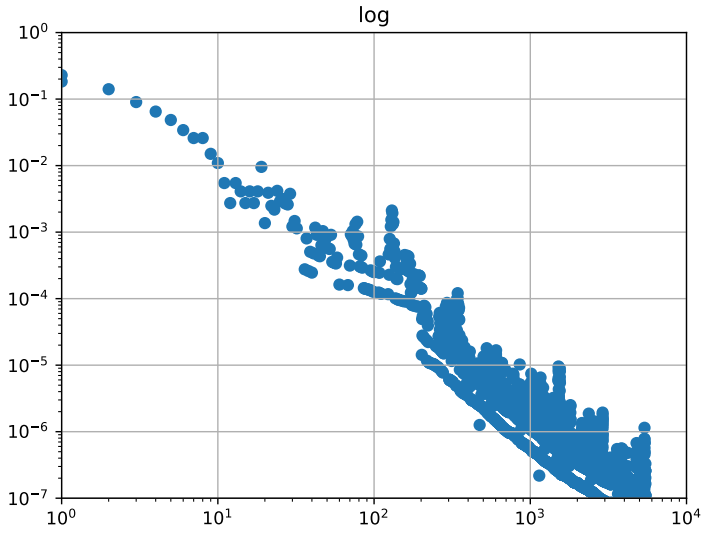


Fig. 2. Probability distribution of the distance of POI pairs

3.1.3. Geographical location factor

We intuitively think the check-in probability may follow the power-law distribution. Nevertheless, we observe from Fig. 2 that the check-in probability of POI pairs visited by the same user over distance does not follow a standard power-law distribution. But the linear part, that is, achieving the short distance, accounts for 90% of the overall data. Hence, we still use the exponential distribution to calculate the geographic factor values and choose the naive Bayesian method [31]. For a user u_i and its visited POI set L_i that the user has checked in, the probability that the user u_i will visit a POI is defined as

$$Pr[L_i] = \prod_{l_m, l_n \in L_i \wedge m \neq n} Pr[d(l_m, l_n)] \tag{5}$$

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 $Pr[d(l_m, l_n)] = a \times d(l_m, l_n)^b$. There is a hypothesis that the distance of each POI pair is independent for each other. Thus, for a given POI l_j, the probability of the user signing in the given POI can be obtained by using

$$Pr[l_j|L_i] = \frac{Pr[l_j \cup L_i]}{Pr[L_i]} = \frac{Pr[L_i] \times \prod_{l_y \in L_i} Pr[d(l_j, l_y)]}{Pr[L_i]} = \prod_{l_y \in L_i} Pr[d(l_j, l_y) \tag{6}$$

Hence, we can calculate the probability $Pr[l_j|L_i]$ $(l_i \in L - L_i)$ 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.

3.1.4. Overall POI factors

As discussed, we can integrate the user similarity factor, the friend relationship factor and the geo-location factor into a linear function, and calculate the probability that the

user will check in a POI. Let $S_{i,j}$ be the probability that user u_i will check in the POI l_j and then let $S^u_{i,j}, S^s_{i,j}, S^g_{i,j}$ represent the factors of user similarity, the friend relationship, and geographical location respectively, where $S_{i,j}$ is defined as

$$S_{i,j} = (1 - \alpha - \beta)S^u_{i,j} + \alpha S^s_{i,j} + \beta S^g_{i,j} \tag{7}$$

Here, it is necessary to normalize the probabilities:

$$S^u_{i,j} = \frac{S^u_{i,j}}{Z^u_i}, Z^u_i = max_{l_j \in L - L_i}\{S^u_{i,j}\}$$
$$S^s_{i,j} = \frac{S^s_{i,j}}{Z^s_i}, Z^s_i = max_{l_j \in L - L_i}\{S^s_{i,j}\}$$
$$S^g_{i,j} = \frac{S^g_{i,j}}{Z^g_i}, Z^g_i = max_{l_j \in L - L_i}\{S^g_{i,j}\}$$

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

3.2. Differential privacy

Definition 3.1 (Database). *[4] A database is a triple $D = (R, A, V_a | a \in A)$ following*

(1) R is a non-empty finite record set.

(2) A is a non-empty finite record set, and

(3) each attribute $a \in A$ is a functiona : $R \to V_a$, where V_a is the range of a, called the domain of a.

Let a triple $D = (R, A, V_a | a \in A)$ be a database, where R and A are non-empty finite record sets, and

Definition 3.2 ((ϵ, δ)-differential privacy). *[9] A randomized mechanism $M_{priv}(\cdot) : D \to S$ gives (ϵ, δ)-differential privacy for every set of outputs S and for two adjacent datasets of D and D'if M_{priv} satisfies*

$$Pr[M_{priv}(D) \in S] \leqslant e^{\epsilon} \cdot Pr[M_{priv}(D') \in S] + \delta \tag{8}$$

Its strictest definition does not include the additive term δ, i.e., if $\delta = 0$, the randomized mechanism $M_{priv}(\cdot)$ gives ϵ-differential privacy. (ϵ, δ)-differential privacy provides freedom to strict differential privacy for some low probability events. ϵ-differential privacy is usually called pure differential privacy, while (ϵ, δ)-differential privacy with $\delta > 0$ is called approximate differential privacy [1].

