A Case Study: Privacy Preserving Release of Spatio-temporal Density in Paris

Gergely Acs, Claude Castelluccia

A Case Study: Privacy Preserving Release of Spatio-temporal Density in Paris

Gergely Acs
INRIA
gergely.acs@inria.fr

Claude Castelluccia
INRIA
claude.castelluccia@inria.fr

ABSTRACT

With billions of handsets in use worldwide, the quantity of mobility data is gigantic. When aggregated they can help understand complex processes, such as the spread viruses, and built better transportation systems, prevent traffic congestion. While the benefits provided by these datasets are indisputable, they unfortunately pose a considerable threat to location privacy.

In this paper, we present a new anonymization scheme to release the spatio-temporal density of Paris, in France, i.e., the number of individuals in 989 different areas of the city released every hour over a whole week. The density is computed from a call-data-record (CDR) dataset, provided by the French Telecom operator Orange, containing the CDR of roughly 2 million users over one week. Our scheme is differential private, and hence, provides provable privacy guarantee to each individual in the dataset. Our main goal with this case study is to show that, even with large dimensional sensitive data, differential privacy can provide practical utility *with* meaningful privacy guarantee, if the anonymization scheme is carefully designed. This work is part of the national project XData (`http://xdata.fr`) that aims at combining large (anonymized) datasets provided by different service providers (telecom, electricity, water management, postal service, etc.).

1. INTRODUCTION

Mobile phone datasets have become widely available in recent years and have opened the possibility to improve our understanding of large-scale social networks by investigating how people exchange information, interact, and develop social interactions. While the benefits provided by these datasets are indisputable, they unfortunately pose a considerable threat to location privacy. Not only this can impact people lives negatively, this also affects research. Because privacy is so important to people, companies and researchers are reluctant to publish mobile phone datasets by fear of being held responsible for potential privacy breaches. It is therefore urgent to develop *practical* tools for private releases and analysis of mobility datasets.

This paper focuses on applications that only need location counts, i.e., the repartition of visitors on a map at a given period of time. This information can typically be published by dividing a map into cells, and then releasing the count (i.e., number of users) associated to each of the cells. However, it is important to ensure that this publication does not leak any information about the mobility patterns of individual users, which might be trivial when, for example, the count values are small.

Why Differential Privacy? Privacy has different definitions and different models have been proposed. In this paper, we use *differential privacy* [10] which has emerged as a compelling privacy model. The advantage of this model, compared to the many others proposed in the literature, is two-fold. First, it provides a formal and measurable privacy guarantee regardless what other background information or sophisticated inference technique the adversary uses even in the future. Second, it is closed with respect to sequential and parallel composition, i.e., the result of the sequential or parallel combination of two differential private algorithms is also differential private.

This has particular importance in practice, since it does not only simplify the design of anonymization solutions, but also allows to measure how much privacy remains when a given dataset, or a set of correlated datasets, is anonymized (and released) several times, possibly by different entities.

Differential-private schemes often requires adding noise to the published data (e.g., to the published location counts) so that it leaks almost no information about any participating individual, but still reveals aggregated information about the population as a whole. The variance of the noise is calibrated to both the *sensitivity* of the counts (i.e., the maximal change of the counts due to the inclusion/exclusion of a single record in a dataset) and a desired privacy level ε. For large-dimensional data, such as temporal density, the sensitivity is usually so large that the added noise is much larger than the actual density count values for stringent privacy requirement (i.e. $\varepsilon < 1$). Consequently, the resultant anonymized data is often meaningless.

