Differentially Private Location Preservation with Staircase Mechanism Under Temporal Correlations

Rong Fang, Jianmin Han(✉), Juan Yu, Xin Yao, Hao Peng, and Jianfeng Lu

College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, China
hanjm@zjnu.cn

Abstract. Location-Based Service (LBS) is one of basic services in collaborative applications. However, LBS applications may disclose user's location privacy, which receives considerable concerns. Many methods have been proposed to protect privacy in LBS. Planar Isotropic Mechanism (PIM) is a typical location privacy preservation method in the scenario of continuous location data release. However, the method is complicated, since it requires two convex hull transformations and one isotropic position transform. To solve the problem, we propose a Staircase Mechanism (SM) based location privacy preservation method for the scenario of continuous location data release. The proposed method replaces PIM with SM, whose implementation is simple and efficient. Furthermore, SM can achieve the same privacy budget with less noise addition, so it can maintain higher quality of services in LBS. Comprehensive experiments conducted on real location data demonstrate that the proposed method is efficient and can maintain high data utility compared with the method based on PIM.

Keywords: Differential privacy · Location privacy · Temporal correlation · Staircase mechanism

1 Introduction

LBS is a kind of basic services in collaborative applications. With the development of positioning technologies and popularity of mobile Internet and smartphones, location-based applications have permeated into our daily life, such as location-based points of interest searching, location-based games, location-based commerce and location-based social networks. To enable the location-based applications, users have to share their locations to service providers. However, the disclosure of users' locations could raise serious privacy concerns. Because locations could reflect users' religion and health conditions, and further expose them to attacks, e.g., unwanted location-based spams, even physical danger [1] etc.

In order to protect the location privacy of users in LBS, various location privacy preservation technologies have been proposed. These technologies can be classified into

H. Gao et al. (Eds.): CollaborateCom 2020, LNICST 350, pp. 75–92, 2021.
https://doi.org/10.1007/978-3-030-67540-0_5

three categories, i.e., Private Information Retrieval (PIR) [2], location generalization [3], and location perturbation [4]. PIR is based on cryptography and can provide provable privacy preservation. However, it tends to be computationally expensive and not practical because that PIR need design different query for different query types. Location generalization hides a user's exact location in an area so that attackers cannot infer the exact location of the user with high probability. However, location generalization technologies rely on syntactic privacy models such as *k*-anonymity, or ad-hoc uncertainty models, and could not provide rigorous privacy. Location perturbation disturbs the sensitive location by adding random noises, so as to protect the sensitive location privacy. Differential privacy based on location privacy preservation are realized via location perturbation. Differential privacy [5] has been widely recognized as a leading privacy preservation method in both industrial and academic community, as it provides a formal and provable privacy guarantee and it can preserve privacy against attackers with arbitrary background knowledge. The idea is that the presence or absence of one single individual in a database shall not change significantly the probability of any outcome of an aggregate function.

Traditional location preservation methods only consider static scenarios or perturb the location at single timestamps without considering the temporal correlations of a moving user, and hence are vulnerable to various inference attacks. Therefore, Xiao et al. [6] considered the privacy leakage issue caused by the temporal correlation between locations, and proposed a systematic framework to protect location privacy for continuous location sharing scenarios. They first modeled the temporal correlation between locations with the Markov chain, and then introduced the concept of δ-location set to hide the real location, finally they proposed a Planar Isotropic Mechanism (PIM) to achieve δ-location set based differential privacy. However, the PIM is computational expensive as it requires two convex hull transformations and one isotropic position transform. Furthermore, the isotropic position transformation relies on an invertible matrix T which is too strict to be obtained.

To solve the problems of PIM, we propose a Staircase Mechanism (SM) based location perturbation method for location privacy preservation in continuous location sharing scenario. Main contributions are summarized as follows:

First, we propose a new perturbation mechanism, i.e., SM. Compared with the PIM, it could achieve the same privacy budget with less noise addition, and thereby maintaining higher quality of location-based services. Second, we verify theoretically and experimentally that SM has lower time complexity and better data utility compared with PIM.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 introduces the basics of differential privacy and the relevant definitions. Section 4 proposes differentially private location preservation method with staircase mechanism under temporal correlations. Section 5 evaluates the performance of the proposed method with extensive experiments. Section 6 concludes the paper.

2 Related Work

Differential privacy is an effective privacy preservation method on location or trajectory data. Machanavajjhala et al. [7] first introduced differential privacy into location

privacy preservation. They proposed differential privacy mechanism to securely release commuting patterns of users without compromising individuals' privacy. Andres et al. [8] proposed a geo-indistinguishability method based on differential privacy to make sure that attackers can hardly identify the difference between the exact location and the approximate locations within a circular region of radius r. Bordenabe et al. [9] devised an optimal geo-indistinguishable mechanism to minimize the LBS service quality loss. Niu et al. [10] investigated the long-term observation attacks on geo-indistinguishable mechanisms and proposed a three-phase differential location privacy framework, i.e., Eclipse, which combines geo-indistinguishability and k-anonymity. Gursoy et al. [11] proposed DP-Star, a framework, a methodical framework for publishing trajectory data with differential privacy guarantee as well as high utility preservation.

The methods mentioned above do not consider data correlations, which will lead in new privacy problems. Different types of correlations should adopt different privacy preservation method. To user-to-user correlation, Zhu et al. [12] defined the correlated differential privacy and the correlated sensitivity, and proposed a correlated data release mechanism to preserve privacy. Liu et al. [13] demonstrated the vulnerability of traditional differential privacy mechanisms under data dependence, and proposed a generalized dependent differential privacy framework, which introduced dependence coefficients to measure the sensitivity of different queries under probabilistic dependence between tuples. Yang et al. [14] concentrated on the privacy leakage caused by probabilistic correlations between tuples, and modeled the correlations by a Gaussian Markov Random Field and proposed Bayesian differential privacy (BDP). The other type is the temporal correlations among single user's data at different timestamp. Xiao et al. [6] investigated adversaries with knowledge of temporal correlations of single user, and proposed PIM for continuous location sharing scenarios. Cao et al. [15, 16] quantified the privacy loss of differential privacy mechanisms caused by temporal correlations in the context of continuous aggregate release.

3 Preliminaries

In this section, we introduce some preliminary definitions. The associated symbols are shown in Table 1. We use bold lowercase letters for vectors, such as **a**, and bold capital letters for matrices, such as **A**.