In Definition 3.2, the private parameter ϵ indicates the privacy budget [9] which gives strong privacy guarantees with a small ϵ. 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 3.3 (Sensitivity). *[8] For a query $Q : D \to S$, and two adjacent datasets D and D', the sensitivity ΔQ of query Q is defined as*

$$\Delta Q = max_{D,D'} \|Q(D) - Q(D')\|_1 \tag{9}$$

Sensitivity ΔQ is only related to the type of query Q. Intuitively, it considers the greatest impact on the query results after adding or deleting any record from the database.

There are two basic mechanisms meeting Definition 3.2, which are widely used currently to realize differential privacy. One is the Laplace mechanism [24] and the other is the exponential mechanism [23]. Laplacian mechanism uses the sensitivity of the Q as a parameter and adds Laplacian noise to the output of the query result. But for non-numeric queries, differential privacy uses an exponential mechanism for noise results because Laplace mechanism failed to solve this problem. We will use Laplacian mechanism in our FRP algorithm 4.6.

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. Recommender 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 buddy relationship attacking model.

3.3.1. 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 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 Furong lake of Xiamen University, her friends can immediately receive the check-in information. The potential attacker will find the her current position easily basing on the POI's information. Maybe it is even possible to know which pavilion she is sitting in around the lake. Hence the guarantee of the protected position is not enough now, and the POI which the user creates by himself/herself need to be blurred.

3.3.2. 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 tendencies, religious beliefs, and even sexual orientation and so on, from the user's preferred POIs. The existing POI recommendation system does not provide any effective way for privacy protection. Each user is a potential attacker, and they can rebuild the user's preferred POIs with their background knowledge by submitting right query [26]. The detailed are illustrated in the following two examples.

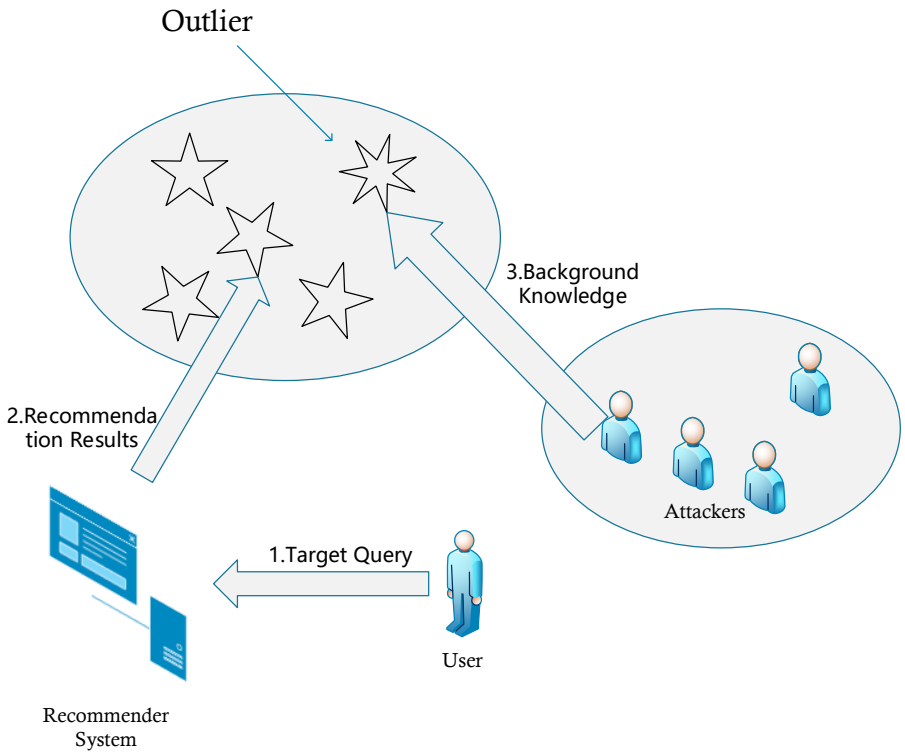


Fig. 3. Attacking mode of POI preference

Example 3.4. *[26] Assuming that user Bron has been using the recommender system service for some time, the system has also recorded the his POI preference. And Anna, a curious friend of Bron, often goes out with Bron together, such as eating dinner and signing in the restaurant, going to the movies and signing in the cinema and so on. And recently Anna learns that Bron likes to go to city C alone every Saturday night, but she does not receive the check-in information of Bron from the system. That is that maybe Bron selected the privacy check in city C. Private checking-in record will not be instantly shared to his friends. So Anna wants to know where Bron went every Saturday night. Anna submits a query to the system. As shown in Fig. 3, the query: "What are the POIs nearby?" is firstly sent to the system server by Anna. The returned result from recommender system 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 C. According to Anna's background knowledge, she can infer that "Hell bar" is the POI where Bron went every Saturday night with a high probability.*

In Example 3.4 and recommender system, Anna and Bron have higher user similarity $S^u_{A,B}$, and they are friends, so their friend relationship probability $S^s_{A,B}$ is also high. In the top-K results returned by the recommender system to Anna, some abnormal points appear. These abnormal points are far away from the current location of hers, i.e, the location factor value $S^g_{A,B}$ is very low. Anna understands the recommendation algorithm so she infers that this POI has the highest predicted check-in probability for her and such anomaly result is mainly because of her best friend Bron who has a similar preference. The following example also shows a probable attacking mode because of the similarity between users.