Contributions. In this paper, we show that, for a given privacy level (i.e., a given ε), the magnitude of noise can be substantially reduced by using several optimizations and by customizing the anonymization mechanisms to the public characteristics of datasets and applications. We observe that the temporal density, of each cell, can be characterized by a periodic time series. This is explained by the fact that

KDD'14, August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623361.

aggregated mobility patterns are quite periodic. Moreover these time series follow very similar trends, as most people in nearby cells have similar calling patterns. As a consequence, time series can be compressed by sampling, clustering and low-pass filtering. Sampling and clustering reduce data sensitivity, which results in lower added noise and better performance. We further attenuate the distortion resulting from the compression and perturbation phases via novel optimization and post-processing algorithms. We show experimentally that the achieved performance is quite high and that differential privacy can be practical in real-world applications. However, we believe that there are probably no differential private anonymization techniques that fit all applications, and that anonymization algorithms have to be customized to each application and dataset.

The XData project. This work was done in the context of the French XData project. XData is a national funded project under the "Big Data program", whose goal is to study the benefit of combining and cross-processing different types of datasets provided by various service providers (such as Orange, Electricité de France, La Poste, etc.). However, according to the European Data Protection laws (Directive 95/46/EC), all datasets have to be anonymized, prior cross-processing, such that data subjects are no longer identifiable. The law does not dictate any specific privacy model, but stipulates that "*to determine whether a person is identifiable, account should be taken of all the means likely reasonably to be used either by the controller or by any other person to identify the said person*". We believe that the best existing model to achieve this goal is probably differential privacy. This paper shows how to anonymize the mobility data, provided by Orange, under the differential privacy model. In particular, we show how to release density information. Geographical density is useful in many of applications envisioned by the XData projet, such as identifying areas where to install new businesses or build new infrastructures.

2. RELATED WORK

Several recent studies have demonstrated the privacy risks of releasing location data by re-identifying individuals from geospatial data sets [8, 28, 13]. As a result, a plethora of privacy-preserving techniques have been introduced, however, most of them do not provide any formal privacy guarantee (see [6] and the references therein).

Differential privacy (DP) was first rigorously presented in [10] with the Laplace mechanism (LPA) as a first generic tool to guarantee DP. There exist two relaxations of ε-DP; (ε,δ)-probabilistic DP [20] and (ε, δ)-indistinguishability [9]. The former guarantees ε-DP with high probability ($\geq 1-\delta$), while the latter relaxes the bound of ε-DP.

Location privacy has also been addressed in the context of DP. [5] applies DP to location, and more generally, sequential data release. However, this scheme does not release time information apart from the sequentiality of locations. Another recent work [2] formalizes location privacy as protecting the users' location within a radius r with a level of privacy that depends on r. This corresponds to a generalized version of DP. They target LBS applications and add noise directly to users' GPS coordinates. Commuting patterns in U.S. were anonymized in the probabilistic DP model in [20]. This scheme has also been deployed in practice within the project called OnTheMap by the U.S. Census Bureau. Other authors [7, 19] apply different spatial decomposition techniques to partition the two dimensional domain into cells, and then obtain noisy counts for each cell. However, these techniques are not concerned with releasing multiple counts over time.

Several DP techniques have been proposed to release time series. In [12], the authors propose a framework to release real-time aggregate statistics under DP based on filtering and adaptive sampling. Some other proposals [11, 4] provide a weaker guarantee on continuous data streams; they provide event-level privacy to protect an event (i.e., one user's presence at a particular time point), rather than the presence of that user. As all these works address the *real-time* (interactive) release of aggregates, they are usually less accurate than *off-line* (non-interactive) approaches, which can access the whole time series and build more accurate models for perturbation.

