Journal of Ambient Intelligence and Humanized Computing (2023) 14:8019–8033
https://doi.org/10.1007/s12652-021-03690-z

ORIGINAL RESEARCH

# Differentially private publication for related POI discovery

**Ximu Zeng[1] · Xue Chen[1] · Xiao Peng[1] · Xiaoshan Zhang[1] · Hao Wang[1] · Zhengquan Xu[2]**

Received: 19 March 2021 / Accepted: 21 December 2021 / Published online: 10 January 2022


**Abstract**
Among the advanced methods, differential privacy (DP), introducing independent Laplace noise, has become an influential privacy mechanism owing to its provable and rigorous privacy guarantee. Nonetheless, in practice, POI data to be protected is always correlated, while independent noise may cause undesirable information disclosure than expected. Recent researches attempt to optimize the sensitivity function of DP with consideration of the correlation strength between POI—but there is a drawback in a substantial growth of noise level. To remedy this problem, this paper exploits the degradation of DP in expected privacy levels for correlated POI data and proposes a solution to mitigate it. We propose a generalized Laplace mechanism to achieve privacy guarantees. Specifically, we design a practical iteration mechanism, including an update function, to conduct a generalized Laplace mechanism when facing large scale queries. Experimental evaluation on real-world datasets over multiple fields show that our solution consistently outperforms state-of-the-art mechanisms in data utility while providing the same privacy guarantee as other approaches for correlated POI data.

**Keywords** POI data · Correlated POI · Differential privacy · Privacy preserving

## 1 Introduction

A point of interest (POI) is either a tourist attraction or a landmark location that is used in an electronic map to indicate interesting locations (Xi et al. 2020; Zhu et al. 2018; Pouke et al. 2016), such as tourist attractions (historical locations, natural landscapes, etc.), public conveniences (parks, public toilets etc.), and public service departments (offices and receptions, etc.). Information obtained from POI data can be used to support product recommendations, advertisements, and navigation.

However, if users' POI behavior privacy are not protected, their privacy could be compromised (Cai et al. 2021), including information about personal interests and geographic location. For example, the check-in data of scenic spots can directly represent user's hobbies and behaviors. But, the problem of privacy leakage caused by POI correlation has not been addressed well in the state-of-the-art work. Therefore, this study aims to provide a method to protect the user's POI behavior privacy data and information.

Recently, differential privacy (DP) (Dwork et al. 2014; Dwork 2006) has become a mainstream privacy preserving method. DP realizes the transition from traditional passive privacy preserving which relies on the security of algorithm to an active preserving method based on probability and statistics. Due to its mathematical security and provability and better data availability, once proposed, it has been widely used in the fields of computer science, economics, bioinformatics, medicine, etc. Plenty of researchers are trying to use differential privacy theory to solve the problem of privacy leakage in POI protection.

Existing differential privacy mechanisms add independent and identically distributed (IID) Laplacian noise to the output count values of the POI. The output is randomly assigned to the third party analytic agencies, to prevent them from identifying the count value and to protect the interest privacy. However, owing to the correlation among the check-in data, merely adding IID noise might leak user's private information. A visitor who has visited one of two highly correlated POIs is likely to visit the other one. Here we give

✉ Hao Wang
haowang@cqupt.edu.cn

[1] Key Laboratory of Tourism Multisource Data Perception and Decision, Ministry of Culture and Tourism, College of Computer Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[2] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Collaborative Innovation Center for Geospatial Technology, Wuhan University, Wuhan 430079, China

**Table 1** Check-in statistics of some POIs in a tourist attraction

| User_ID | Spot_8 | Spot_13 | Spot_5 | Spot_7 | Spot_20 | Spot_39 |
|---|---|---|---|---|---|---|
| 8 | 1 | 0 | 1 | 1 | 0 | 1 |
| 9 | 1 | 1 | 1 | 0 | 0 | 1 |
| 10 | 0 | 1 | 1 | 0 | 1 | 1 |
| 11 | 1 | 1 | 1 | 1 | 0 | 1 |
| 12 | 0 | 1 | 1 | 1 | 0 | 1 |
| 13 | 0 | 1 | 1 | 1 | 1 | 1 |
| 14 | 0 | 0 | 0 | 0 | 1 | 1 |
| Count | 3 | 5 | 6 | 4 | 3 | 7 |

an example to illustrate this issue, as shown in Table 1 and Example 1.

***Example 1*** Table 1 summarizes the check-in statistics of a few users in a tourist attraction. Visitor check-in frequency data from a POI (Table 1) is submitted to a third party analytic agency to gauge the number of visitors and their visiting trends. Such information could be used to understand the popularity of the tourist attraction and tourists' travel preferences, and to improve the facilities at the attractions. Although the third-party analytics agencies would not have access to the secure database of the POIs, the information of a specific visitor at the POI, can be obtained by statistically analyzing this data. For example, by analyzing the check-in information (Table 1) that the users 8–13 visited Spot_20 two times, while users 8–14 visited three times, a third party analytics agency can calculate the difference and decipher that the user 14 is more interested in Spot_20.

The above example shows that protecting POI information requires the study of differential privacy protection algorithms that are applicable to relevant POI data. While current differential privacy methods face the following two challenges:

- Existing differential privacy mechanisms prefer to protect data privacy by adding IID noise in POI data. However, this causes the privacy protection intensity to be lower than the setting value, which is also the key problem of DP when it is applied to protect correlated data.
- Some researchers consider increasing the scale of noise to offset the effect of destroying data availability. However, such an approach results in a significant reduction in the quality of recommendations.

In response to the first problem, this paper endeavor to avoid increasing the noise intensity. Therefore, this paper applies related noise, instead of IID noise, to protect POI's privacy. To address the second problem, this paper needs to solve how to express noise correlation. This paper regards the connection and change between POIs as a Bayesian network, so that the correlation between POIs can be calculated by the transition probability between different scenic spots. Considering that POIs are typical tuple data, this paper uses an autocovariance matrix to express their correlation, and then generates noise with the same correlation as the POIs data. Since the correlation of noise is the same as that of POI tuple data, there is no need for more noise to achieve the same privacy degree as IID noise case.

In order to protect the privacy of correlated POI, this paper proposes a correlation calculation method, in which the related Laplacian random variables are generated by combining the exponential distribution and the Gaussian distribution. Our contributions can be summarized as follows:

- We propose an idea that the protection requirements of DP can be met without increasing the noise scale by generating Laplacian noise whose correlation is consistent with POI. This provides a promising perspective for differentially private correlated POI data protection.
- We propose a generalized mechanism in Wang and Wang (2021). In this paper, we design a specific Laplacian mechanism to generate noise variables consistent with the correlation of the POI. It can also support repeated POI release with high correlation by using iteration and update mechanisms to update the noise.
- In order to evaluate the performance of the proposed mechanism on POI protection, this paper analyzes privacy degree and the utility loss theoretically. We also carry out experiments on real-life datasets. Theoretical analysis and experimental evaluation demonstrates that our solution is better than existing algorithms, which verifies the effectiveness of our solution.

The organization of this paper is as follows. We first describe related work in Sect. 2. Then we introduce the related notations in our work and demonstrate the challenges of current schemes in Sect. 3. Section 4 describes our methodology. Finally, Sect. 5 evaluates the performance of our solution and Sect. 6 concludes our work.

## 2 Related work

When users upload their POIs, their geographical location, hobbies and other private information are revealed. A Space-Twist solution (STS) (Yiu et al. 2011) was proposed to protect user's private information in POI applications. However, a trusted third-party platform needs to be introduced for the STS. Here, the location service request received from the service provider (SP), is transferred to a trusted third party by location service provider (LSP). But, LSP does not directly upload the user's real location to the SP. In this way, the third-party analytics platform sends fake location information to the SP, to protect the privacy of user's location. Moreover, the introduction of a private information retrieval (PIR) scheme (Yadav et al. 2020), within the LBS system, was proposed to increase its security. An anonymous interval algorithm based on an anonymous method was proposed to protect user's privacy. This algorithm, based on quadtree structure, recursively divides the geographical area into four squares of equal area until the user's minimum area requirement is satisfied. Upon a user's location query, the point information in this area will be uploaded randomly at a certain time, thereby hiding the real location information of the user. Since most of POI data-based recommended systems (Lu et al. 2019) rely on sensitive information of the users, several techniques to integrate privacy protection methods into the recommendation systems were proposed. Ren et al. (2021b) proposed a practical homomorphic encryption scheme that can effectively protect the privacy of key data. Xu et al. (2020) studied adversarial robustness through randomized perturbations. Gambs et al. (2007) employed secure multiparty computing to prevent sensitive data disclosure to untrusted recommendation systems, similar to other techniques. However, this was unable to prevent background knowledge attacks. Therefore, the differential privacy has been used for efficient privacy protection against background knowledge attacks.

Eltarjaman et al. (2016) proposed the private top-k method to protect individual's POI privacy. McSherry and Talwar (2007) indicated that several existing recommendation technologies can apply differential privacy technology, without a significant reduction in the quality of the recommendations. Our technique is based on POI behavior's similarity; users receive recommendations from the other users having the same POI preferences. Although the method proposed by McSherry and Talwar (2007) provided conventional protection against background knowledge attacks, this paper considers that it is not suitable for POI discovery because it did not consider the relevance and streams publishing-characteristics of POI data. For example, a visitor who has visited one of the two highly correlated POIs is likely to visit another POI. However, the existing methods do not consider this. Therefore, we focus on this lacuna in this paper.