Example 3.5. *[26] It is still assumed that user Bron has used the recommender 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 in the POI_d Bron would like to choose a private signing. 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 indicate 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 high score of the three POIs publicly. For the above reasons, Anna can infer that POI_d is also Bron's preferred POI with a high probability.*

In Examples 3.4 and 3.5, we have assumed that the attacker knows the recommendation algorithm of the recommender system, but not yet the value of each parameter in the algorithm. These tell us two different attacking patterns basing 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 users location and friend ties in the previous sections so as to put forward 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 recommender framework as our novel idea to return recommendations enjoying privacy guarantees followed by strict theoretical analysis.

4.1. GLP: Geographical location privacy-preserving algorithm

In this part, we will propose the GLP algorithm to preserve location privacy and deal with the shortcomings of $\langle k, s\rangle$-privacy [20] as well for the geographical attacking model discussed in Section 3.3.

4.1.1. $\langle r, h\rangle$-Privacy

Our $\langle r, h\rangle$-privacy inherits the idea of $\langle k, s\rangle$-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. In such a low probability case, the attacker may also have a kind of violent way to find the user, but the $\langle r, h\rangle$-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 $\langle k, s\rangle$-privacy. If a user wants to send a position to the server, a fuzzy one will simultaneously be sent in our settings.

Definition 4.1 ($\langle r, h\rangle$-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 to anywhere in the circle, then the algorithm satisfies $\langle r, h\rangle$-privacy.*

Our $\langle r, h\rangle$-privacy algorithm shown in Algorithm 4.2 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.(10) (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 θ 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 4.2. *$\langle r, h\rangle$-Privacy algorithm*

Input: *Position $p_i(x_i, y_i)$ of POI_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.(10)*
3: ***if*** $h_i < 3$ ***then***
4: $\quad l \leftarrow random(r_i/2, r_i)$
5: ***else***
6: $\quad l \leftarrow random(0, r_i)$
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:

$$R(h_i) = -\frac{r_{max} - r_{min}}{h_{max}} \cdot h_i + r_{max} \tag{10}$$

where r_{max} and r_{min} represent the upper and lower bounds of the virtual circle radius respectively (their computations will be discussed later in Section 5.1.4), and h_{max} represents the maximum number of historical check-in numbers in all POIs ($h_{max} = max\{H(POI_1), H(POI_2), \ldots, H(POI_n)\}$). r_{min} is set to prevent some POI hotspots from losing the privacy guarantee since the virtual circle radius will be too small when the number of user checkouts increases.

We will also face some extreme situations when using $\langle r, h\rangle$-privacy. For example, 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 center o' of its virtual circle satisfies $dist(p, o') \in [r/2, r]$ to ensure that the two points are not too close, where r is the radius of the circle.

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

We provide a attractive theoretical analysis of GLP algorithm by using differential entropy [25] which supplies a quantitative indicator for the effect of blurring, i.e., the privacy guarantee.

Lemma 4.3 (Distance distribution). *In the xOy cartesian coordinate system, there is a point $H(h,0)$ on the x-axis and a circle with center O (the origin) and radius r $(0 \leqslant r \leqslant h)$. Let $P(x_0, y_0)$ be any point in the circle and z is the distance between H and P. Then the probability density function of z is $f_Z(z) = 2z/r^2$.*

Proof. Since the coordinates (X, Y) of point P in the circle follow the uniform distribution, their joint probability density function is $1/(\pi r^2)$. Then the distribution function of z is

$$\begin{aligned} F_Z(z) &= P\left\{\sqrt{(x_0-h)^2+y_0^2} \leqslant z\right\} \\ &= \frac{1}{\pi r^2} \iint_{\sqrt{(x_0-h)^2+y_0^2}\leqslant z} 1\, dx\, dy \\ &= \frac{1}{\pi r^2}\pi z^2 = \frac{z^2}{r^2} \end{aligned}$$

Hence,

$$f_Z(z) = d\, F_Z(z)/d\, z = d(z^2/r^2)/d\, z = 2z/r^2.$$

□

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 according to Eq.(5) in Section 2 when calculating the geographical location factor. We define the distance $z_{ij} = dist(POI_i, POI_j)$ between two private POIs as the distance between the center of one virtual circle and any point of the other virtual one so as to facilitate the calculation. Therefore, the expectation of this distance is

$$E(z_{ij}) = \int_{h-r}^{h+r} z_{ij} \cdot \frac{2z_{ij}}{r^2}\, dz_{ij} = \frac{12h^2+4r^2}{3r}$$

which will be used in Section 5. Since the uncertainty of the distance between POIs, the differential entropy [25] of the random variable $z_{i,j}$ can be calculated as

$$\begin{aligned} h(z) &= -\int_{h-r}^{h+r} \frac{2z}{r^2}\log(\frac{2z}{r^2})\, dz = \frac{2z^2\ln r + \frac{1}{2}z^2 - z^2\ln 2z}{r^2}\,|_{h-r}^{h+r} \\ &= \frac{(h+r)^2[2\ln r + \frac{1}{2} - \ln(2h+2r)] - (h-r)^2[2\ln r + \frac{1}{2} - \ln(2h-2r)]}{r^2} \qquad (11) \\ &> 0 \end{aligned}$$

We conclude this section in the following theorem.