The most related work to ours is [24, 17, 23]. All of them address the off-line release of time series under DP. Rastogi and Nath [24] proposed a Discrete Fourier Transform (DFT) based algorithm which guarantees DP by perturbing the discrete Fourier coefficients. This technique was further improved for DP histogram release in [1]. We reuse the improved private DFT algorithm from [1], and further improve its accuracy at the cost of some privacy. In [17], the time series are pre-processed by pre-sampling and smoothing them via averaging. These techniques diminish the global sensitivity of the data, and thereby allows to lower the injected Laplace noise. We also use a similar sampling technique to [17] in order to compress time series. However, [17] is a more general solution and targets even non-periodic time series where averaging may be a better perturbation model than DFT. As aggregated location counts are typically periodic, DFT is a more accurate perturbation model for our application. Finally, DP-WHERE [23] adds noise to the set of empirical probability distributions which is derived from the original CDR datasets, and samples from these distributions to generate synthetic CDRs. Instead of releasing the whole CDR dataset, we only aim at releasing the temporal density of IRIS cells which is a more specific problem. This mapping between CDRs and IRIS cells influences utility and is not considered in [23].

3. PRELIMINARIES

3.1 Differential Privacy

Intuitively, differential privacy [10] (DP) requires that the outcome of any computation be insensitive to the change of a single record in the dataset. Consequently, for a record owner, it means that any privacy breach will not be due to participating in the database.

Definition 1 (Differential Privacy) *A privacy mechanism $\mathcal{A}$ gives ε-differential privacy if for any database $\mathcal{D}_1$ and $\mathcal{D}_2$ differing on at most one record, and for any possible output $O \in Range(\mathcal{A})$,*

$$e^{-\varepsilon} \times Pr[\mathcal{A}(\mathcal{D}_2) = O] \leq Pr[\mathcal{A}(\mathcal{D}_1) = O] \leq e^{\varepsilon} \times Pr[\mathcal{A}(\mathcal{D}_2) = O]$$

where the probability is taken over the randomness of $\mathcal{A}$.

A relaxation of DP is probabilistic-DP [20], where privacy breaches may occur with very small probability.

Definition 2 (Probabilistic Differential Privacy [20]) *A privacy mechanism $\mathcal{A}$ gives (ε, δ)-probabilistic differential privacy if for any database $\mathcal{D}_1$ and $\mathcal{D}_2$ differing on at most one record, and for any possible output $O \in Range(\mathcal{A})$, we can partition the output space Ω into Ω_1 and Ω_2 such that (1) for all $O \in \Omega_1$,*

$$e^{-\varepsilon} \times Pr[\mathcal{A}(\mathcal{D}_2) = O] \leq Pr[\mathcal{A}(\mathcal{D}_1) = O] \leq e^{\varepsilon} \times Pr[\mathcal{A}(\mathcal{D}_2) = O]$$

and (2) for any database $\mathcal{D}$, $Pr[\mathcal{A}(\mathcal{D}) \in \Omega_2] \leq \delta$ where the probability is taken over the randomness of $\mathcal{A}$.

The latter definition guarantees that algorithm $\mathcal{A}$ achieves DP with high probability ($\geq 1 - \delta$), and the set Ω_2 contains all outputs that are privacy breaches according to Definition 1. The probability of these outputs are bounded by δ. Notice that with $\delta = 0$ probabilistic DP boils down to Definition 1. Although probabilistic DP has weaker privacy guarantee than Definition 1, it provides higher utility in practice.

The definition of differential privacy enjoys the property of sequential composition, which specifies the privacy guarantee in a sequence of computation.

Theorem 1 ([21]) *Let $\mathcal{A}_i$ each provide ε_i-differential privacy. A sequence of $\mathcal{A}_i(\mathcal{D})$ over the dataset $\mathcal{D}$ provides $\sum_i \varepsilon_i$-differential privacy.*

3.2 Differential Private Mechanisms

Three principal techniques for achieving (probabilistic) DP are *Laplace mechanism* [10] (LPA), *Gaussian mechanism* [15] (GPA), and *Exponential mechanism* [22]. A fundamental concept of all these techniques is the *global sensitivity* of a function [10]:

Definition 3 (Global Sensitivity) *For any function $f : \mathcal{D} \to \mathbb{R}^d$, the sensitivity of f is $\Delta f = \max_{\mathcal{D}_1, \mathcal{D}_2} ||f(\mathcal{D}_1) - f(\mathcal{D}_2)||_1$ for all $\mathcal{D}_1, \mathcal{D}_2$ differing in at most one record.*

The global sensitivity is also called as L_1-sensitivity due to the L_1-norm used in its definition and is denoted by $\Delta_1 f$. Similarly, the L_2-sensitivity $\Delta_2 f$ of a function f, which is used later in this paper, is defined by the L_2-norm $|| \cdot ||_2$.