Definition 1 (δ-location set) [6]. δ-location set is a set containing minimum number of locations that have prior probability sum not less than $1 - \delta$,

$$\Delta X_t = min\left\{ s_i | \sum_{s_i} p_t^-[i] \geq 1 - \delta \right\} \tag{1}$$

where $\boldsymbol{p_t^-}$ is the prior probability vector of a user's location at timestamp t, the size of ΔX_t is related to the size of δ.

For example, S denote the domain of space. If we divide the space S into $\{s_1, s_2, s_3, s_4, s_5, s_6\}$, then $p_t^- = [0.1, 0.5, 0.05, 0.3, 0.03, 0.02]$ corresponds to $[s_1, s_2, s_3, s_4, s_5, s_6]$, where p_t^- represents the probability that a user appears in each

Table 1. Summary of notations.

Symbols	Symbolic meaning
s_i	The grid number of the user's real location after area grid
$u^* = (x^*, y^*)$	User's real location coordinates
$u_t^* = (x_t^*, y_t^*)$	User's real location coordinates at timestamp t
$z_t = (x_t, y_t)$	User's location after the disturbance at timestamp t
$\Pr(z_t \mid u_t^* = s_i)$	The probability of releasing z_t given $u_t^* = s_i$
$\boldsymbol{p_t^-}$	The prior probability vector at timestamp t
$p_t^-[i]$	The prior probability of the i th location of $\boldsymbol{p_t^-}$
$\boldsymbol{p_t^+}$	The posterior probability vector at timestamp t
$p_t^+[i]$	The posterior probability of the i th location of $\boldsymbol{p_t^+}$
ΔX	δ-location set
ΔX^1	The set of horizontal coordinates of the points in the δ-location set
ΔX^2	The set of vertical coordinates of points in the δ-location set
$\|\cdot\|_p$	L_p norm
$\boldsymbol{M}$	Transfer matrix
m_{ij}	The probability of a user moving from grid i to grid j

cell. When $\delta = 0.1$, $\Delta X_t = \{s_2, s_4, s_1\}$; $\delta = 0.05$, $\Delta X_t = \{s_2, s_4, s_1, s_3\}$. In special case, when $\delta = 0$, δ-location set contains all the possible locations, where s_i represents the i th cell of the grid partition.

In practice, there is a small probability that the real location is not in δ-location set. We denote this phenomenon as drift. If it happens, we handle it with a surrogate approach. The two concepts are defined as follows.

Definition 2 (Drift) [6]. A real location which is not in δ-location set is defined Drift.

Definition 3 (Surrogate) [6]. Surrogate $\overline{u} = (\overline{x}, \overline{y})$ is the closest location in ΔX to the real location $u^* = (x_t^*, y_t^*)$.

$$\overline{u} = (\overline{x}, \overline{y}) = \underset{(x_s, y_s) \in \Delta X}{\operatorname{argmin}} \; dist\big((x_s, y_s), (x_t^*, y_t^*)\big) \tag{2}$$

where $dist(\cdot)$ is the Euclidean distance.

Definition 4 (Differential privacy on δ-location set) [6]. Given a random mechanism $f : x \to z$. At timestamp t, f satisfies ε-differential privacy on δ-location set ΔX_t, if for any output z_t and any two locations x_1 and x_2, the following holds.

$$\frac{\Pr(f(x_1) = z_t)}{\Pr(f(x_2) = z_t)} \leq e^{\varepsilon} \tag{3}$$

where ε is privacy budget.

Differential privacy is achieved by adding random noise. The magnitude of the noise is influenced by the sensitivity of a query function. As we focus on protecting the user's real two-dimensional location coordinates, the sensitivity on two-dimensional is defined as follows.

Definition 5 (Sensitivity on δ-location set). Given a query function $f : \Delta X \rightarrow R$, where ΔX is δ-location set and R is the return result of the query function. The sensitivity of any two adjacent locations in ΔX is as follows.

$$\Delta f = \max_{u_1^*, u_2^* \in \Delta X} \left\| u_1^* - u_2^* \right\|_1 = \max(\Delta_1, \Delta_2) \tag{4}$$

where $\Delta X^1 = \{x^* | u^* = (x^*, y^*) \in \Delta X\}$ represents the set of x coordinates (longitudes), $\Delta X^2 = \{y^* | u^* = (x^*, y^*) \in \Delta X\}$ is the set of y coordinates (latitudes), u^* is the real location, $\Delta_1 = \max_{a,b \in \Delta X^1} |a - b|$, and $\Delta_2 = \max_{a,b \in \Delta X^2} |a - b|$.

Definition 6 (Distance). The distance between the real location $u^* = (x^*, y^*)$ and the perturbed location $z = (x, y)$ is defined as.

$$dis(u^*, z) = \left\| u^* - z \right\|_2. \tag{5}$$

$dis(u^*, z)$ can be used as an error measurement between the real location and the perturbed location.

Definition 7 (Staircase mechanism) [18]. Given a multidimensional query function $f : D \rightarrow R^d$, the Staircase mechanism is defined as:

$$M(D) = f(D) + Staircase(\Delta, \varepsilon, \gamma)^d \tag{6}$$

where $Staircase(\Delta, \varepsilon, \gamma)^d$ is taken to be random variable subjecting to Staircase distribution. Δ is sensitivity, ε is privacy budget and $\gamma \in [0, 1]$.

4 Location Release Model

4.1 Framework

We focus on the scenario of continuous location data release, in which users need to send their real-time locations to servers frequently to obtain the corresponding services during their moving. We assume that LBS providers are untrusted, and users' locations should not be directly released to the providers to protect their location privacy. The user's real location under each timestamp is treated as a sensitive location, so the real location is only visible to the user himself.

In order to protect users' location privacy under continuous timestamps, we propose a location preservation framework. If we want to release a location of a timestamp, we should construct a δ-location set for it. And then, we adopt a differential privacy mechanism to release a perturbed location.

Specifically, the framework steps are as follows.

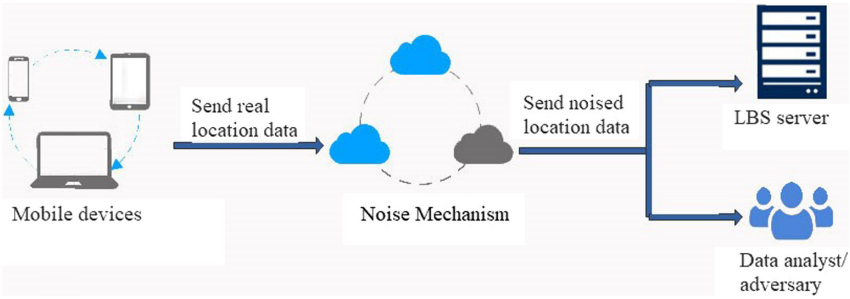


Fig. 1. Location release framework.