Theorem 4.4 (Privacy Gain). *When adopting GLP algorithm to calculate the location factor weight, instead of the normal method in Eq.(6), we will get a reasonable information gain named privacy gain which is positive because of the distance $z_{i,j}$ between virtual circles.*

4.2. FRP: Friend relationship privacy-preserving algorithm

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

4.2.1. FRP algorithm

In the social networking recommender system, each user's friends are potential attackers. The attacker may infer the target user's privacy information with a high probability if the user is very close to his friends (i.e., the value of friend relationship is high) in the POI recommender system. Our FRP algorithm try 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 related private information. Therefore, the algorithm will prevent the potential attacks.

We choose ϵ-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 4.5 (Laplacian ϵ-differential privacy). *[33] Let $F : D \to R^k$ be a score function, its sensitivity is ΔF, then $F(D) + Lap^k(\Delta F/\epsilon)$ satisfies ϵ-differential privacy, where $Lap^k(\varphi)$ is a k-dimensional vector obtained from the Laplacian distribution with a standard variance of $\sqrt{2}\varphi$.*

The FRP algorithm is shown in Algorithm 4.6. We first give a function to compute friend similarity for any two users according Eq.(4)(lines 1-6). Second randomly delete one user u_i's one friend relationship link from the database F_i, i.e., they are no longer friends, and then generate a new neighbor database F_i' (lines 10-12). Then, calculate user u_i and all his friends u_k's friend relationship factors and save them in lists $SI_i(F_i)$ and $SI_i(F_i')$ basing on dataset F_i and F_i' respectively(lines 13-14). Hence, we can obtain the ∞-vector norm of the two friend relationship factor vectors, that is, the global sensitivity of the query function(line 16). Then we adopt the position parameter and scale parameter of Laplacian noise, where the former is 0 and the latter can be obtained from the sensitivity of the query function(lines 17-18). According to them, we can get the complete Laplace distribution and add Laplacian noise to the original friend relationship factor finally(line 19). We can see that the friend relationship factor become smoother and closer, and it will not exist the situation that the friend relationship weights of some users are particularly high from the experiment result after we implement FRP in the recommendation system.

Algorithm 4.6. *Friend relationship privacy-preserving algorithm*

Input: *the privacy parameter ϵ, Friend dataset F, POI dataset L, Buddy coefficient γ*

Output: $SI'_{i,k}$

1: ***function*** *FriendSimilarity(F,L,γ)*
2: ***if*** F_i, F_k, L_i, L_k *are not empty* ***then***
3: ***return*** $\gamma \cdot (F_i \bigcap F_k)/(F_i \bigcup F_k) + (1-\gamma) \cdot (L_i \bigcap L_k)/(L_i \bigcup L_k)$
4: ***else***
5: ***return*** 0
6: ***end if***

7: ***end function***
8: ***function*** *Social Differential Privacy(the privacy parameter ϵ)*
9: *initialize two list $SI_i(F_i), SI_i(F_i')$*
10: ***for*** *each u_i with each user u_k's $SI_{i,k}'$,* ***do***
11: *database F_i denotes all u_is relationship*
12: *F_i' denotes delete one of u_is relationship, $F_i' \leftarrow F_i - \{u_i, u_{random}\}$*
13: *$SI_i(F_i) \leftarrow append \quad FriendSimilarity(F, L, \gamma)$*
14: *$SI_i(F_i') \leftarrow append \quad FriendSimilarity(F', L, \gamma)$*
15: ***end for***
16: *sensitivity $\Delta f \leftarrow ||SI_i(F_i) - SI_i(F_i')||_\infty$*
17: *position factor $pf \leftarrow 0$*
18: *scale factor $sf \leftarrow \Delta f/\epsilon$*
19: *$\forall SI_{i,k}' \leftarrow SI_{i,k} + Laplace(pf, sf)$*
20: ***return*** *$SI_{i,k}'$*
21: ***end function***

4.2.2. Theoretical analysis of FRP

We still use the Shannon information entropy to evaluate the FRP algorithm because it mainly adopts wiping out the friend relationship weights with great differences for fuzzy implementation. There is a property that the bigger difference between users' friend relationship weights, 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 4.7 (Correctness). *Friend relationship factor with differential privacy increases the information entropy of the friend relationship factor distribution, and improves the uncertainty of the weight distribution, that is $H(SI_i') - H(SI_i) > 0$ protects the user information privacy.*

Proof. Let SI_i be the weight distribution between user u_i and other users. Let SI_i' be the privacy weight distribution of adding noise of Laplacian mechanism on SI_i. Then:

$$\begin{aligned} H(SI_i') - H(SI_i) &= H(SI_i + Lap(0, \frac{\Delta f}{\epsilon})) - H(SI_i) \\ &= H(Lap(0, \frac{\Delta f}{\epsilon})) \\ &= \log(\frac{2 \cdot \mathrm{e}}{\Delta f/\epsilon}) \\ &= 1 + \log(\frac{\mathrm{e} \cdot \epsilon}{\Delta f}) \end{aligned} \tag{12}$$

Then

$$2^{H(SI_i') - H(SI_i)} = 2^{1 + \log(\frac{\mathrm{e} \cdot \epsilon}{\Delta f})} = 2 \times \frac{\mathrm{e} \cdot \epsilon}{\Delta f}$$

where the result $4 \times \Delta f/\epsilon$ is the support set of the differential entropy, and it is always positive. Hence, $H(SI_i') - H(SI_i) > 0$. □

4.3. Private POI recommendation method

Combined with Algorithm and We say that it is a privacy recommendation one.