## 3 Preliminaries

### 3.1 Autocovariance matrix

Since POI has a typical tuple data structure, its correlation can be represented by a correlation matrix, which can be constructed using either a covariance matrix or a Pearson correlation coefficient. The correlation in this paper refers to the correlations between tuple data. Correspondingly, either the covariance matrix or the Pearson coefficient refers to either the autocovariance matrix or the Pearson correlation coefficient of the tuple data, respectively. Here, the autocovariance matrix is used to represent the correlation of the tuple, which is defined as follows:

**Definition 1** *(Autocovariance matrix)* The autocovariance $C(x_i, x_j)$ of any of the two elements, $x_i$ and $x_j$ in the tuple dataset $X$ is defined as:

$$C_{x_i,x_j} = E\left[\left(x_i - \mu\right)\left(x_j - \mu\right)\right]. \tag{1}$$

The Matrix

$$\mathbf{C} = \begin{bmatrix} C_{x_1,x_1} & C_{x_1,x_2} & \cdots & C_{x_1,x_n} \\ C_{x_2,x_1} & C_{x_2,x_2} & \cdots & C_{x_2,x_n} \\ \vdots & \vdots & \vdots & \vdots \\ C_{x_n,x_1} & C_{x_n,x_2} & \cdots & C_{x_n,x_n} \end{bmatrix} \tag{2}$$

represents the autocovariance matrix of the tuple dataset $X$, where $x_i, x_j \in X$, and $\mu$ is the mean of the elements in $X$.

### 3.2 Differential privacy

DP is a state-of-the-art privacy preservation model which can guarantee the security of indistinguishability. Essentially, it is a noisy perturbation privacy preserving mechanism. By adding perturbation to raw data or statistical results, DP can guarantee that changing a single record's value has minimal effect on the output results. Thus, DP can preserve the privacy of data to be protected, while supporting mining results well. Definition 1 is its formalized form.

**Definition 1** ($\epsilon$-DP[5]) Considering two adjacent datasets, $D$ and $D^{'}$, which have the same admeasurement but differ in one record to be protected. If the random perturbation mechanism $M$ makes every set of results $S$ satisfy the following equation on $D$ and $D^{'}$, then $M$ satisfies $\epsilon$-DP.

$$Pr[M(D) \in S] \leq e^{\epsilon} \times Pr[M(D^{'}) \in S], \tag{3}$$

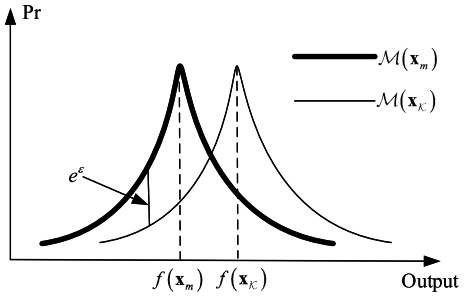


**Fig. 1** Probability density function of random algorithm $M$ on the statistical output of $D$ and $D^{'}$

where $S \subseteq Range(M)$, $Range(M)$ is the value range of random algorithm $M$. $Pr[\cdot]$ indicates probability density function (PDF) and $\epsilon$ represents privacy budget parameter.

A smaller $\epsilon$ is related with high-level privacy. Figure 1 shows the probability density function of random algorithm $M$ on the statistical output of $D$ and $D^{'}$.

Privacy budget $\epsilon$ is mainly limited by random algorithm $M$. In fact, Laplace mechanism is usually used to realize $M$. The Laplace mechanism is defined as follows.

**Definition 2** (Laplacian Mechanism (McSherry and Talwar 2007; Schillings et al. 2020; Wang et al. 2018b)) Let $f(\cdot)$ be the statistical function of the output result. The noisy samples $Z \sim Lap(\lambda)$ obeying Laplacian distribution can ensure the random perturbed result $M(D) = f(D) + Z$ satisfy $\epsilon$-DP, where $\lambda$ is the scale of Laplacian distribution. The Laplacian distribution is formalized by the following formula

$$\rho(z) = \frac{1}{2\lambda} exp\left(-\frac{|z|}{\lambda}\right). \tag{4}$$

The scaling parameter $\lambda$ is decided by the sensitivity function $\Delta f$ and privacy protection intensity $\epsilon$:

$$\lambda = \frac{\Delta f}{\epsilon}, \tag{5}$$

where $\Delta f$ is the largest effect of a single record on the statistical results.

$$\Delta f = \max_{D^{'}} \|f(D) - f(D^{'})\|_1. \tag{6}$$

For example, consider a dataset whose sensitivity is 1. Based on the concept of DP, the noise (added to the real answer) distributed according to $Lap(1/\epsilon)$ is enough to guarantee $\epsilon$ -DP.

## 4 Methodology

The correlation calculation method of related POI data is provided, followed by the design of the generalized Laplacian mechanism that is applicable to the relevant POI data. Further, the noise required by the generalized Laplacian mechanism is generated through an iterative mechanism.

In this section, the paper proposes a method to generate Laplace noise with a specific correlation matrix, which is calculated by the POI data. Firstly, the correlation matrix of the POI data is calculated in Sect. 4.1. Secondly, the paper shows the form of the noise distribution and formalizes it as Definition 5 in Sect. 4.2. Thirdly, Sect. 4.3 is the designed practical noise generation mechanism, which generates the binary Laplace noise with a specific correlation. Finally, Sect. 4.4 gives the practical algorithm to generate needed Laplace noise and Sect. 4.5 is the time complexity analysis.

### 4.1 POI correlation

Although the POI data is a tuple type, a correlated representation of the POI data is required. Owing to the connections between different users when they visit neighboring attractions, the connections and changes between POIs are regarded as Bayesian networks. The correlation between the POIs can be calculated from the probability of transition between different attractions. The POI and POI check-in datasets are formally defined as the following.

**Definition 3** (POI) $p_i$ is a semantic geographic object of an abstract geographic location, such as schools, banks, restaurants, and other places of interest.

**Definition 4** (Check-in POI dataset) Dataset of check-in information. An user $U_i$ visited a POI $p_i$, and checked in at $p_i$. This is recorded either as 0 or 1, wherein 1 indicates user's interest in this POI and 0, the opposite. The POI dataset for all the users constitutes a numerical sequence, denoted as $X = \{x_1, \ldots, x_i, \ldots, x_n\}$, where $x_i$ refers to the number of times that all users visited this POI $p_i$.

In order to describe the relationship between different POIs, we use a graph model (Fig. 2). The nodes $p_1$, $p_2$, and $p_3$ represent the three attractions. Assuming that there is only one path from $p_1$ to $p_2$, the transition probability from to is denoted as $Pr(p_2|p_1)$. Similarly, the transition probability from $p_1$ to $p_3$ is denoted as $Pr(p_3|p_1)$, while the transition

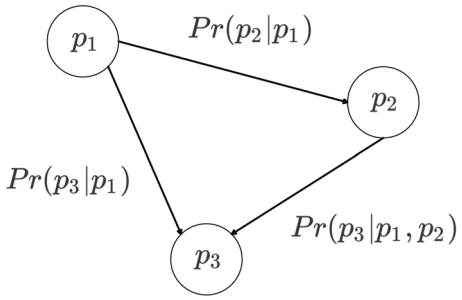


**Fig. 2** Model diagram of probability of transitions between different attractions

probability for $p_1$ to $p_3$ via $p_2$ is denoted as $Pr(p_3|p_1,p_2)$. The transition probabilities represent the correlations between different POIs. For example, for $Pr(p_2|p_1)$, the probability $Pr(p_1,p_2)$ of visiting both attractions, $p_1$ and $p_2$, is calculated, followed by the probability of visiting only attraction $p_1$ ($Pr(p_1)$). The transition probability from $p_1$ to $p_2$ is obtained using the equation, $Pr(p_2|p_1) = Pr(p_1,p_2)/Pr(p_1)$. To apply the relevant tuple differential privacy protection algorithm, proposed in Sect. 4.4, the covariance between different POIs is calculated.

Extending the three nodes in Fig. 2 to $n$ nodes and assuming that the joint probability distribution of $n$ nodes is $Pr(p_1,p_2,\ldots,p_n)$, the joint probability distribution can be written as a product of conditional probabilities, based on Bayesian criterion:

$$\begin{aligned} Pr(p_1,p_2,\ldots,p_n) &= Pr(p_n|p_1,\ldots,p_{n-1}) \\ &\cdots Pr(p_2|p_1)\cdot Pr(p_1). \end{aligned} \tag{7}$$

For a given $n$, the joint probability distribution can be represented as a directed graph with $n$ nodes, with each node corresponding to a certain conditional probability distribution on the right side of the equation (7).

In this paper, we give the hypothesis that the numbers of visitors on different POIs obey Gaussian distribution according to the theorem of large numbers. There are popular and unpopular scenic spots. In the previous research, we investigate this phenomenon from the view of statistical theory and gathered statistics of the numbers of visitors on different POIs. According to the theorem of large numbers, we find out that POI data should be a Gaussian distribution (Wang et al. 2018b).

Since the check-in POI dataset ($X = \{x_1,\ldots,x_i,\ldots,x_n\}$) approximates the Gaussian distribution, the node $p_i$ can be regarded as a random variable obeying the Gaussian distribution. Considering the arbitrary directed acyclic graph, composed of $n$ variables, the conditional probability of the node $p_i$ would be a linear combination of the states of its parent nodes $pa_i$:

$$Pr(p_i|pa_i) = N\left(p_i|\sum_{j\in pa_i} w_{ij}p_j + b_i, v_i\right). \tag{8}$$

where, $w_{ij}$ and $b_i$ are the parameters that control the mean and $v_i$ is the variance of the conditional probability. In the above representation of the linear combination, the natural logarithm of the joint probability distribution equals the natural logarithm of the product of the node of conditional distribution in the directed graph:

$$\begin{aligned} \ln Pr(\mathbf{P}) &= \sum_{i=1}^{n} \ln Pr(p_i|pa_i) \\ &= -\sum_{i=1}^{n} \frac{1}{2v_i}\left(p_i - \sum_{j\in pa_i} w_{ij}p_j - b_i\right)^2 + B. \end{aligned} \tag{9}$$

where $\mathbf{P} = (p_1,\ldots,p_n)'$, $B$ represents a constant term that is unrelated to $\mathbf{P}$. Equation (9) can be treated as a quadratic function of $\mathbf{P}$, and the joint probability distribution $Pr(\mathbf{P})$ as a multivariate Gaussian distribution variable.