Step 1. Grid partition: partition the map area with grid where the data set is located.
Step 2. Calculate the transition matrix $\boldsymbol{M}$ between locations. Each $m_{ij} \in \boldsymbol{M}$ represents the probability of the user moving from grid i to grid j.
Step 3. Calculate the prior probability vector $\boldsymbol{p_t^-}$ under each timestamp, where the prior probability refers to the probability before releasing the perturbed location. $\mathcal{G}(\mathcal{G} \in \{1, 2, \ldots, i, \ldots\}$ is the number of grid cells. $\mathcal{T}_t$ is the total number of locations at timestamp t. ω_t^i is the number of locations in grid cell i. Then the prior probability vector $\boldsymbol{p_t^-}$ at timestamp t is calculated as follows:

$$\begin{cases} \boldsymbol{p}_1^- = \left[\frac{\omega_1^1}{\mathcal{T}_1}, \frac{\omega_1^2}{\mathcal{T}_1}, \ldots, \frac{\omega_1^i}{\mathcal{T}_1}, \ldots\right] & t = 1 \\ \boldsymbol{p_t^-} = \boldsymbol{p}_{t-1}^+ M & t \geq 2 \end{cases} \tag{7}$$

where $\boldsymbol{p}_{t-1}^+$ represents the posterior probability of the timestamp $t-1$.
Step 4. Calculate the posterior probability vector $\boldsymbol{p}_{t-1}^+$ under each timestamp,

$$p_{t-1}^+[i] = \Pr\left(u_{t-1}^* = s_i | z_{t-1}\right) = \frac{\Pr\left(z_{t-1} | u_{t-1}^* = s_i\right) p_{t-1}^-[i]}{\sum_j \Pr\left(z_{t-1} | u_{t-1}^* = s_j\right) p_{t-1}^-[j]} \tag{8}$$

where $p_{t-1}^+[i]$ is the ith element in $\boldsymbol{p}_{t-1}^+$. $\Pr\left(z_{t-1} | u_{t-1}^* = s_i\right)$ represents emission probability and z_{t-1} represents the perturbed location of $t-1$ timestamp.

To make it easier to understand, Table 2 shows the flow of alternate calculation of prior probability vector and posterior probability vector. The posterior probability vector $\boldsymbol{p_t^+}$ can be obtained according to the emission probability and the prior probability at timestamp t.

Table 2. Prior probability and posterior probability under each timestamp.

timestamp 1	timestamp 2	timestamp 3	⋯	timestamp $t-1$	timestamp t
p_1^-	p_1^+	p_2^+	⋯	p_{t-2}^+	p_{t-1}^+
↓(B)	↓(M)	↓(M)	⋯	↓(M)	↓(M)
p_1^+	p_2^-	p_3^-	⋯	p_{t-1}^-	p_t^-
	↓(B)	↓(B)	⋯	↓(B)	↓(B)
	p_2^+	p_3^+	⋯	p_{t-1}^+	p_t^+

4.2 Protecting Location with Staircase Mechanism

Algorithm 1 is the implementation process of the proposed framework. Step 1–3 calculates the prior probability vector at timestamp t. Steps 4–10 is the location releasing process. Specifically, when the user's location needs to be released at timestamp t, δ-location set ΔX_t is constructed and the real location is hidden in the δ-location set. If the real location $u_t^* = (x_t^*, y_t^*)$ is not in ΔX_t, we choose a surrogate location $\overline{u} = (\overline{x}, \overline{y})$ in ΔX_t. Then, we generate the perturbed location z_t through adding noise to the real location or the surrogate location. Finally, we release the z_t. Steps 11 calculates the posterior probability vector $\boldsymbol{p_t^+}$, which is subsequently used to compute the prior probability for the next timestamp $t+1$. When the location at timestamp $t+1$ needs to be released, we repeat the above process. The process of generating δ-location set and the process of noise generation are shown in details in Algorithm 2 and Algorithm 3, separately.

Algorithm 1. Location_perturbation

Input: $\varepsilon, \delta, \boldsymbol{M}, \boldsymbol{p_1^-}, u_t^* = (x_t^*, y_t^*), \Delta_1, \Delta_2, \gamma \in [0,1]$
Output: $z_t = (x_t, y_t)(t = 1, \dots, T)$
1. $\boldsymbol{p_1^-} = \left[\frac{\omega_1^1}{\mathcal{T}_1}, \frac{\omega_1^2}{\mathcal{T}_1}, \dots, \frac{\omega_1^i}{\mathcal{T}_1}, \dots\right]$
2. for t in $(2, \dots, T)$:
3. $\boldsymbol{p_t^-} = \boldsymbol{p_{t-1}^+ M}$ //Markov transition
4. Construct ΔX_t (Algorithm 2);
5. $if\ (x_t^*, y_t^*) \notin \Delta X_t\ then$
6. Compute the surrogate $(\overline{x}, \overline{y})$ according to equation (2)
7. $(x_t^*, y_t^*) = (\overline{x}, \overline{y})$
8. $end\ if$
9. $(\alpha_t, \beta_t) \leftarrow Staircase\_noise\ (\varepsilon, \Delta_1, \Delta_2, \gamma \in [0,1]))$(Algorithm 3)
10. $z_t = (x_t^* + \alpha_t, y_t^* + \beta_t)$
11. Compute the posterior probability vector $\boldsymbol{p_t^+}$ according to equation (8)
12. end for
13. $return\ z_t = (x_t, y_t)(t = 1, \dots, T)$

Algorithm 2 is the process of generating δ-location set, where δ-location set is a set containing minimum number of locations that have prior probability sum no less than $1-\delta$. Algorithm 2 input the prior probability vector $\boldsymbol{p}_t^-$ for timestamp t. The prior probability vector $\boldsymbol{p}_t^-$ is obtained by the posterior probability vector $\boldsymbol{p}_{t-1}^+$ of the timestamp $t-1$ and the Markov transition matrix $\boldsymbol{M}$, i.e., $\boldsymbol{p}_t^- = \boldsymbol{p}_{t-1}^+ \boldsymbol{M}$, where $\boldsymbol{p}_{t-1}^+$ is calculated by Eq. (8).