We employ shannon entropy to measure the degree of privacy guarantee.

We calculate the entropy of an individual record released by a database as

$$H'(X_i) = -\sum_{x=1}^{k} p'_{ix} lg(p'_{ix}) \tag{13}$$

where $H'(X_i)$ is the entropy of the ith record after a statistical release in the database and $H(X_i)$ is associated with the ith record, i.e., $H(X_i) - H'(X_i)$ is the amount of information leakage of the record r_i in the database. Similarly, we can transfer this method to calculate information loss of every record in the database. From this we can define privacy disclosure:

Definition 4.8 (Privacy disclosure). *[4] Let $H(X_1), H(X_2), \cdots, H(X_i)$ be the entropies of each record in the database and $H'(X_1), H'(X_2), \cdots, H'(X_i)$ be the entropies of each record after a statistical release. Therefore, privacy disclosure can be defined as the maximum information loss of a record in the database. It is*

$$PrivacyDisclosure(P) = \max_{i \leq n}(H(X_i) - H'(X_i)) \tag{14}$$

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

Lemma 4.9 (Functional entropy-chain rule). *Let X, Y, Z be three discrete random variables which are mutually independent and a one-to-one corresponding function $M(X, Y, Z) = X + Y + Z$. There exists $H(M(X, Y, Z)) = H(X) + H(Y) + H(Z)$.*

Proof. The random variables X, Y, Z are independent of each other so there exists $H(X, Y, Z) = H(X) + H(Y|X) + H(Z|X, Y) = H(X) + H(Y) + H(Z)$ according to [12]. At the same time, $M(X, Y, Z)$ a one-to-one corresponding function with X, Y, Z, so there are

$$\begin{aligned} H((X, Y, Z), M(X, Y, Z)) &= H(X, Y, Z) + H(M(X, Y, Z)|(X, Y, Z)) \\ &= H(X, Y, Z) + 0 \\ &= H(X, Y, Z) \end{aligned}$$

and

$$\begin{aligned} H(M(X, Y, Z), (X, Y, Z)) &= H(M(X, Y, Z)) + H((X, Y, Z)|M(X, Y, Z)) \\ &\geq H(M(X, Y, Z)) \end{aligned}$$

where $H((X, Y, Z)|M(X, Y, Z)) \geq 0$. And $H((X, Y, Z)|M(X, Y, Z)) = 0$ iff there is a one-to-one correspondence between the random variables (X, Y, Z) and their function $M(X, Y, Z)$. Hence,

$$H(M(X, Y, Z)) = H(X, Y, Z) = H(X) + H(Y) + H(Z) \tag{15}$$

□

Theorem 4.10 (Balance formula). *Under the assumption that the three parameters are pairwise independent, the entropy of the POI probability distribution obtained by the private POI recommendation algorithm is greater than the normal recommendation algorithm.*

Proof.

It is proved that the information entropy of location and friend relationship increased after adding privacy-preserving algorithm in Theorems 4.4 and 4.7, where $H(z) = H(2z/R^2) > 0$ and $H(SI'_i) - H(SI_i) > 0$, z represents the probability distribution of the distance between a POI to another fuzzy circle of a POI, SI represents the probability distribution of the friend relationship factor between the user and all other users. Based on Eq.(11), Eq.(12) and Lemma 4.9, the balance formula is given by

$$\begin{aligned}
\Delta H &= H(S'_{i,j}) - H(S_{i,j}) \\
&= H(S^u_{i,j} + S'^s_{i,j} + S'^g_{i,j}) - H(S^u_{i,j} + S^s_{i,j} + S^g_{i,j}) \\
&= H(S^u_{i,j}) - H(S^u_{i,j}) + H(S'^s_{i,j}) - H(S^s_{i,j}) + H(S'^g_{i,j}) - H(S^g_{i,j}) \\
&= H(S'^s_{i,j} + S'^g_{i,j}) - H(S^s_{i,j} + S^g_{i,j}) \\
&= H(\frac{2z}{R^2}) + H(Lap(\frac{\Delta f}{\epsilon})) \\
&> 0
\end{aligned} \tag{16}$$

□

In the Eq.(16), ΔH is the incremental information entropy after adding the privacy-preserving algorithm, and it represents the degree of privacy guarantee. And differential privacy parameter ϵ which is from the friend relationship factor and the query function sensitivity Δf can be set by users before to ensure a reasonable privacy guarantee. On the location factor, in addition to the system default radius of the virtual circle, users can also customize the radius, since the parameter R is also a reasonable control of privacy guarantee for them.

5. Empirical Study

In this section, we will test all the proposed algorithms and evaluate the performance of these methods by examining the quality of them with different metrics.