The mean and variance of the joint probability distribution can be obtained by a recursive method. Since the variable $p_i$ is a conditional probability distribution of the state of the parent node, there is

$$p_i = \sum_{j\in pa_i} w_{ij}p_j + b_i + \sqrt{v_i}\varphi_i, \tag{10}$$

where $\varphi_i$ is a Gaussian random variable, $E[\varphi_i] = 0$, $E[\varphi_i\varphi_j] = I_{ij}$, and $I_{ij}$ is the $i-th$ and $j-th$ elements of the identity matrix. Therefore, equation (10) leads to the following:

$$E[p_i] = \sum_{j\in pa_i} w_{ij}E[p_j] + b_i, \tag{11}$$

Starting from a node with the lowest sequence number and recursively calculating along the graph, each element of $E[\mathbf{P}] = (E[p_1],\ldots,E[p_n])'$ can be obtained. Similarly, combining the equations (10) and (11), the $i-th$ and $j-th$ elements of the covariance matrix $Pr(\mathbf{P})$ can be calculated by the recursive method:

$$\begin{aligned} \mathbf{C}_{p_i,p_j} &= E\left[(p_i - E[p_i])(p_j - E[p_j])\right] \\ &= E\left[(p_i - E[p_i])\left\{\sum_{k\in pa_i} w_{jk}(p_k - E[p_k]) + \sqrt{v_j}\varphi_j\right\}\right] \\ &= \sum_{k\in pa_i} w_{jk}\mathbf{C}_{p_i,p_k} + I_{ij}v_j, \end{aligned} \tag{12}$$

### 4.2 Generalized Laplace mechanism

Although there are methods to generate high-dimensional Laplacian noise, no one satisfies a specific correlation matrix. Therefore, we provide a noise mechanism to meet the DP definition, the generalized Laplacian mechanism, which is described in Definition 5 in this section.

**Definition 5** (*Generalized Laplacian mechanism* (Wang and Wang 2021)) Let vector $Y = (y_1, y_2, \cdots, y_n)'$ be the noise added in the query result. If the noise vector obeys the generalized Laplacian distribution, $Y\tilde{G}L(\lambda, \mathbf{C_Q})$, the privacy protection mechanism $M$ can be guaranteed to satisfy $\epsilon$-DP. The probability density function of the generalized Laplacian distribution is,

$$\rho(\lambda, Y) = \frac{1}{(2\pi)^{(1/2)}} \frac{2}{\lambda} \frac{K_{-0.5}\left(\sqrt{2q(Y)/\lambda}\right)}{\left(\sqrt{\lambda q(Y)/2}\right)^{-1/2}}, \tag{13}$$

where

$$q(Y) = Y' \mathbf{C}_\mathbf{Q}^{-1} Y, \tag{14}$$

where $Y'$ is the transposed matrix of the noise vector $Y$; $\mathbf{C_Q}$ is the correlation matrix of the query output, and $K_m(\cdot)$ represents the second type of $m$-order-modified Bessel function.

Although the Definition 5 gives the probability density function of the generalized Laplacian mechanism, it is challenging to generate a noise sequence that follows the probability density function during continuous queries. An algorithm, which generates generalized Laplacian noise in Definition 5 with an iterative mechanism and is used for practical applications, is provided in the following section.

### 4.3 Noise iterative algorithm

This section designs an iterative mechanism to generate variables that obey the generalized Laplacian distribution with a particular correlation. A bivariate Laplace variable is generated followed by the production of the noise sequence with specific correlation, by applying the designed iterative mechanism and Gaussian distribution.

A Laplacian random variable can be generated by multiplying an exponential random variable and a Gaussian variable. Since Gaussian random variables with specific covariance matrices can be generated, the exponential distribution and Gaussian distribution are combined in this paper, as the mechanism to generate related Laplacian random variables.

**Lemma 1** *Considering that* $\mathbf{G}_{\mathcal{K},\mathcal{U}}$ *is a pair of zero-mean bivariate Gaussian random variables, the covariance matrix equals the original data autocovariance matrix,* $\mathbf{C}_{\mathcal{K},\mathcal{U}}$ *and assuming that W is an exponentially distributed random variable, a set of bivariate correlation Laplacian random variable,, with covariance matrix,* $Y_{\mathcal{K},\mathcal{U}}$, $\mathbf{C}_{\mathcal{K},\mathcal{U}}$ *can be generated by*:

$$Y_{\mathcal{K},\mathcal{U}} = \sqrt{W} \mathbf{C}_{\mathcal{K},\mathcal{U}}^{(1/2)} \mathbf{G}_{\mathcal{K},\mathcal{U}}, \tag{15}$$

*where W and* $\mathbf{G}_{\mathcal{K},\mathcal{U}}$ *are generated independently. The probability density function is*

$$p_W(w) = \frac{1}{\lambda} \exp\left(-\frac{w}{\lambda}\right). \tag{16}$$

Importantly, a Gaussian variable with a specific covariance matrix is required to decompose the symmetric positive definite covariance matrix into two diagonal matrices and a positive definite matrix, by using eigenvalues, singular values and Cholesky decomposition. The sensitivity function corresponding to the uncorrelated probability density is known as the Euclidean distance, and the corresponding probability density, as discussed in this paper, is known as the covariance distance (or Mahalanobis distance).

Therefore, two practical considerations are: (1) Employing the Laplacian random variable pairs, which are provided in the previous section, to counter consecutive queries, and (2) Countering repeated queries initiated by third party agencies. We present an iterative mechanism to answer the continuous and repeated queries. Particularly, when a given query is different from a previous one, the mechanism generates a new Laplacian noise based on the Gaussian distribution. Moreover, the variables generated by the exponential distribution are updated with a renewal function, to counter the repeated query. Further, the Gaussian distribution is used to solve the first problem.

The conditional distribution of a bivariate Gaussian variable is a Gaussian distribution. We employed this property to generate the required noise to counter consecutive queries. The conditional distribution of the bivariate Gaussian distribution is normalized, as described in the Theorem 1.

**Theorem 1** *The bivariate Gaussian distribution is denoted by* $\mathbf{G}_{\mathcal{K},\mathcal{U}} \sim \tilde{N}(\boldsymbol{\mu}, \mathbf{C}_{\mathcal{K},\mathcal{U}})$. *The scale parameters are*:

$$\mathbf{G}_{\mathcal{K},\mathcal{U}} = \begin{pmatrix} G_\mathcal{K} \\ G_\mathcal{U} \end{pmatrix}, \boldsymbol{\mu} = \begin{pmatrix} \mu_\mathcal{K} \\ \mu_\mathcal{U} \end{pmatrix}, \mathbf{C}_{\mathcal{K},\mathcal{U}} = \begin{pmatrix} C_{\mathcal{K}\mathcal{K}} & C_{\mathcal{K}\mathcal{U}} \\ C_{\mathcal{U}\mathcal{K}} & C_{\mathcal{U}\mathcal{U}} \end{pmatrix}. \tag{17}$$

The conditional distribution, $G_\mathcal{U}$, satisfies $G_{\mathcal{K}|\mathcal{U}} \sim \tilde{N}(\mu_{\mathcal{K}\cdot\mathcal{U}}, C_{\mathcal{K}\cdot\mathcal{U}})$, where $\mu_{\mathcal{K}\cdot\mathcal{U}} = \mu_\mathcal{K} + C_{\mathcal{K}\mathcal{U}} C_{\mathcal{U}\mathcal{U}}^{-1}(G_\mathcal{U} - \mu_\mathcal{U})$ and $C_{\mathcal{K}\cdot\mathcal{U}} = C_{\mathcal{K}\mathcal{K}} - C_{\mathcal{K}\mathcal{U}} C_{\mathcal{U}\mathcal{U}}^{-1} C_{\mathcal{U}\mathcal{K}}$.

***Proof*** Let $\mathbf{A} = \begin{pmatrix} 1 & C_{\mathcal{K}\mathcal{U}} C_{\mathcal{U}\mathcal{U}}^{-1} \\ 0 & 1 \end{pmatrix}$, and

$$\mathbf{L}_{\mathcal{K},\mathcal{U}} = \begin{pmatrix} L_{\mathcal{K}} \\ L_{\mathcal{U}} \end{pmatrix} = \mathbf{A}\mathbf{G}_{\mathcal{K},\mathcal{U}} = \begin{pmatrix} 1 & -C_{\mathcal{K}\mathcal{U}}C_{\mathcal{U}\mathcal{U}}^{-1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} G_{\mathcal{K}} \\ G_{\mathcal{U}} \end{pmatrix} \cdot = \begin{pmatrix} G_{\mathcal{K}} - C_{\mathcal{K}\mathcal{U}}C_{\mathcal{U}\mathcal{U}}^{-1}G_{\mathcal{U}} \\ G_{\mathcal{U}} \end{pmatrix} \tag{18}$$

Therefore,

$$E(L_{\mathcal{K}}) = \mu_{\mathcal{K}} - C_{\mathcal{K}\mathcal{U}}C_{\mathcal{U}\mathcal{U}}^{-1}\mu_{\mathcal{U}}$$

$$Var(L_{\mathcal{K}})$$

and
$$= \left(1, -C_{\mathcal{K}\mathcal{U}}C_{\mathcal{U}\mathcal{U}}^{-1}\right) \begin{pmatrix} C_{\mathcal{K}\mathcal{K}} & C_{\mathcal{K}\mathcal{U}} \\ C_{\mathcal{U}\mathcal{K}} & C_{\mathcal{U}\mathcal{U}} \end{pmatrix} \left(1, -C_{\mathcal{U}\mathcal{U}}C_{\mathcal{U}\mathcal{K}}^{-1}\right)'$$
$$= C_{\mathcal{K}\mathcal{K}} - C_{\mathcal{K}\mathcal{U}}C_{\mathcal{U}\mathcal{U}}^{-1}C_{\mathcal{U}\mathcal{K}}$$
$$= C_{\mathcal{K}\cdot\mathcal{U}}$$