Algorithm 2. Generate_δ-location set

Input : D: location set, $p_t^-[i]$: the probability of each location, δ
Output : δ-location set ΔX_t

1.$sort(D, p_t^-[i])$ //Sort by probability in descending order
2.$\Delta X_t = \emptyset$
3.$p = 0$
4.$for\ x\ in\ D$:
5. $p+= p_t^-[i]$
6. $\Delta X_t = \Delta X_t + \{x\}$ // Add the ith location to location set D.
7. if $p \geq \delta$:
8. $break$
9. end if
10. end for
11.$return\ \Delta X_t$

Algorithm 3 is the process of generating random noise following the Staircase distribution for real locations. In Algorithm 3 S determines the noise symbol, G determines the interval in which the noise is located $[G\Delta, (G+1)\Delta)$, B determines the interval in which the noise is located $\left[G\Delta, (G+\gamma)\Delta\right)$ and $\left[(G+\gamma)\Delta, (G+1)\Delta\right)$($\Delta$ is sensitivity), and U contributes to the uniform appearance interval.

Algorithm 3. Staircase_noise

Input: $\varepsilon, \Delta_1, \Delta_2, \gamma \in [0,1]$
Output: (α_t, β_t)

1. Generate a $r.v.\ S$ with $\Pr[S=1] = \Pr[S=-1] = 1/2$.
2. Generate a geometric $r.v.\ G$ with $\Pr[G=i] = (1-b)b^i$ for integer $i \geq 0$, where $b = e^{-\varepsilon}$.
3. Generate a $r.v.\ U$ uniformly distributed in $[0,1]$.
4. Generate a binary $r.v.\ B$ with $\Pr[B=0] = \gamma/(\gamma + (1-\gamma)b$ and $\Pr[B=1] = (1-\gamma)b/(\gamma + (1-\gamma)b)$
5. $\alpha_t \leftarrow S((1-B)((G+\gamma U)\Delta_1) + B((G+\gamma+(1-\gamma)U)\Delta_1)$
6. $\beta_t \leftarrow S((1-B)\big((G+\gamma U)\Delta_2\big) + B((G+\gamma+(1-\gamma)U)\Delta_2)$
7. $return\ (\alpha_t, \beta_t)$

The time complexity of Algorithm 1 is $O\left(Tn^2\right)$, where n is the number of the grids, T is the number of timestamps. Step 1 is original prior probability. Step 2–2 is T-cycle about timestamps. Where step 3 calculates the prior probability at each timestamp, it takes $O(n)$. Step 4 generates δ-location set ΔX by Algorithm 2, it takes $O(nlogn)$. Steps 9–10 add random noise by Algorithm 3, it takes $O(1)$. Step 11 computes the posterior

probability vector $\boldsymbol{p}_t^+$ by Bayesian inference, it takes $O(n^2)$. So, the time complexity is $T(O(n) + O(nlogn) + O(1) + O(n^2)) = O(Tn^2)$.

4.3 Privacy Analysis

In this subsection, we focus on analyzing the privacy level of the proposed methods, and prove that it satisfies differential privacy. As the proof depends on the definition of Adversarial Privacy, we firstly introduce the definition, then formalize a theorem and prove it.

Definition 8 (Adversarial Privacy) [20]. A mechanism satisfies the ε-adversarial privacy for any location $s_i \in S$, if and only if any output z and any adversaries knowing the real location in ΔX, the following holds:

$$\frac{\Pr(u_t^* = s_i|z_t)}{\Pr(u_t^* = s_i)} \leq e^{\varepsilon} \tag{8}$$

$\Pr(u_t^* = s_i)$ is the prior probability, $\Pr(u_t^* = s_i|z_t)$ is the posterior probability for adversaries, z_t is the location of the release, $u_t^* = s_i$ is the user's real location.

Theorem 1. At any timestamp t, Algorithm 1 satisfies ε- differential privacy on δ-location set.

Proof. Algorithm 1 include construct ΔX_t and add noise on releasing locations corresponding to Algorithm 2 and Algorithm 3 respectively. Where Algorithm 2 include Algorithm 3, because Algorithm 2 need to add noise during construct ΔX_t. Therefore, prove Algorithm 1 satisfies ε- differential privacy, which is equivalent to prove Algorithm 2 satisfies ε- differential privacy.

If Algorithm 2 satisfies ε -differential privacy on 0-location set, then it also satisfies ε-differential privacy on δ-location set, since 0-location set contains all possible locations in δ-location set. Therefore, we only need to prove Algorithm 2 satisfies ε-differential privacy on 0-location set.

If $u_t^* = (x_t^*, y_t^*) \in \Delta X_t$, then the location z_t of perturbed by Algorithm 3 is released, where $z_t = (x_t^* + \alpha_t, y_t^* + \beta_t)$, (α_t, β_t) is a noise following Staircase distribution. Staircase distribution satisfies the differential privacy has been proved in [18], then

$$\frac{\Pr(u_t^* = s_i|z_t)}{\Pr(u_t^* = s_i)} \leq e^{\varepsilon_t}. \tag{10}$$

If $u_t^* = (x_t^*, y_t^*) \notin \Delta X_t$, releases $\widetilde{v}_t = (\widetilde{x}_t, \widetilde{y}_t)$ for u_t^* in ΔX_t. Then

$$\frac{\Pr(u_t^* = s_i|z_t)}{\Pr(u_t^* = s_i)} = \frac{\sum_k \Pr(u_t^* = s_i|\widetilde{x}_t = s_k)\Pr(\widetilde{x}_t = s_k|z_t)}{\sum_k \Pr(u_t^* = s_i|\widetilde{x}_t = s_k)\Pr(\widetilde{x}_t = s_k)} \leq e^{\varepsilon_t} \tag{11}$$

It has been proved in [6] that differential privacy for continuous location data release is equivalent to adversarial privacy. Therefore, Algorithm 1 satisfies the ε- differential privacy on 0-location set. That is, theorem 1 is proved.

5 Experimental Evaluation

5.1 Experimental Settings

In this section, we evaluate the performance of the proposed SM based location perturbation algorithm on Geolife data. We compare it with the LM and PIM in terms of the size of ΔX, drift ratio and distance, the precision and recall on k NN query. All of the algorithms are implemented in Python on a machine with AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx 8 CPUs 2.0 GHz and 8192 MB RAM, running Windows 10.