Laplace mechanism (LPA). A standard mechanism to achieve differential privacy is to add Laplace noise to the true output of a function. The noise is generated according to a Laplace distribution with the probability density function (PDF) $p(x|\lambda) = \frac{1}{2\lambda} e^{-|x|/\lambda}$, where λ is calibrated as follows.

Theorem 2 ([10]) *For any function $f : \mathcal{D} \to \mathbb{R}^d$, the mechanism $\mathcal{A}$*

$$\mathcal{A}(\mathcal{D}) = f(\mathcal{D}) + \langle \mathcal{L}_1(\lambda), \ldots, \mathcal{L}_d(\lambda) \rangle$$

gives ε-differential privacy, if $\mathcal{L}_i(\lambda)$ are i.i.d Laplace variables with scale parameter $\lambda = \Delta f / \varepsilon$.

Gaussian mechanism (GPA). An alternative technique to achieve probabilistic DP is to add Gaussian noise to the true output of a function. The noise is generated according to the Gaussian distribution with the PDF $p(x|\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}$.

Theorem 3 ([15]) *For any function $f : \mathcal{D} \to \mathbb{R}^d$, the mechanism $\mathcal{A}$*

$$\mathcal{A}(\mathcal{D}) = f(\mathcal{D}) + \langle \mathcal{G}_1(\sigma), \ldots, \mathcal{G}_d(\sigma) \rangle$$

gives (ε, δ)-probabilistic differential privacy for any $\varepsilon \leq 1$ and $\sigma^2 \geq 2(\Delta_2 f)^2 \ln(4/\delta)/\varepsilon^2$, where $\mathcal{G}_i(\sigma)$ are i.i.d Gaussian variables with variance σ^2.

Assuming identical values of ε, a Gaussian random variable is more concentrated around 0 than a Laplace random variable thereby ensuring better utility for GPA. However, this larger accuracy also entails weaker privacy, since there is a small probability δ that ε-DP will not hold.

Exponential mechanism. The exponential mechanism [22] captures all DP mechanisms with a measurable output space. In particular, it assigns exponentially greater probabilities of being selected to outputs of higher utility so that the final output would be close to the optimum yet still differential private.

Theorem 4 ([22]) *Given a utility function $u : (\mathcal{D} \times \mathcal{R}) \to \mathbb{R}$ for a database $\mathcal{D}$, the mechanism $\mathcal{A}$,*

$$\mathcal{A}(\mathcal{D}, u) = \left\{ return\ r\ with\ probability \propto \exp\left(\frac{\varepsilon u(\mathcal{D}, r)}{2\Delta u}\right) \right\}$$

gives ε-differential privacy, where $\Delta u = \max_{\forall r, \mathcal{D}_1, \mathcal{D}_2} |u(\mathcal{D}_1, r) - u(\mathcal{D}_2, r)|$.

3.3 Utility Metrics

The error between the private and original time series is measured by the following metrics. Consider counts $\mathbf{X} = \{X_0, \ldots, X_{n-1}\}$. We denote the original time series by $\mathbf{X}$, the sanitized one by $\hat{\mathbf{X}}$.

Mean Relative Error (MRE): $\text{MRE}(\mathbf{X}, \hat{\mathbf{X}}) = (1/n) \sum_{i=0}^{n-1} \frac{|\hat{X}_i - X_i|}{\max(\gamma, X_i)}$, where the sanity bound γ mitigates the effect of very small counts. Following the convention [27], we adjust γ to 0.1% of $\sum_{i=0}^{n-1} X_i$.