Therefore, $L_{\mathcal{K}} \sim \tilde{N}(\mu_{\mathcal{K}} - C_{\mathcal{K}\mathcal{U}}C_{\mathcal{U}\mathcal{U}}^{-1}\mu_{\mathcal{U}}, C_{\mathcal{K}\cdot\mathcal{U}})$. Since $L_{\mathcal{K}}$ and $L_{\mathcal{U}}$ are independent,

$$\rho(\mathbf{L}_{\mathcal{K},\mathcal{U}}) = \rho(L_{\mathcal{K}}) \cdot \rho(L_{\mathcal{U}})$$

Moreover, $J(\mathbf{G}_{\mathcal{K},\mathcal{U}} \to \mathbf{L}_{\mathcal{K},\mathcal{U}}) = |\mathbf{A}^{-1}| = |\mathbf{A}|^{-1}$ and $J(\mathbf{G}_{\mathcal{K},\mathcal{U}} \to \mathbf{L}_{\mathcal{K},\mathcal{U}}) = |\mathbf{A}^{-1}| = |\mathbf{A}|^{-1}$. Thus,

$$J(\mathbf{L}_{\mathcal{K},\mathcal{U}} \to \mathbf{G}_{\mathcal{K},\mathcal{U}}) = 1/J(\mathbf{G}_{\mathcal{K},\mathcal{U}} \to \mathbf{L}_{\mathcal{K},\mathcal{U}}) = 1$$

Therefore,

$$\begin{aligned} \rho(\mathbf{G}_{\mathcal{K},\mathcal{U}}) &= \rho(\mathbf{L}_{\mathcal{K},\mathcal{U}})|J(\mathbf{G}_{\mathcal{K},\mathcal{U}} \to \mathbf{L}_{\mathcal{K},\mathcal{U}})| \\ &= \rho(\mathbf{L}_{\mathcal{K},\mathcal{U}}) \\ &= \rho(L_{\mathcal{K}}) \cdot \rho(L_{\mathcal{U}}) \\ &= \rho(L_{\mathcal{K}}) \cdot \rho(G_{\mathcal{U}}) \end{aligned}$$

Thus, the probability density function for given $G_{\mathcal{U}}$ and $G_{\mathcal{K}|U}$ is

$$\begin{aligned} p(G_{\mathcal{K}|\mathcal{U}}) &= \frac{p(\mathbf{G}_{\mathcal{K},\mathcal{U}})}{p(G_{\mathcal{U}})} = p(L_{\mathcal{K}}) \\ &= (2\pi)^{-1/2}|C_{\mathcal{K}\cdot\mathcal{U}}|^{-1/2} \\ &\quad \cdot \exp[-\frac{1}{2}(L_{\mathcal{K}} - \mu_{\mathcal{K}} + C_{\mathcal{K}\mathcal{U}}C_{\mathcal{U}\mathcal{U}}^{-1}\mu_{\mathcal{U}})^2 C_{\mathcal{K}\cdot\mathcal{U}}^{-1}] \\ &= (2\pi)^{-1/2}|C_{\mathcal{K}\cdot\mathcal{U}}|^{-1/2} \\ &\quad \cdot \exp[-\frac{1}{2}(G_{\mathcal{K}} - C_{\mathcal{K}\mathcal{U}}C_{\mathcal{U}\mathcal{U}}^{-1}G_{\mathcal{U}} - \mu_{\mathcal{K}} \\ &\quad + C_{\mathcal{K}\mathcal{U}}C_{\mathcal{U}\mathcal{U}}^{-1}\mu_{\mathcal{U}})^2 C_{\mathcal{K}\cdot\mathcal{U}}^{-1}] \\ &= (2\pi)^{-1/2}|C_{\mathcal{K}\cdot\mathcal{U}}|^{-1/2}\exp[-\frac{1}{2}(G_{\mathcal{K}} \\ &\quad - \mu_{\mathcal{K}\cdot\mathcal{U}})^2 C_{\mathcal{K}\cdot\mathcal{U}}^{-1}] \end{aligned}$$

Therefore, $G_{\mathcal{K}|\mathcal{U}} \sim \tilde{N}(\mu_{\mathcal{K}\cdot\mathcal{U}}, C_{\mathcal{K}\cdot\mathcal{U}})$.

The following inferences can be obtained from the Theorem 1. If the initial Laplacian random variable, $y_{\mathcal{U}}$, is generated by a pair of independent exponentials and Gaussian random variables,

$$y_{\mathcal{U}} = \sqrt{W} \cdot G_{\mathcal{U}}. \tag{19}$$

This method is used to independently generate another Laplacian random variable, $y_{\mathcal{K}}$, where $G_{\mathcal{K}} \sim \tilde{N}(\mu_{\mathcal{K}\cdot\mathcal{U}}, C_{\mathcal{K}\cdot\mathcal{U}})$, and the covariance of $y_{\mathcal{K}}$ and $y_{\mathcal{U}}$ is $\mathbf{C}_{\mathcal{K},\mathcal{U}}$. Proof. The proof process is the inverse to Theorem 1. □

**Definition 6** (Repeated renewal function) Let $Q_1, Q_2, \cdots, Q_n$ be the query sequence. If $Q_{t+1} = Q_t$, the function $U(\cdot)$ is defined as a repeated renewal function. If $U(\cdot)$ satisfies $y_{t+1} = U(y_t)$,

$$y_{t+1} = \sqrt{W_{t+1}} \cdot G_t, \tag{20}$$

where, $G_t$ is the Gaussian random variable to generate Laplace noise for the previous query, and $W_{t+1}$ is the newly generated standard exponential variable.

Instead of generating Laplacian noise with greater sensitivity, the iterative renewal process updates the exponential variable, to regenerate Laplacian random noise that counters the repeated queries.

### 4.4 Algorithm design

The statistical investigation demonstrates that the POI discovery application is a counting query. When the differential privacy mechanism is applied to protect the investigation of the POI discovery, the maximum impact of a single record on the statistical result is 1, $\Delta f = 1$. The statistical query dataset initiated by the record is denoted as $\mathbf{Q} = \{Q_1, \ldots, Q_n\}$, with a correlation between any two queries, such as $Q_i$ and $Q_j$. According to the indistinguishable theory of related data proposed in this paper, the goal of the differential privacy for POI discovery is to generate Laplacian noise with the consistency of the query results. We employ the covariance matrix to represent the correlation between the query results. Section 4.1 presents the formula for calculating the covariance between any two POIs, such as $p_i$ and $p_j$. The following section calculates the covariance matrix $\mathbf{C}_{Q_i,Q_j}$ between two random queries, $Q_i$ and $Q_j$, as described in Theorem 2.

If the attacker launches the same queries, that is, $Q_{i+1} = Q_i$, the privacy budget will increase in related work because the privacy degree may decrease along with the same queries. However, this problem does not exist in different queries. Considering this issue, the paper uses different noise according to whether $Q_{i+1}$ repeats $Q_i$.

**Theorem 2** *Given that covariance matrix between two POIs, $p_i$ and $p_j$, is $\mathbf{C}_{p_i,p_j}$, $Q_i$ and $Q_j$ are two random count queries in the query dataset* $\mathbf{Q}$, *and the POIs datasets to be queried are, $P_{\mathcal{K}}$ and $P_{\mathcal{U}}$, respectively, where, $P_{\mathcal{K}}, P_{\mathcal{U}} \in \mathbf{P}$ and the query results are $f(P_{\mathcal{K}})$ and $f(P_{\mathcal{U}})$, respectively. The*

*covariance matrix between the query results of $Q_i$ and $Q_j$ is*, $\mathbf{C}_{Q_i,Q_j} = \sum_{p_i \in P_\mathcal{K}, p_i \in P_\mathcal{U}} \mathbf{C}_{p_i,p_j}$.

***Proof*** Upon expanding $\mathbf{C}_{Q_i,Q_j}$, we obtain

$$\mathbf{C}_{Q_i,Q_j} = \mathrm{cov}\left[f(P_\mathcal{K}), f(P_\mathcal{U})\right] = \mathrm{cov}\left[\sum_{p_i \in P_\mathcal{K}} p_i, f(P_\mathcal{U})\right]. \quad (21)$$

According to the operation of the covariance matrix,

$$\begin{aligned}
\mathbf{C}_{Q_i,Q_j} &= \mathrm{cov}\left[\sum_{p_i \in P_\mathcal{K}} p_i, f(P_\mathcal{U})\right] \\
&= \sum_{p_i \in P_\mathcal{K}} \mathrm{cov}\left[p_i, f(P_\mathcal{U})\right] \\
&= \sum_{p_i \in P_\mathcal{K}} \mathrm{cov}\left[p_i, \sum_{p_j \in P_\mathcal{U}} p_j\right] \\
&= \sum_{p_i \in P_\mathcal{K}, p_i \in P_\mathcal{U}} \mathrm{cov}(p_i, p_j) \\
&= \sum_{p_i \in P_\mathcal{K}, p_i \in P_\mathcal{U}} \mathbf{C}_{p_i,p_j}.
\end{aligned} \quad (22)$$