Geolife Data [21]**.** Geolife data was collected from 182 users over a period of more than five years (from April 2007 to August 2012). The data has a series of tuples containing latitude, longitude, and timestamps. We select the trajectories within the third ring of Beijing to calculate the Markov transition matrix with the map is divided into 0.34×0.34 km^2 cells.

Evaluation Metrics. We use size of ΔX, drift ratio and distance as experimental metrics.

Size of ΔX [6]**.** Because our privacy definition is based on ΔX, we evaluated the size of ΔX to understand how ΔX grows or changes. The smaller size of ΔX means the better experimental results, when ε is the same.

Drift Ratio [19] **.** If δ is not properly valued, the real location may be filtered out with a small probability. We denote it as drift. $Driftratio = \frac{l_t}{S_t}$ is drift ratio at timestamp t, where l_t is the number of locations that drift ratio at timestamp t, S_t is the total number of locations in ΔX at timestamp t. The smaller drift ratio means the better experimental results, when ε is the same.

Distance. The distance means Euclidean distance between the real location and the released location, which is used to measure data utility. The smaller distance means the better experimental results, when ε is the same.

5.2 Performance Over Time

To evaluate the performance of the release mechanism when a user moves under a continuous timestamp. We randomly selected a test track containing 500 timestamps from the Geolife data. We tested LM, PIM and SM at each timestamp with $\varepsilon = 1$ and $\delta = 0.01$. Each method runs 20 times and releases the average.

(1) Track Release
Figure 2(a) shows the original trajectory. Figure 2(b), 2(c), and 2(d) show the release locations under each timestamp. We can see that the released locations of SM are closer to real locations, compare with LM and PIM. It means that SM has less perturbation at each timestamp with $\varepsilon = 1$ and $\delta = 0.01$, compare with LM and PIM.

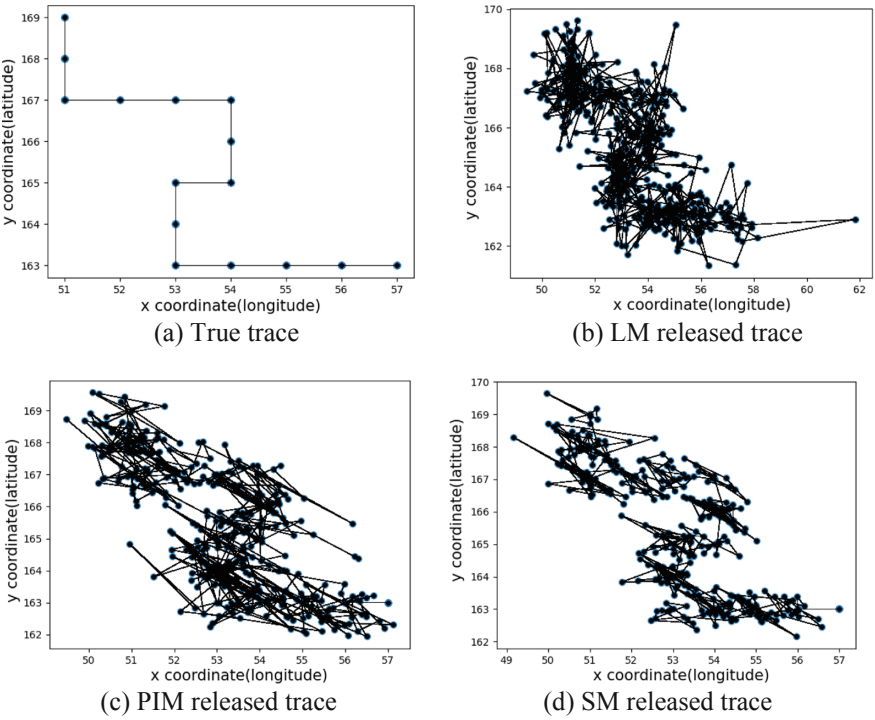


Fig. 2. Track Release: (a) True trace; (b) LM released trace; (c) PIM released trace; (d) SM released trace. ($\varepsilon = 1$ and $\delta = 0.01$)

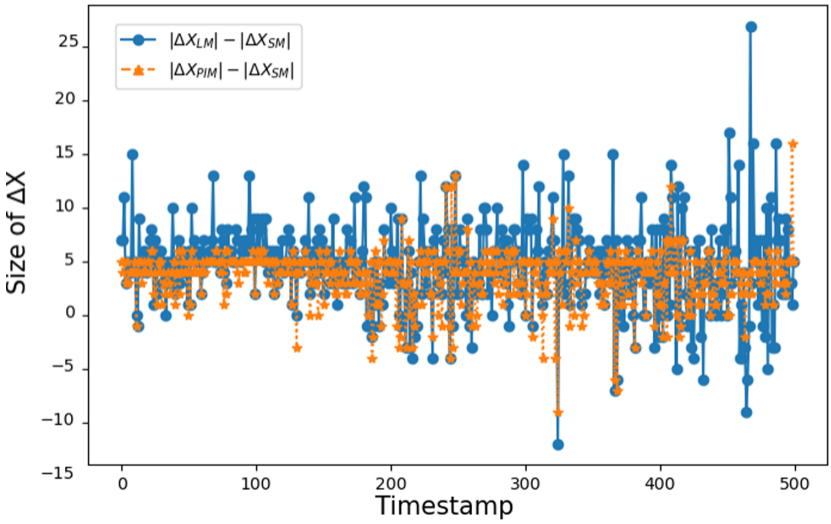


Fig. 3. Size of ΔX over time. ($\varepsilon = 1$ and $\delta = 0.01$)

(2) Size of ΔX
Figure 3 shows the change of size of ΔX over time. $|\Delta X_{LM}| - |\Delta X_{SM}|$ represents the difference between size of ΔX on the LM and size of ΔX on the SM. $|\Delta X_{PIM}| - |\Delta X_{SM}|$ shows the difference between Size of ΔX on PIM and size of ΔX on the SM. Figure 3 as a whole, the difference between size of ΔX is greater than 0. The size of ΔX generated by SM is less than that generated by LM and PIM. It shows that SM achieve the same privacy budget with less noise addition, and we can get high data utility in most case.

(3) Drift ratio
Figure 4 shows the change of drift ratio over time. $|Driftratio_{LM}| - |Driftratio_{SM}|$ represents the difference between drift ratio on the LM and drift ratio on the SM. $|Driftratio_{PIM}| - |Driftratio_{SM}|$ shows the difference between drift ratio on PIM and drift ratio on the SM. The orange line is closer to 0. It means that SM has a smaller drift ratio than LM and PIM in most case.