Pearson Correlation (PC): $\text{PC}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{\sum_{i=0}^{n-1} (X_i - \sum_i X_i/n)(\hat{X}_i - \sum_i \hat{X}_i/n)}{\sqrt{\sum_{i=0}^{n-1} (X_i - \sum_i X_i/n)^2} \sqrt{\sum_{i=0}^{n-1} (\hat{X}_i - \sum_i \hat{X}_i/n)^2}}$. PC measures the linear correlation between the noisy and the original time series (i.e., whether they have similar trends), and it always falls between -1 and 1.

4. PROBLEM DEFINITION

Our goal is to release the spatio-temporal density of 989 non-overlapping areas in Paris, called IRIS cells. Each cell is defined by INSEE[1] and covers about 2000 inhabitants. The set of all IRIS cells is denoted by $\mathbb{L}$ henceforth, and are depicted in Figure 1 based on their contours[2].

We aim to release the number of all individuals who visited a specific IRIS cell in each hour over a whole week. Since human mobility trajectories exhibit a high degree of temporal and spatial regularity [14], one week long period should be sufficient for most practical applications. Therefore, we are interested in the time series $\mathbf{X}^L = \langle X_0^L, X_1^L, \ldots, X_{167}^L \rangle$ of any IRIS cell $L \in \mathbb{L}$, where X_t^L denotes the number of individuals at L in the $(t+1)$th hour of the week, such that any single individual can visit a tower only once in an hour.

[1]National Institute of Statistics and Economics: `http://www.insee.fr/fr/methodes/default.asp?page=zonages/iris.htm`

[2]Available on IGN's website (National Geographic Institute): `http://professionnels.ign.fr/contoursiris`

We will omit t and L in the sequel, if they are unambiguous in the given context. $\mathbf{X}^{\mathbb{L}}$ denotes the set of time series of all IRIS cells in the sequel.

4.1 Dataset

To compute these time series, we use a CDR (Call Data Record) dataset provided by the French telecom company Orange[3], where $\mathbb{T}$ represents the set of cell towers of the operator, and a cell tower $T \in \mathbb{T}$ is visited by an individual at time t, if the operator has a recorded event at time t at tower T related to the individual. An event can be an incoming/outgoing call or message to/from the individual. This dataset contains the events of $N = 1,992,846$ users at $|\mathbb{T}| = 1303$ towers within the administrative region of Paris (i.e., the union of all IRIS cells) over a single week (10/09/2007 - 17/09/2007). Within this interval, the average number of events per user is 13.55 with a standard deviation of 18.33 (assuming that an individual can visit any tower cell only once in an hour) and with a maximum at 732. The set of all events related to an individual constitute his/her record (trajectory) in the dataset. Similarly to IRIS cells, we can create another set of time series $\mathbf{X}^{\mathbb{T}}$, where X_t^T denotes the number of visits of tower T in the $(t+1)$th hour of the week.

4.2 Computing the time series of IRIS cells

We map the counts in $\mathbf{X}^{\mathbb{T}}$ to $\mathbf{X}^{\mathbb{L}}$ as follows. First, we compute the Voronoi tesselation of the towers cells $\mathbb{T}$ which is shown in Figure 1. Then, we calculate the count of each IRIS cell in each hour from the counts of its overlapping tower cells; each tower cell contributes with a count which is proportional to the size of the overlapping area. More specifically, if an IRIS cell L overlaps with tower cells $\{T_1, T_2, \ldots, T_c\}$, then

$$X_t^L = \sum_{i=1}^{c} X_t^{T_i} \times \frac{\mathsf{size}(T_i \cap L)}{\mathsf{size}(T_i)} \qquad (1)$$

at time t.