5.1. Experiment setup

5.1.1. Dataset description

We conduct experimental studies using data crawled from Foursquare which is one of the most representative location-based services sites. The data 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. 4, after summarizing the check-in records, the user and POI check-in matrix is generated and its density is only about 2.41×10^{-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 $x\%$ ($x = 10, 30, 50,$

UserID	POIID	Position(latitude,longitude)	Time	DateID
USER_1675	LOC_1967	1.3095228064610511,103.90178203582764	0:48	0
USER_1544	LOC_2505	1.3429556294180167,103.77525687217712	2:16	0
USER_855	LOC_3369	1.2952893191369519,103.82989883422852	18:55	14
USER_855	LOC_2909	1.2914708266088921,103.84985983371735	17:57	13
USER_2103	LOC_4944	1.3549042405307776,103.83102536201477	18:27	20
USER_2189	LOC_4633	1.4432620341955018,103.78509521484375	9:21	28
USER_2186	LOC_2614	1.3561698888933271,103.98703336715698	11:25	30
USER_2186	LOC_4424	1.3011779381831814,103.83841753005981	12:32	12

(a) Check-in Data from Foursquare

User1ID	User2ID
6	1961
13	377
14	86
14	575
18	364
18	1956
20	341
22	80

(b) Friendties

Fig. 4. Dataset from Foursquare

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 quality of recommended POIs, we use two different metrics, namely, recall@K and precision@K, where 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

$$Recall@K = \frac{|A \cap B|}{|A|}$$

$$Precision@K = \frac{|A \cap B|}{|B|}$$

$$F = \frac{2}{\frac{1}{R} + \frac{1}{P}} = \frac{2RP}{R+P}$$

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

5.1.3. 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.(7). Here, we employ our private algorithms 4.2 and 4.6 to calculate $S^s_{i,j}$ and $S^g_{i,j}$ respectively. More importantly, we need to determine their weights, that is, the values of α and β. Let $K = 5$, we use various combinations of α and β to test precision and recall of POI recommendation method, and we choose the values when the best performance is achieved, see Section 5.2.1. We also use the obtained values of α and β for other top-K recommendation.

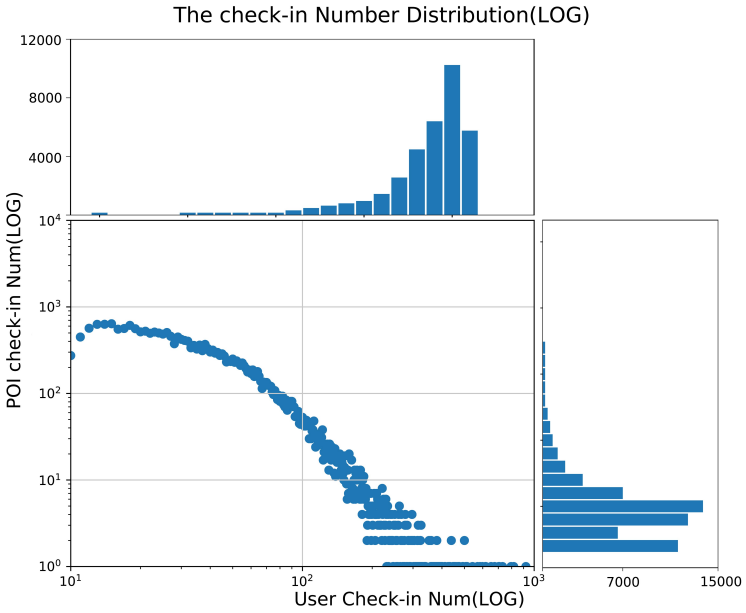


Fig. 5. The check-in number of POI (logarithmic)

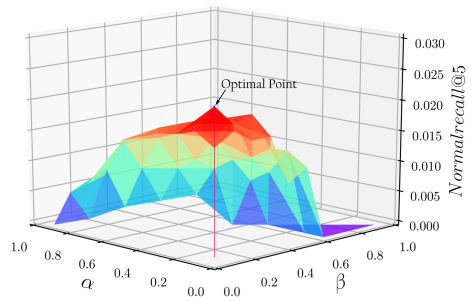
5.1.4. Determining the radius in the GLP algorithm

There are several parameters not determined yet in Eq.(10) for obtaining the radius of the virtual circle in our GLP algorithm. As shown in logarithmic Fig. 5, 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 10,000, and overview all the POI historical records, only a small amount of POI can reach 10,000 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 let $h_{max} = 100$ and for 43593 POIs, almost 500 of the hottest POIs' virtual radius will achieve the minimum r_{min}, on the contrary, for those unsigned POIs but their creators, i.e., their history check-in number are 1 and they have a amount of more than 7000, so 18% of POIs' virtual radius will achieve the maximum r_{max}. The empirical value of r_{max} is 200 (meter) so that the maximum area of the virtual circle is $S_{circle} = \pi \cdot r_{max}^2 \approx 12500m^2$, only the 1/8 of maximum private area in $\langle k, s \rangle$-privacy algorithm.