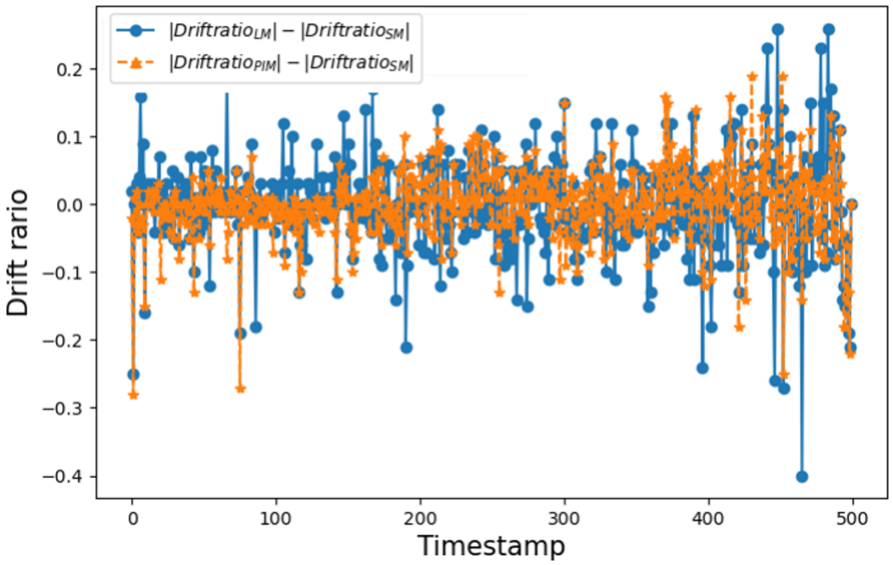


Fig. 4. Drift ratio over time. ($\varepsilon = 1$ and $\delta = 0.01$)

(4) Distance

Figure 5 shows the change in distance over time. It shows that SM's distance is less than LM's and PIM's distance. SM releases locations more accurately under each timestamp. It has the small noise, so its distance is closer to 0. SM has better data utility compared with LM and PIM under $\varepsilon = 1$ and $\delta = 0.01$. The emission probability and prior probability are more accurate at timestamp t, we can calculate the posterior probability by Eq. (8) at timestamp t. Therefore, in most case, the posterior probability distribution obtained by SM is more accurate than that obtained by LM and SM.

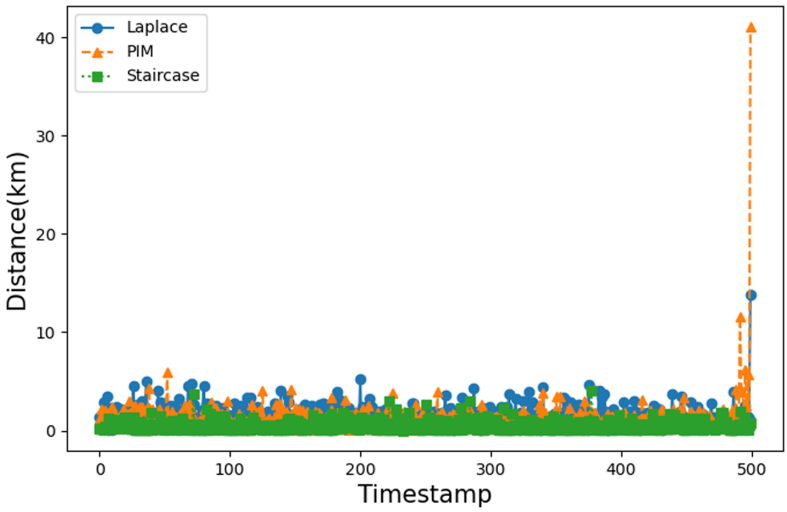
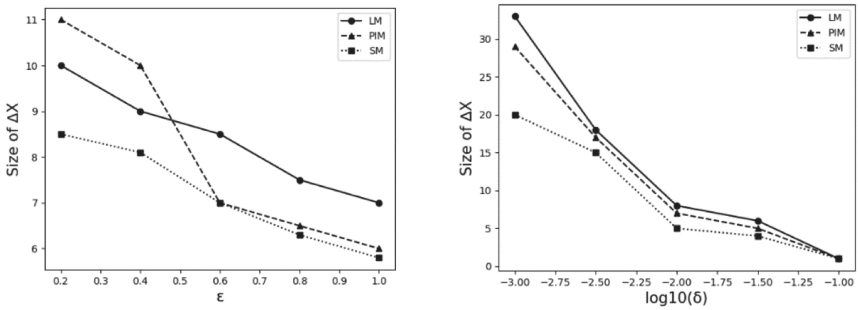


Fig. 5. Distance over time. ($\varepsilon = 1$ and $\delta = 0.01$)

5.3 Impact of Parameters

Different trajectories may have some influence on the accuracy of experiment results. We select 100 trajectories from 100 users, each containing 500 timestamps, to evaluate the overall performance of method and the impact of different parameters. In this section, Size of ΔX, Drift ratio and Distance represent the average of 100 trajectories from 100 users under 500 timestamps.



(a)Size of ΔX vs. $\varepsilon(\delta = 0.01)$ (b) Size of ΔX vs. $\delta(\varepsilon = 1)$

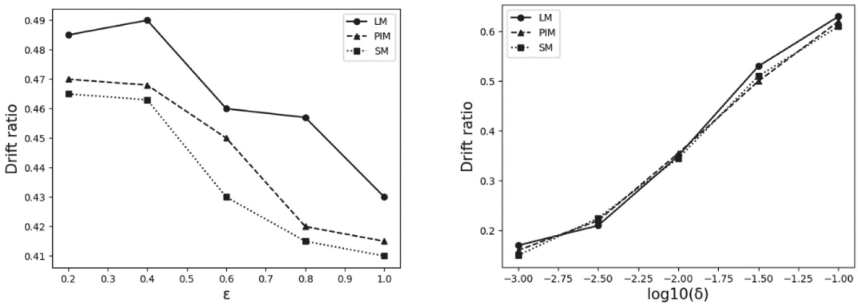
Fig. 6. Impact of ε and δ on size of ΔX.

Size of ΔX vs. ε.
Figure 6(a) demonstrates the variation of size of ΔX with the change of ε while fixing δ. It shows that when $\delta = 0.01$, the sizes of ΔX generated by the three mechanisms all decrease with the increasing of ε. Because a larger ε implies lower privacy, and a large

ε requires a small ΔX for hiding the real location. In addition, Fig. 6(a) also shows that the size of ΔX generated by SM is the smallest for each ε. That is to say that SM has the lowest data perturbation and the best data utility compared with LM and PIM under the same privacy budget ε.