The correlation matrix of the query output, $\mathbf{C_Q}$ can be obtained. According to the generalized Laplace mechanism, proposed in Sect. 4.3, for countering continuous queries, an arbitrary Gaussian variable noise, $G_\mathcal{U}$, is generated followed by conditional Gaussian variable noise, $G_{\mathcal{K}|\mathcal{U}}$, based on covariance matrix. The covariance matrix of the bivariate Gaussian variable, $\mathbf{G}_{\mathcal{K},\mathcal{U}} = (G_{\mathcal{K}|\mathcal{U}}, G_\mathcal{U})'$ is $\mathbf{C}_{\mathcal{K},\mathcal{U}}$. Therefore, the Laplace noise $y_\mathcal{K} = \sqrt{W} \cdot G_{\mathcal{K}|\mathcal{U}}$ and $y_\mathcal{U} = \sqrt{W} \cdot G_\mathcal{U}$, generated from the bivariate Gaussian variable are the bivariate Laplacian noise with the covariance matrix, $\mathbf{C}_{\mathcal{K},\mathcal{U}}$. Algorithm 1 denotes the implementation process for differential privacy protection for POI discovery. □

**Algorithm 1** Differential Privacy for POI Discovery

**Require:** $\epsilon$, $\mathbf{Q} = \{Q_1, Q_2, \cdots, Q_n\}$, $S(f)$.

**Ensure:** $\mathbf{Q}' = \{Q'_1, Q'_2, \cdots, Q'_n\}$.

**for** each round $i \leftarrow 1, \cdots, n$ **do**

1. Launch query $Q_i$;
2. Initialize $y_i \leftarrow Laplace(S(f)/\varepsilon)$, $S(f) = 1$;
3. Compute $Q'_i = Q_i + y_i$;
4. Launch new query $Q_{i+1}$;

**if** $Q_{i+1} = Q_i$ **then**

5. Generate new normal Exponential variable $W_{i+1}$;
6. $y_{i+1} \leftarrow \sqrt{W_{i+1}} \cdot G_i$;

**else**

7. Generate conditional Gaussian variable $G_{i+1|i}$ according to Theorem 1;
8. Generate new Laplace variable $Y_{i+1}$ according to Theorem 2;

**end if**

9. Compute perturbation $Q'_{i+1} = Q_{i+1} + y_{i+1}$;

**end for**

**return** $\mathbf{Q}'$.

In terms of Algorithm 1, it is an algorithm to generate Laplace noise variables with a specific correlation matrix, but the attacker always tries to analyze the results by sending repeated queries. In this case, we must generate another new noise to protect individual's true value. So steps 5 and 6 are to generate another new Laplace variable to answer repeated queries, while steps 7 and 8 are to generate a new Laplace variable to answer other queries. Even if they are different in noise generation methods, the generated noise can meet the required correlation matrix.

To generate the Laplace noise with a specific correlation matrix, we utilize the property of Gaussian distribution. We have known that the conditional distribution of a Gaussian distribution also follows Gaussian, and Theorem 1 gives the form of the conditional Gaussian distribution form if we want these variables to meet a required correlation variance. So we firstly initialize a Gaussian noise, then we generate another Gaussian variable according to the form of conditional Gaussian distribution in Theorem 1. These two Gaussian variables can meet the correlation calculated in Sect. 4.1.

To generate new Laplace variables to answer different queries, we firstly calculated the correlation of different queries. Then we generate conditional Gaussian variables which follow the correlation according to Theorem 2. Finally, we generate new Exponential distribution variables and get the new Laplace variable according to Eq. (20).

### 4.5 Complexity analysis

In this section, we analyze the time complexity of our solution over correlated POI data. Since running environments, programming languages and coding styles vary in different systems, generally, the computation complexity is evaluated by the notation "$\mathcal{O}$", which counts critical programming statements in iterations.

As shown in Algorithm 1, the practical procedure of our solution includes 9 steps. Among them, step 8 costs the most complexity, $\mathcal{O}(2n^2)$. While the complexity of the other steps, including steps 1, 2, 3, 4, 5, 6, 7 and 9, are $\mathcal{O}(n)$, $\mathcal{O}(1)$, $\mathcal{O}(n)$, $\mathcal{O}(n)$, $\mathcal{O}(n)$, $\mathcal{O}(1)$, $\mathcal{O}(n^2)$ and $\mathcal{O}(n)$ respectively. Thus, the total computation complexity of our solution, $T(n)$, is

$$\begin{aligned} T(n) &= \mathcal{O}(n) + \mathcal{O}(1) + \mathcal{O}(n) + \mathcal{O}(n) + \mathcal{O}(n) + \mathcal{O}(1) \\ &\quad + \mathcal{O}(n^2) + \mathcal{O}(2n^2) + \mathcal{O}(n) = \mathcal{O}(3n^2 + 5n + 2) \approx \mathcal{O}(n^2) \end{aligned} \tag{23}$$

Equation (23) indicates that our solution, which has a low computation complexity, can be conducted in polynomial time.

## 5 Experimental evaluation

We evaluate our correlated POI release solution from security, utility and computational cost and compare it with current representative schemes.

### 5.1 Experimental setup

We evaluate the performance of our solution on real-world datasets. The experiments are conducted on a Windows 10 machine equipped with Intel Core 2 Quad 3.5 Hz and 16 GB memory.

Three real-world datasets are tested in this paper, with each experiment running 1000 times. The dataset details are as follows:

Foursquare[1]: The geo-location-based service website, Foursquare.com, hosts the user's check-in data from March 2010 to December 2011, including 18,293 users, 43,186 POIs and 1,903,909 check-ins.

Gowalla[2]: Similar to Foursquare, Gowalla is a mobile-phone based application that provides geolocation-based services. Users can check in at the nearby POIs via local mobile apps or mobile websites. After pre-processing, the experimental dataset contains 18,995 POIs from 3,887 users.

Check-in[3]: This data set consists of check-in data generated by more than 49,000 users in New York and 31,000 users in Los Angeles, and the users' social structures. Each check-in record includes a POI ID, a POI category, a timestamp, and a user ID.

After the pre-data clean-up, integration and reduction of the three datasets, a check-in matrix is generated. The row and column vectors of the check-in matrix are the POI and the user ID, respectively. The matrix elements 1 and 0 indicate the presence and absence of check-in information for that user at the POI. The statistical significance of the data is investigated.

Here, the impact of the prevalent POI discovery algorithm and the Top-$k$ recommendation algorithm are evaluated in POI discovery applications, as an example. Our algorithm is compared with the Top-$k$ recommendation algorithm (Baseline) without privacy protection (Eltarjaman et al. 2016), Markov model-based DCHRG and algorithm proposed by Wang et al. (2018a). Here, the recall rate (Recall, $R$), precision (Precision, $P$) and $F$ value are used to measure the accuracy of the recommendation. $R$ is the ratio of the number of recommendations obtained for a POI by the recommended algorithms to that for the total number of POIs. P is the ratio of the number of recommendations obtained for the

[1] https://foursquare.com/.

[2] https://mashable.com/category/gowalla/.

[3] https://sites.google.com/site/yangdingqi/home/foursquare-dataset.

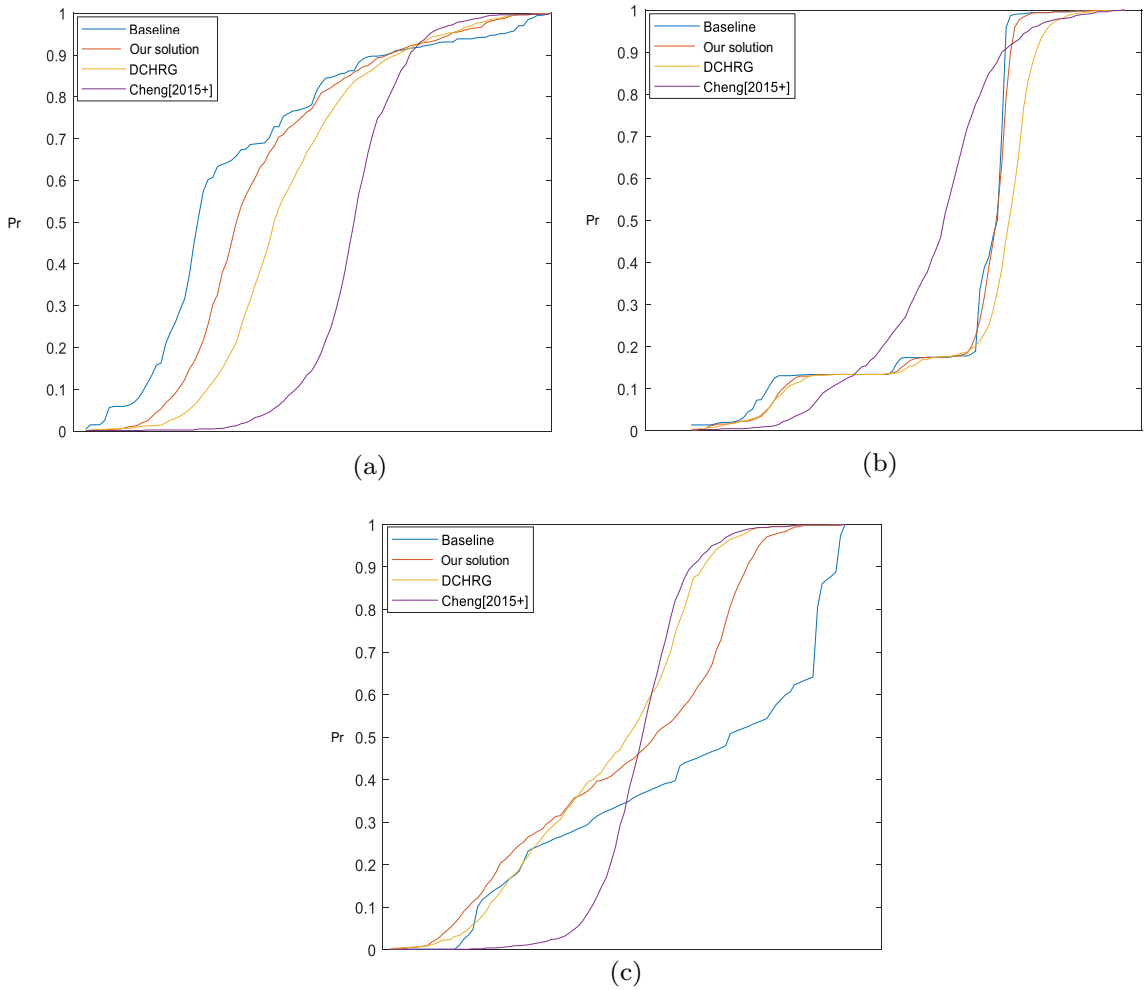


**Fig. 3** Comparison of probability distributions of three methods under the indicated datasets

POIs by the recommendation algorithm to the number of all POIs returned by the recommendation algorithm. The $F$ value is a comprehensive indicator for adjusting the average $R$ and $P$. Assuming that $A$ represents a set of all POIs, and $B$ represents a set of POIs returned by the recommendation algorithm,

$$R = \frac{|A \cap B|}{|A|}. \tag{24}$$

$$P = \frac{|A \cap B|}{|B|}. \tag{25}$$

$$F = \frac{|R + P|}{|R \cdot P|}. \tag{26}$$

## 5.2 Experimental results and analysis

This section evaluates the performance of the differential privacy protection algorithm in POI discovery. The experiment evaluates the algorithm from two aspects, a) including privacy security assessment and b) data availability assessment. For privacy security assessment, the comparison of probability distributions of different methods, before and after the attack, are provided. For the data availability assessment, the errors in queries under different methods and the impact on the recommendation for performance of POI discovery are evaluated.