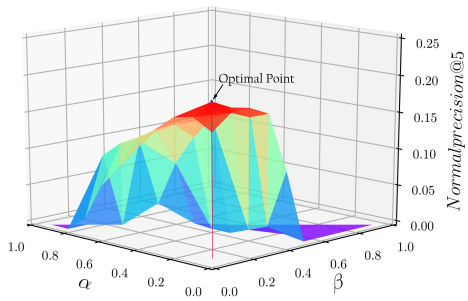
5.2. Experimental results

5.2.1. Weights of POI factors when $K = 5$

In this part, we conduct experiments on normal algorithm and privacy algorithm to determine the weights of the three POI factors when $K = 5$, respectively. In Fig. 6, the best performance of normal algorithm is achieved when α and β 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



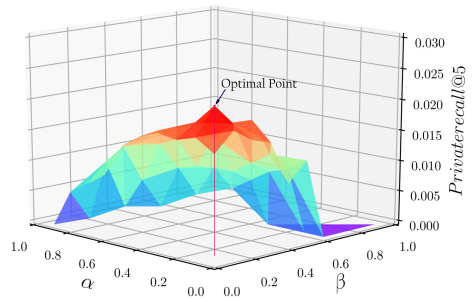
(a) Recall@5 of normal algorithm



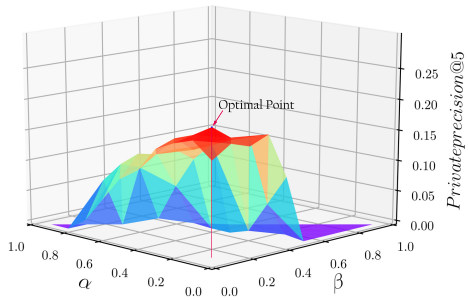
(b) Precision@5 of normal algorithm

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

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 location factor has the minimum effect on the performance of recommendation among all the factor combinations. The contribution of the location factor is less than the others. Note that if we only adopt the user similarity, the performance of recommendation will much close to the maximum.



(a) Recall@5 of privacy algorithm



(b) Precision@5 of privacy algorithm

Fig. 7. Recall@5 and Precision@5 of private algorithm

As shown in Fig. 7, in the adjustment of the linear parameters of the private version, it can be seen that the private algorithm does not reduce the validity of the proposed algorithm (???) significantly, and both α and β are exactly equal to 0.1 when recall rate and accuracy rate get the optimal point. And even if the privacy algorithm does not have a very large

impact on the accuracy of recommendation algorithm in the extreme situation, we also conclude that the weight of friend relationship is greater than that of location.

5.2.2. Effect of varying K

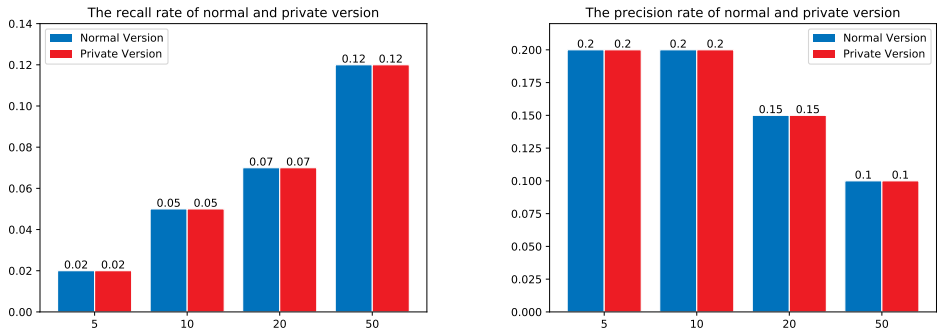


Fig. 8. Recall@K and Precision@K of normal and private version

We will use the optimal linear parameters $\alpha = 0.1$ and $\beta = 0.1$ for the following experiments. The recall rate $@K$ and the precision rate $@K$ (where $K = 5, 10, 20, 50$) of the recommendation algorithm for the normal and private versions are shown in Fig. 8. No matter the value K is, the performance of the normal version and the private version recommendation algorithm is roughly the same, and the private version 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 about 0.17 under the sparseness of 7.8×10^{-4} in [17]. Thus, the precision rate 0.2 in our experiment is reasonable enough with a matrix sparseness of 2.41×10^{-3}.

5.2.3. Effect of varying ϵ

Evaluation of the privacy recommendation algorithm (Algorithm ???) needs 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 privacy algorithm is as good as the normal one.

When the privacy parameter $\epsilon = 0$, the recommendation algorithm is a normal one. The smaller ϵ, the greater noise added in the factor of friend relationship, and the higher degree of privacy protection. The experimental results are shown in Fig. 9. All the POIs 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 ϵ ($\epsilon = 0.1, 0.3, 0.5, 0.7, 0.9$), when ϵ 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, when $\epsilon = 0.5$, the information entropy has dropped rapidly to 1/5 of that when it is 0.1. However, the information entropy is still more than double of the normal version, so we conclude that the guarantee of privacy of the private algorithm has been greatly improved compared to the normal one. Fig. 9 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 $K = 10$.