The rationale behind this mapping is that users are usually registered at the geographically closest tower at any time. We acknowledge that this mapping algorithm might sometimes be incorrect, since the real association of users and towers depends on several other factors such as signal strength or load-balancing. Nevertheless, without more details of the cellular network beyond the towers' GPS position, we are not aware of any better mapping technique.

5. PRIVACY PRESERVING RELEASE OF SPATIO-TEMPORAL DENSITY

Our aim is to transform the time series of all IRIS cells $\mathbf{X}^{\mathbb{L}}$ to a sanitized version $\hat{\mathbf{X}}^{\mathbb{L}}$ such that $\hat{\mathbf{X}}^{\mathbb{L}}$ satisfies Definition 1. That is, the distribution of $\hat{\mathbf{X}}^{\mathbb{L}}$ will be insensitive (up to ε) to all the visits of any single user during the whole week, meanwhile the error between $\hat{\mathbf{X}}^{\mathbb{L}}$ and $\mathbf{X}^{\mathbb{L}}$ is small.

Our sanitization algorithm is sketched in Algorithm 1. First, the input dataset is pre-sampled such that only ℓ visits are retained per user (Line 1). This ensures that the global sensitivity of all the time series (i.e., $\mathbf{X}^{\mathbb{L}}$) is no more than ℓ. Then, the pre-sampled time series of each IRIS cell is computed from that of the tower cells using Voronoi-tesselation and Formula (1) (Line 2). After the largest cells, which cover the whole city and also have large counts, from $\mathbb{T} \cup \mathbb{L}$ are identified (Line 3), their time series are perturbed to guarantee privacy (Line 4). In order to mitigate the distortion of the previous steps, we apply smoothing on the perturbed time series as a post-processing step (Line 5). Finally, the time series of all IRIS cells are computed from the post-processed time series in $\mathbb{C}$ (Line 6).

[3] `http://www.orange.com`

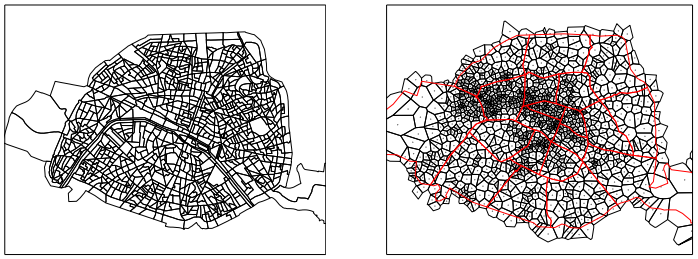


Figure 1: IRIS cells of Paris (left) and Voronoi-tesselation of tower cells (right)

Algorithm 1 Our sanitization scheme

Input: $\mathbf{X}^{\mathbb{T}}$ - input time series (from CDR), (ε, δ)-privacy parameters, $\mathbb{L}$ - IRIS cells, ℓ - maximum visits per user
Output: Noisy time series $\hat{\mathbf{X}}^{\mathbb{L}}$

1: Create $\overline{\mathbf{X}}^{\mathbb{T}}$ by sampling at most ℓ visits per user from $\mathbf{X}^{\mathbb{T}}$ (Section 5.1)
2: Compute the IRIS time series $\overline{\mathbf{X}}^{\mathbb{L}}$ from $\overline{\mathbf{X}}^{\mathbb{T}}$ (Section 4.2)
3: Compute the minimum cover $\mathbb{C} \subseteq \mathbb{T} \cup \mathbb{L}$ and the corresponding time series $\overline{\mathbf{X}}^{\mathbb{C}}$ (Section 5.2)
4: Perturb $\overline{\mathbf{X}}^{\mathbb{C}}$ into $\hat{\mathbf{X}}^{\mathbb{C}}$ (Section 5.3) `//see Algorithm 2`
5: Apply smoothing on $\hat{\mathbf{X}}^{\mathbb{C}}$ (Section 5.4)
6: Compute $\hat{\mathbf{X}}^{\mathbb{L}}$ from $\hat{\mathbf{X}}^{\mathbb{C}}$ using Formula (1)