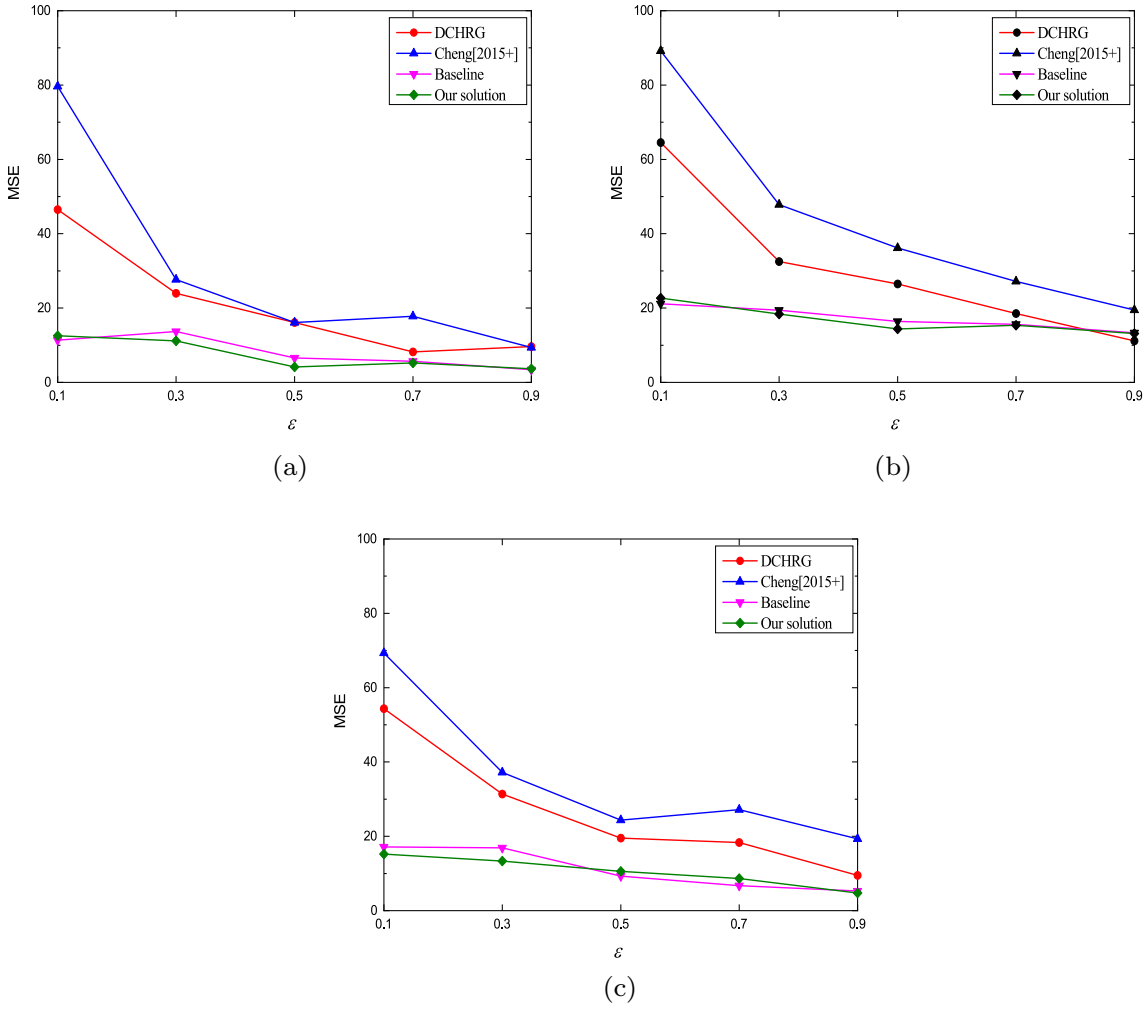


**Fig. 4** Comparison of MSEs of three methods under the indicated datasets

#### 5.2.1 Privacy assessment

Figure 3 depicts the probability distribution of the Top-$k$ recommendation, where $Pr$ represents the abbreviation of probability. Figure 3 also compares algorithm on the three tested data sets before and after utilizing the proposed method. While, the Top-$k$ recommendation algorithm (Eltarjaman et al. 2016) is set as a baseline without noise, DCHRG (Wang et al. 2018a; Ren et al. 2021a) and our algorithm are set with $\epsilon = 1$. The probability distribution with our algorithm, in the three tested datasets, is closest to that of the Top-$k$ recommendation algorithm, suggesting comparable statistical characteristics with the Top-$k$ algorithm. Therefore, despite knowing the relevance of the query results, the attackers would be unable to filter out the noise that is following the relevant characteristics of the query results, and consequently unable to infer the private POI information.

#### 5.2.2 Usability assessment

The experiment evaluated the usability of the algorithm by testing the MSE, $R$, $P$, and $F$ values of different algorithms on three datasets.

#### 5.2.3 MSE

Figure 4 and Table 2 depict the comparison of MSE on the three datasets for the employed methods. Since our algorithm does not need to increase the noise to protect the privacy, similar to the original mechanism, the MSE of our algorithm is similar to that of the original differential privacy mechanism, while the algorithms proposed by DCHRG and Cheng et al. show deviance. Therefore, the methods proposed by DCHRG and Cheng et al. would need to recalculate the added noise as the sensitivity measure and thus require increased noise for efficient privacy protection (Tables 3, 4, 5).