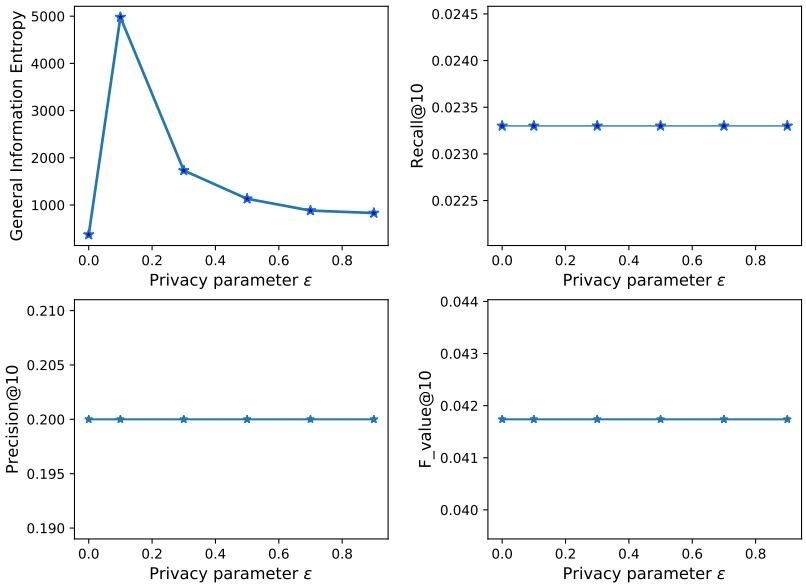


Fig. 9. Performances under different privacy parameters

6. Conclusions and Future Work

We conclude that exploiting differential privacy for a POI recommendation system is significative and feasible without taking visible hit in the recommendations accuracy. Our idea is to incorporate user interest, fusion social ties and fusion geographical position in the recommendation. We propose two privacy-preserving algorithms independently, namely $\langle r, h\rangle$-privacy and friend relationship privacy-preserving algorithms, and provide multiple proof and evaluation methods for demonstrating their utilities. We conduct a comprehensive performance evaluation over a large-scale real dataset collected from Foursquare. In our experiments we tuned enough parameters that had a potential to vary freely, and it is natural to expect that further experimentation could lead to dramatically improved prediction accuracy. The chosen smoothing weights and distribution of 'accuracy' ϵ between the calculations could be fixed and possibly improved.

Directions for future work include efficient methods for the generalization of privacy algorithms which are need to be defined further abstractly because of the new framework of the recommender algorithm and privacy-preserving algorithm designed at a more abstract level.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grants No. 61772442, 61402306 and 11671335).

References

[1] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 363–378. Springer, 2013.

[2] Shlomo Berkovsky, Yaniv Eytani, Tsvi Kuflik, and Francesco Ricci. Enhancing privacy and preserving accuracy of a distributed collaborative filtering. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 9–16. ACM, 2007.

[3] John Canny. Collaborative filtering with privacy. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 45–57. IEEE, 2002.

[4] Vinod Chirayath, Luc Longpré, and Vladik Kreinovich. Measuring privacy loss in statistical databases. 2006.

[5] Tore Dalenius. Towards a methodology for statistical disclosure control. *statistik Tidskrift*, 15(429-444):2–1, 1977.

[6] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[7] Cynthia Dwork. Differential privacy in new settings. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 174–183. SIAM, 2010.

[8] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.

[9] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[10] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[11] Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.

[12] Robert M Gray. *Entropy and information theory*. Springer Science & Business Media, 2011.

[13] Shuguo Han, Wee Keong Ng, and S Yu Philip. Privacy-preserving singular value decomposition. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 1267–1270. IEEE, 2009.

[14] Thomas Hofmann and D Hartmann. Collaborative filtering with privacy via factor analysis. In *Proceedings of the 2005 ACM symposium on applied computing*, pages 791–795, 2005.

[15] Cihan Kaleli and Hüseyin Polat. P2p collaborative filtering with privacy. *Turkish Journal of Electrical Engineering & Computer Sciences*, 18(1):101–116, 2010.

[16] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[17] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 195–202. ACM, 2009.

[18] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 195–202. ACM, 2009.

[19] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007.

[20] Hua Lu, Christian S Jensen, and Man Lung Yiu. Pad: privacy-area aware, dummy-based location privacy in mobile services. In *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 16–23. ACM, 2008.

[21] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 24–24. IEEE, 2006.

[22] Ashwin Machanavajjhala, Aleksandra Korolova, and Atish Das Sarma. Personalized social recommendations: accurate or private. *Proceedings of the VLDB Endowment*, 4(7):440–450, 2011.

[23] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007.

[24] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30. ACM, 2009.

[25] Fazlollah M Reza. *An introduction to information theory*. Courier Corporation, 1961.

[26] Daniele Riboni and Claudio Bettini. Private context-aware recommendation of points of interest: An initial investigation. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 584–589. IEEE, 2012.

[27] Anand D Sarwate and Kamalika Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE signal processing magazine*, 30(5):86–94, 2013.

[28] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[29] Christine Task and Chris Clifton. A guide to differential privacy theory in social network analysis. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 411–417. IEEE Computer Society, 2012.

[30] Raymond Chi-Wing Wong, Jiuyong Li, Ada Wai-Chee Fu, and Ke Wang. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 754–759. ACM, 2006.

[31] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 325–334. ACM, 2011.

[32] Elena Zheleva and Lise Getoor. Privacy in social networks: A survey. In *Social network data analytics*, pages 277–306. Springer, 2011.

[33] Tianqing Zhu, Gang Li, Wanlei Zhou, and S Yu Philip. Differentially private data publishing and analysis: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1619–1638, 2017.