Size of ΔX vs. δ.
Figure 6(b) demonstrates the variation of size of ΔX with the change of δ while fixing ε. It shows that when $\varepsilon = 1$, the sizes of ΔX generated by the three mechanisms all decrease with the increasing of δ. With δ increasing, the size of ΔX become smaller according to definition 1. In practice, δ can't be too large because ΔX should contain at least one location (real location). Therefore, we set $\delta = 0.01$ as the default value. In addition, Fig. 6(b) also shows that the size of ΔX generated by SM is smaller than LM and PIM under the same δ. It shows that SM provide higher data utility than LM and PIM in most case.

Drift Ratio vs. ε.
Figure 7(a) demonstrates the variation of drift ratio with the change of ε while fixing δ. It shows that when $\delta = 0.01$, with ε increasing, the drift ratio becomes smaller, because a large ε implies less location perturbation. In addition, Fig. 7(a) also shows that drift ratio of SM is smaller than LM and PIM under the same ε. It shows that SM provides higher data utility than LM and PIM in most case.



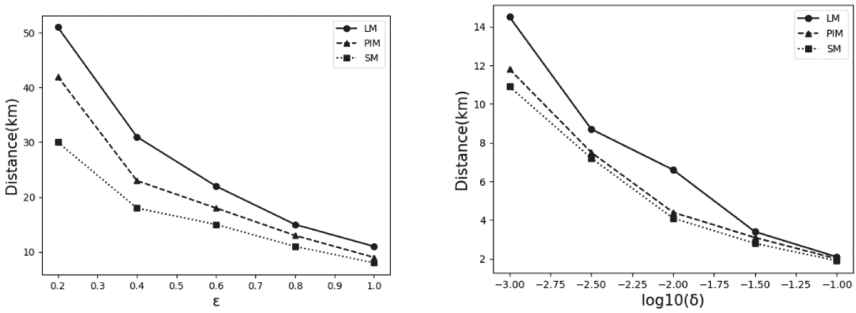
(a)Drift ratio vs. $\varepsilon(\delta = 0.01)$ (b) Drift ratio vs. $\delta(\varepsilon = 1)$

Fig. 7. Impact of ε and δ on drift ratio.

Drift Ratio vs. δ.
Figure 7(b) demonstrates the variation of drift ratio with the change of δ while fixing ε. It shows that when $\varepsilon = 1$, the drift ratio increases with δ increasing, because with δ increasing, the size of ΔX decreases. In addition, Fig. 7(b) also shows that drift ratio of SM is smaller than LM and PIM under the same ε. It shows that SM provides higher data utility than LM and PIM in most case.

Distance vs. ε.
Figure 8(a) demonstrates the variation of distance with the change of ε while fixing δ.

It shows that when $\delta = 0.01$, the distance decreases with ε increasing, because with ε increasing, the size of ΔX is smaller. At this time, the location in ΔX is closer, so the distance is smaller. In addition, Fig. 8(a) shows that the distance of SM is smaller than LM and PIM under the same ε. It shows that SM provide higher data utility than LM and PIM in most case.



(a)Distance vs. ε ($\delta = 0.01$) (b) Distance vs. $\delta(\varepsilon = 1)$

Fig. 8. Impact of ε and δ on distance.

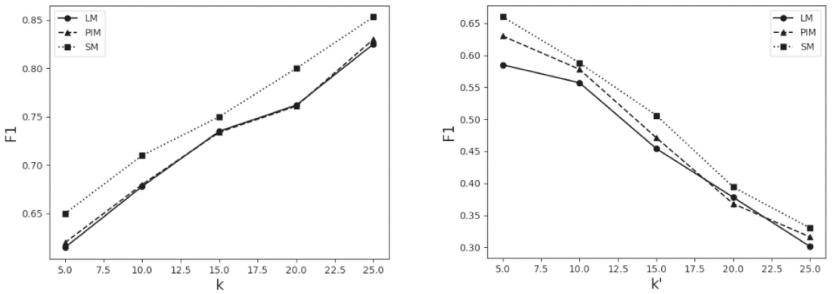
Distance vs. δ.
Figure 8(b) demonstrates the variation of distance with the change of δ while fixing ε. It shows that when $\varepsilon = 1$, the distance decreases with δ increasing. Because with δ increasing, the size of ΔX is smaller. The rest of the location in ΔX is closer. Both the real location and the release location are in ΔX, so the distance is smaller. In addition, SM has a smaller distance and better data utility than LM and PIM under same δ. It shows that SM provide higher data utility than LM and PIM in most case.

5.4 Utility for Location Based Queries

In order to verify the utility of released locations, we used k NN query on 100 trajectories which has 500 timestamps to obtain the query precision and recall under each timestamp. We use k NN on the original trajectories and k'NN on the release trajectories, when $\varepsilon = 1$, $\delta = 0.01$.

Figure 9(a) shows that when $k = k'$(Precision = Recall), the F1 increases with $k\left(k'\right)$ increasing. Because the scope of the query will be expanded, the neighboring locations will be queried over a larger area. In addition, SM has better F1 than LM and PIM. It shows that SM provide more accurate query results in most case.

Figure 9(b) shows that when $k = 5$, SM's F1 is bigger than LM and PIM. It shows that SM is more accurate than LM and PIM.



(a)F1(k=k') (b)F1

Fig. 9. k NN results: (a) F1 under $k = k'$; (b) F1 vs k'.

5.5 Running Time

Figure 10 shows the running time of LM, PIM, and SM with continuous timestamps. It shows that when $\delta = 0.01$ and $\varepsilon = 1$, the running time for three mechanisms all increase with the large timestamp. Because if we need to calculate the prior probability and the posterior probability timestamp t, we need to calculate the prior probability and the posterior probability of the preceding timestamp $t - 1$. Compared with the LM and PIM, SM has a shorter running time.