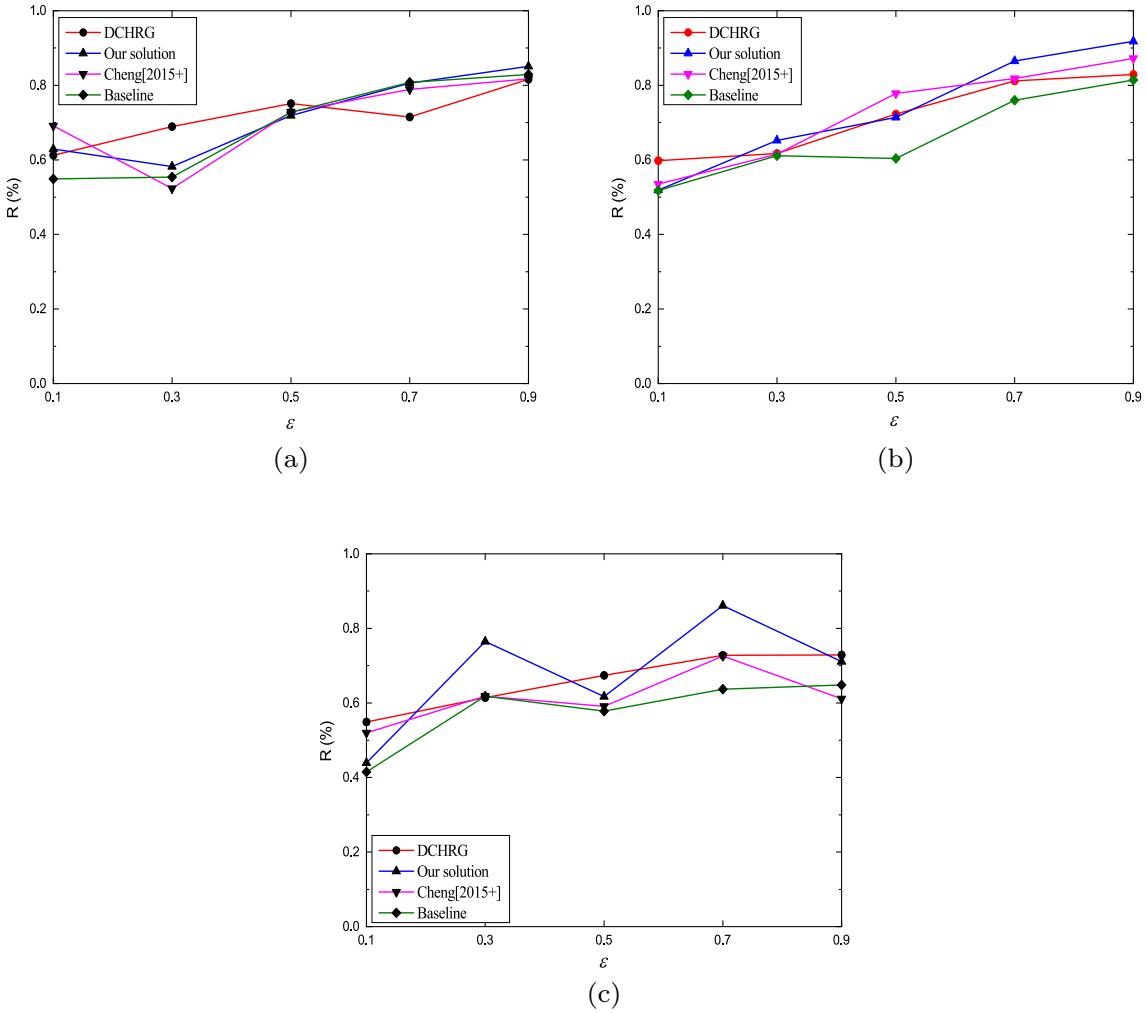


**Fig. 5** Comparison of the *R* of three methods under the indicated datasets

#### 5.2.4 *R*, *P*, and *F* value

Figures 5, 6 and 7 depict *R*, *P* and *F* values for the different methods employed for the three datasets. Most of the existing methods can maintain a *R* of more than 60% and the differences between recall performance of different methods is comparable (Fig. 5). However, compared to the existing methods, our algorithm can maintain a higher *R*, in most of the tested cases. Moreover, increase in privacy budget, $\epsilon$, has been demonstrated to decrease its protection strength and increase the *R* of all the algorithms, owing to reduction in the noise added to the POI data. Similar trend can be observed in Figs. 6 and 7. Comparison of comprehensive performance (Fig. 7) demonstrated that smaller $\epsilon$ leads to lower differences in the performance of the existing methods, since with smaller $\epsilon$, larger noise is added to the POI data, which overshadows statistical outcome. Increasing $\epsilon$ gradually reduces the noise added to the POI data. This feature strengthens the usability of the proposed algorithm. When $\epsilon$ is increased to 0.9, the proposed algorithm has better *F* value compared to the existing algorithms, for all the three datasets tested, thereby verifying the effectiveness of this algorithm.

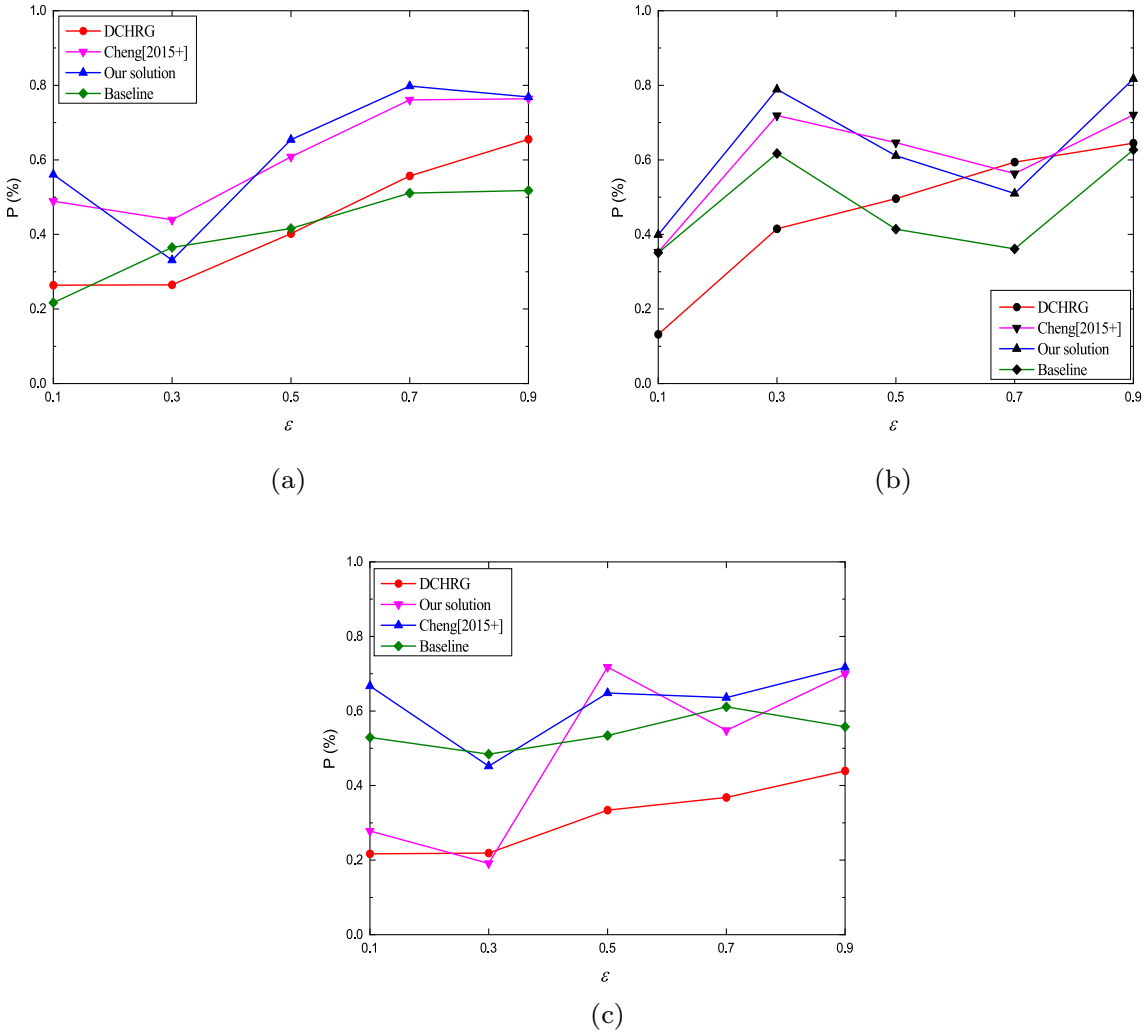


**Fig. 6** Comparison of the *P* of three methods under the indicated datasets

## 6 Conclusions and future works

Although DP provides a better trade-offs between privacy preserving and data utility, there remains a limiting assumption in the standard DP that can severely serve for independent data. In this paper, we analyze the properties of current mechanisms for differentially private publication of correlated POI data and demonstrate that the model based or resizing sensitivity will lead to rigorous restriction and introduce extra noise.

Consequently, instead of IID noise, we present an efficient publishing approach by introducing a correlated Laplace mechanism. It renders the correlation of noise and POI indistinguishable to an adversary and guarantees unconditional security. Extensive experiments on real-life datasets demonstrate that our solution outperforms the other approaches for a large volume of queries and maintains significantly high levels of data utility while preserving the privacy.

Although our solution is effective, there are still some aspects to be improved in the future. Future work includes expanding our solution to other scenarios, such as correlated trajectory prediction, trajectory pattern mining, etc.

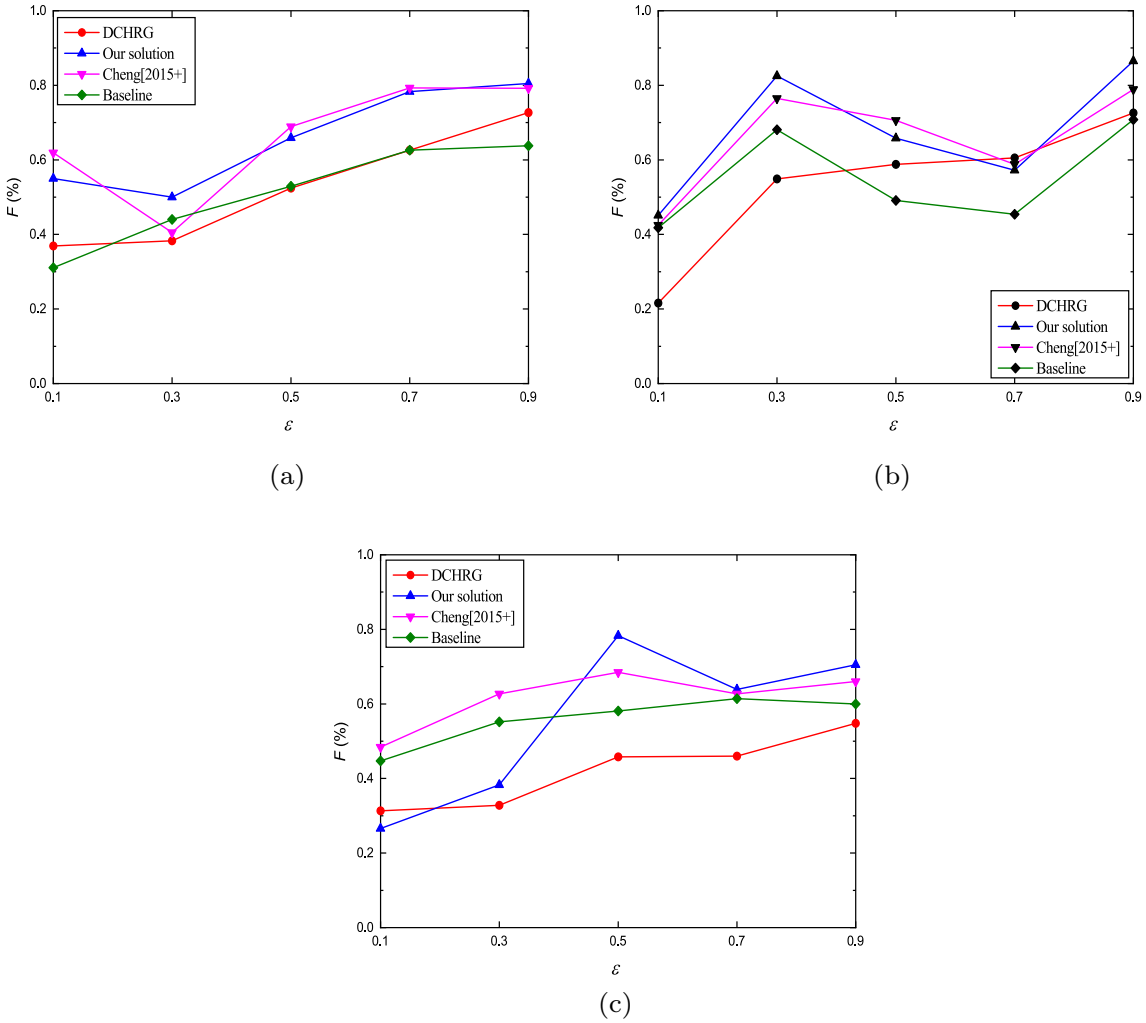


**Fig. 7** Comparison of the $F$ values of three methods under the indicated datasets

**Table 2** MSE of different methods under different datasets

| Datasets | $\varepsilon$ | DCHRG | Cheng[2015+] | Baseline | Our solution |
|---|---|---|---|---|---|
| Foursquare | 0.1 | 46.515 | 79.648 | 11.369 | 12.536 |
| | 0.3 | 23.957 | 27.673 | 13.698 | 11.156 |
| | 0.5 | 16.102 | 16.085 | 6.592 | 4.165 |
| | 0.7 | 8.201 | 17.809 | 5.698 | 5.245 |
| | 0.9 | 9.653 | 9.411 | 3.387 | 3.651 |
| Gowalla | 0.1 | 64.548 | 89.156 | 21.167 | 22.678 |
| | 0.3 | 32.486 | 47.846 | 19.396 | 18.359 |
| | 0.5 | 26.496 | 36.185 | 16.415 | 14.365 |
| | 0.7 | 18.516 | 27.155 | 15.615 | 15.379 |
| | 0.9 | 11.165 | 19.491 | 13.353 | 13.145 |
| Check-in | 0.1 | 54.346 | 69.319 | 17.153 | 15.197 |
| | 0.3 | 31.394 | 37.189 | 16.897 | 13.328 |
| | 0.5 | 19.497 | 24.365 | 9.265 | 10.584 |
| | 0.7 | 18.349 | 27.187 | 6.698 | 8.657 |
| | 0.9 | 9.489 | 19.329 | 5.297 | 4.765 |

**Table 3** $R$ of different methods under different datasets

| Datasets | $\varepsilon$ | DCHRG | Cheng[2015+] | Baseline | Our solution |
|---|---|---|---|---|---|
| Foursquare | 0.1 | 0.612 | 0.629 | 0.691 | 0.549 |
| | 0.3 | 0.689 | 0.582 | 0.523 | 0.554 |
| | 0.5 | 0.751 | 0.719 | 0.729 | 0.728 |
| | 0.7 | 0.715 | 0.806 | 0.789 | 0.808 |
| | 0.9 | 0.816 | 0.851 | 0.817 | 0.829 |
| Gowalla | 0.1 | 0.598 | 0.519 | 0.535 | 0.518 |
| | 0.3 | 0.617 | 0.652 | 0.615 | 0.611 |
| | 0.5 | 0.723 | 0.714 | 0.778 | 0.604 |
| | 0.7 | 0.812 | 0.865 | 0.818 | 0.76 |
| | 0.9 | 0.829 | 0.918 | 0.872 | 0.814 |
| Check-in | 0.1 | 0.549 | 0.439 | 0.52 | 0.415 |
| | 0.3 | 0.614 | 0.765 | 0.618 | 0.618 |
| | 0.5 | 0.674 | 0.617 | 0.591 | 0.578 |
| | 0.7 | 0.728 | 0.861 | 0.726 | 0.637 |
| | 0.9 | 0.729 | 0.711 | 0.611 | 0.648 |

**Table 4** *P* of different methods under different datasets

| Datasets | $\varepsilon$ | DCHRG | Cheng[2015+] | Baseline | Our solution |
|---|---|---|---|---|---|
| Foursquare | 0.1 | 0.264 | 0.561 | 0.489 | 0.217 |
| | 0.3 | 0.265 | 0.331 | 0.439 | 0.365 |
| | 0.5 | 0.402 | 0.654 | 0.608 | 0.416 |
| | 0.7 | 0.557 | 0.798 | 0.761 | 0.511 |
| | 0.9 | 0.655 | 0.769 | 0.764 | 0.518 |
| Gowalla | 0.1 | 0.132 | 0.399 | 0.353 | 0.351 |
| | 0.3 | 0.415 | 0.789 | 0.719 | 0.617 |
| | 0.5 | 0.496 | 0.611 | 0.646 | 0.414 |
| | 0.7 | 0.594 | 0.51 | 0.563 | 0.361 |
| | 0.9 | 0.645 | 0.817 | 0.721 | 0.627 |
| Check-in | 0.1 | 0.217 | 0.667 | 0.278 | 0.529 |
| | 0.3 | 0.219 | 0.452 | 0.191 | 0.484 |
| | 0.5 | 0.334 | 0.648 | 0.718 | 0.534 |
| | 0.7 | 0.368 | 0.636 | 0.548 | 0.611 |
| | 0.9 | 0.439 | 0.717 | 0.699 | 0.558 |

**Table 5** *F* of different methods under different datasets

| Datasets | $\varepsilon$ | DCHRG | Cheng[2015+] | Baseline | Our solution |
|---|---|---|---|---|---|
| Foursquare | 0.1 | 0.369 | 0.55 | 0.619 | 0.311 |
| | 0.3 | 0.383 | 0.5 | 0.405 | 0.44 |
| | 0.5 | 0.524 | 0.659 | 0.689 | 0.529 |
| | 0.7 | 0.626 | 0.783 | 0.793 | 0.626 |
| | 0.9 | 0.727 | 0.805 | 0.792 | 0.638 |
| Gowalla | 0.1 | 0.216 | 0.451 | 0.425 | 0.418 |
| | 0.3 | 0.549 | 0.825 | 0.765 | 0.681 |
| | 0.5 | 0.588 | 0.658 | 0.706 | 0.491 |
| | 0.7 | 0.605 | 0.572 | 0.588 | 0.454 |
| | 0.9 | 0.726 | 0.865 | 0.789 | 0.708 |
| Check-in | 0.1 | 0.313 | 0.266 | 0.484 | 0.447 |
| | 0.3 | 0.328 | 0.383 | 0.627 | 0.552 |
| | 0.5 | 0.458 | 0.783 | 0.685 | 0.581 |
| | 0.7 | 0.46 | 0.639 | 0.627 | 0.614 |
| | 0.9 | 0.548 | 0.705 | 0.66 | 0.6 |

In addition, we will continue to study the applicability and universality of the method in this paper.

**Acknowledgements** This work was supported in part by the National NaturalScience Foundation of China (42001398), Chongqing Natural Science Foundation (cstc2020jcyj-msxmX0635), Scienceand Technology Research Project of Chongqing EducationCommission (KJQN201900612), Open Fund of State Lab-oratory of Information Engineering in Surveying, Mappingand Remote Sensing, Wuhan University (20S02), China Post-doctoral Science Foundation (2021M693929), the PhD StartsFund Project of Chongqing University of Posts and Telecom-munications (A2019-302) and SRTP of CQUPT (A2020-106).

## References

Cai L, Wen W, Wu B, Yang X (2021) A coarse-to-fine user preferences prediction method for point-of-interest recommendation. Neurocomputing 422:1–11

Dwork C (2006) Differential privacy. In: International Colloquium on Automata, Languages, and Programming, Springer, pp 1–12

Dwork C, Roth A et al (2014) The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci 9(3–4):211–407

Eltarjaman W, Dewri R, Thurimella R (2016) Private retrieval of poi details in top-k queries. IEEE Trans Mob Comput 16(9):2611–2624

Gambs S, Kégl B, Aïmeur E (2007) Privacy-preserving boosting. Data Min Knowl Disc 14(1):131–170

Lu YS, Shih WY, Gau HY, Chung KC, Huang JL (2019) On successive point-of-interest recommendation. World Wide Web 22(3):1151–1173

McSherry F, Talwar K (2007) Mechanism design via differential privacy. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), IEEE, pp 94–103

Pouke M, Goncalves J, Ferreira D, Kostakos V (2016) Practical simulation of virtual crowds using points of interest. Comput Environ Urban Syst 57:118–129

Ren W, Lian X, Ghazinour K (2021) Effective and efficient top-k query processing over incomplete data streams. Inf Sci 544:343–371

Ren W, Tong X, Du J, Wang N, Li SC, Min G, Zhao Z, Bashir AK (2021) Privacy-preserving using homomorphic encryption in mobile iot systems. Comput Commun 165:105–111

Schillings C, Sprungk B, Wacker P (2020) On the convergence of the laplace approximation and noise-level-robustness of laplace-based monte carlo methods for bayesian inverse problems. Numer Math 145(4):915–971

Wang H, Wang H (2021) Correlated tuple data release via differential privacy. Inf Sci 560:347–369

Wang H, Shen H, Ouyang W, Cheng X (2018a) Exploiting poi-specific geographical influence for point-of-interest recommendation. In: IJCAI, pp 3877–3883

Wang R, Li Y, Li F (2018) Probabilistic robustness for dispersive-dissipative wave equations driven by small laplace-multiplier noise. Dyn Syst Appl 27:165–183

Xi D, Zhuang F, Liu Y, Zhu H, Zhao P, Tan C, He Q (2020) Exploiting bi-directional global transition patterns and personal preferences for missing poi category identification. Neural Netw 132:75–83

Xu N, Feyisetan O, Aggarwal A, Xu Z, Teissier N (2020) Differentially private adversarial robustness through randomized perturbations. arXiv preprint arXiv:200912718

Yadav VK, Verma S, Venkatesan S (2020) Efficient and secure location-based services scheme in vanet. IEEE Trans Veh Technol 69(11):13,567-13,578

Yiu ML, Jensen CS, Møller J, Lu H (2011) Design and analysis of a ranking approach to private location-based services. ACM Trans Database Syst (TODS) 36(2):1–42

Zhu Q, Wang S, Cheng B, Sun Q, Yang F, Chang RN (2018) Context-aware group recommendation for point-of-interests. IEEE Access 6:12,129-12,144