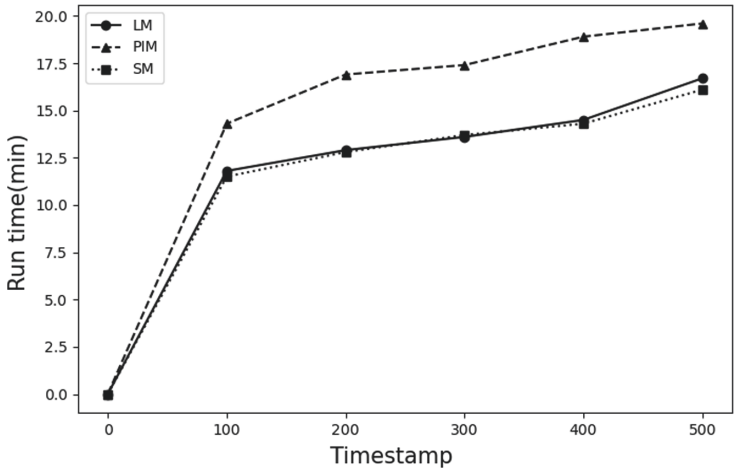


Fig. 10. Run time over time. ($\varepsilon = 1$ and $\delta = 0.01$)

6 Conclusion

A differential privacy location preservation method based on Staircase mechanism is proposed to solve the problem of location privacy preservation under temporal correlations. Considering the temporal correlation between locations, this method protects the

privacy of the user's real location in each timestamp, making it difficult for the adversary to infer the user's real location from the released location information. Experiments show that SM has good utility while achieving the effect of location differential privacy.

However, this method does not take into account the privacy preservation level of the user's location, and the default location posted by the user is a strong sensitive location. The further work will consider the appropriate privacy level allocation of users' location, and more privacy budget should be allocated to the highly sensitive location, so that it can be fully protected. For weakly sensitive locations, less privacy budget is allocated to improve the availability of location data on the premise of satisfying differential privacy preservation.

Acknowledgment. The authors would also like to appreciate the anonymous reviewers for their valuable suggestions, which lead to a substantial improvement of this paper. This research has been funded by the National Natural Science Foundation of China (Grant No. 61672468, 61702148).

References

1. Primault, V., Boutet, A., Mokhtar, S.B., Brunie, L.: The long road to computational location privacy: a survey. IEEE Commun, Surv. Tutorials **21**(3), 2772–2793 (2019). https://doi.org/10.1109/COMST.2018.2873950
2. Tan, Z., Wang, C., Yan, C., Zhou, M., Jiang, C.: Protecting privacy of location-based services in road networks. IEEE Trans. Intell. Transport. Syst. 14 (2020). https://doi.org/10.1109/TITS.2020.2992232
3. Chatzikokolakis, K., Elsalamouny, E., Palamidessi, C., Pazii, A.: Methods for location privacy: a comparative overview. Found. Trends® Privacy Secur. **1**(4), 199–257 (2017). https://doi.org/10.1561/3300000017
4. Wei, J., Lin, Y., Yao, X., Zhang, J.: Differential privacy-based location protection in spatial crowdsourcing. IEEE Trans. Serv. Comput. 1 (2019). https://doi.org/10.1109/TSC.2019.2920643
5. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
6. Xiao, Y., Xiong, L.: Protecting locations with differential privacy under temporal correlations. In: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, pp. 1298–1309 (2015). https://doi.org/10.1145/2810103.2813640
7. Machanavajjhala, A., Kifer, D., Abowd, J., Gehrke, J., Vilhuber, L.: Privacy: theory meets practice on the map. In: 2008 IEEE 24th International Conference on Data Engineering, pp. 277–286, April 2008. https://doi.org/10.1109/ICDE.2008.4497436
8. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: differential privacy for location-based systems. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 901–914, New York, NY, USA (2013). https://doi.org/10.1145/2508859.2516735
9. Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Optimal geo-indistinguishable mechanisms for location privacy. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS 2014, Scottsdale, Arizona, USA, pp. 251–262 (2014). https://doi.org/10.1145/2660267.2660345

10. Niu, B., Chen, Y., Wang, Z., Li, F., Wang, B., Li, H.: Eclipse: preserving differential location privacy against long-term observation attacks. IEEE Trans. Mob. Comput. 1 (2020). https://doi.org/10.1109/TMC.2020.3000730
11. Gursoy, M.E., Liu, L., Truex, S., Yu, L.: Differentially private and utility preserving publication of trajectory data. IEEE Trans. Mob. Comput. **18**(10), 2315–2329 (2019). https://doi.org/10.1109/TMC.2018.2874008
12. Zhu, T., Xiong, P., Li, G., Zhou, W.: Correlated differential privacy: hiding information in non-IID data set. IEEE Trans. Inf. Forensics Secur. **10**(2), 229–242 (2015). https://doi.org/10.1109/TIFS.2014.2368363
13. Liu, C., Chakraborty, S., Mittal, P.: Dependence Makes You Vulnerable: Differential Privacy Under Dependent Tuples. In: 23rd Annual Network and Distributed System Security Symposium, NDSS 2016, p. 15. CA, USA, San Diego (Feb. 2016)
14. Yang, B., Sato, I., Nakagawa, H.: Bayesian differential privacy on correlated data. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, May 2015, pp. 747–762 (2015). https://doi.org/10.1145/2723372.2747643
15. Cao, Y., Yoshikawa, M., Xiao, Y., Xiong, L.: Quantifying differential privacy in continuous data release under temporal correlations. IEEE Trans. Knowl. Data Eng. **31**(7), 1281–1295 (Jul. 2019). https://doi.org/10.1109/TKDE.2018.2824328
16. Cao, Y., Yoshikawa, M., Xiao, Y., Xiong, L.: Quantifying differential privacy under temporal correlations. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), San Diego, CA, USA, April 2017, pp. 821–832 (2017). https://doi.org/10.1109/ICDE.2017.132
17. Dwork, C., Naor, M., Pitassi, T., et al.: Differential privacy under continual observation. Stoc 715–724 (2010)
18. Geng, Q., Kairouz, P., Oh, S., Viswanath, P.: The staircase mechanism in differential privacy. IEEE J. Sel. Topics Sig. Process. **9**(7), 1176–1184 (2015). https://doi.org/10.1109/JSTSP.2015.2425831
19. Geng, Q., Viswanath, P.: The optimal mechanism in differential privacy. In: International Symposium on Information Theory, pp. 2371–2375 (2014)
20. Chen, R., Fung, B.C.M., Desai, B.C.: Differentially private trajectory data publication. arXiv Preprint, arXiv: 1112.2020 (2011)
21. Yu, Z., Zhang, L., Xie, X., Ma, W.-Y.: Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of International Conference on World Wild Web (WWW 2009), Madrid, Spain, pp. 791–800. ACM Press (2009)