5.1 Pre-sampling

To perturb the time series of all IRIS cells, we first have to compute their sensitivity, i.e., $\Delta_1(\mathbf{X}^{\mathbb{L}})$. To this end, we first need to calculate the sensitivity of the time series of all tower cells, i.e., $\Delta_1(\mathbf{X}^{\mathbb{T}})$. Indeed, Formula (1) does not change the L_1-sensitivity of tower counts, and hence, $\Delta_1(\mathbf{X}^{\mathbb{T}}) = \Delta_1(\mathbf{X}^{\mathbb{L}})$.

$\Delta_1(\mathbf{X}^{\mathbb{T}})$ is given by the maximum *total* number of (tower) visits of a single user in *any* input dataset. This upper bound must universally hold for all possible input datasets, and is usually on the order of few hundreds; recall that the maximum number of visits per user is 732 in our dataset. This would require excessive noise to be added in the perturbation phase. We instead follow a different approach and divide the whole sanitization process into two main steps. We first perturb a pre-sample of our dataset which better withstands perturbation, and then mitigate the distortion effect of sampling in a post-processing step described later in Section 5.4.

In particular, we truncate each record of any input dataset by considering at most one visit per hour for each user, and then select at most ℓ of such visits per user uniformly at random over the whole week. This implies that a user can

contribute with at most ℓ to all the counts in total regardless of the input dataset, and hence, the L_1-sensitivity of the dataset always becomes ℓ. The pre-sampled dataset is denoted by $\overline{\mathbf{X}}$, and $\Delta_1(\overline{\mathbf{X}}^{\mathbb{T}}) = \Delta_1(\overline{\mathbf{X}}^{\mathbb{L}}) = \ell$.

5.2 Computing the largest covering cells

To sanitize $\overline{\mathbf{X}}^{\mathbb{L}}$, there are two basic (naive) approaches. First, we can directly perturb the IRIS counts $\overline{\mathbf{X}}^{\mathbb{L}}$ by applying the Laplace mechanism: $\hat{X}_t^L = \overline{X}_t^L + \mathcal{L}(\ell/\varepsilon)$ for all $L \in \mathbb{L}$. Alternatively, we can first perturb the tower counts $\overline{\mathbf{X}}^{\mathbb{T}}$ to obtain $\hat{\mathbf{X}}^{\mathbb{T}}$, then compute the noisy IRIS counts $\hat{\mathbf{X}}^{\mathbb{L}}$ from $\hat{\mathbf{X}}^{\mathbb{T}}$ by applying Formula (1). Although both techniques guarantee ε-DP according to Theorem 2, in terms of utility, they are suboptimal.

Cells with small counts have larger relative error, whereas larger counts better resist noise. This is due to the fact that the injected noise is independent of the magnitude of the original counts but only depends on their sensitivity. Therefore, the best approach is to first select cells having the largest counts (which can be either tower or IRIS cells) such that they cover whole Paris, perturb the counts of these cells, and then recompute the noisy counts of the smaller IRIS cells, which were not selected in the cover, from the larger (noisy) tower counts.

However, selecting the optimal cover of Paris (i.e., the set of cells having the largest counts) must also be differential private. Fortunately, a simple heuristic helps us to accurately approximate the optimal cover without using the true counts of the cells (that would require to introduce more noise): cells with large size tend to have large counts[4]. This is also confirmed by Figure 2a and 2b. Hence, we re-state the problem as follows.

How can we select the minimum cardinality subset of cells $\mathbb{C} \subseteq \mathbb{T} \cup \mathbb{L}$ such that $\mathbb{C}$ is a complete cover (i.e., covers whole Paris)?

More formally, let $G(V, E)$ denote a graph, where each vertex corresponds to a cell in $\mathbb{T} \cup \mathbb{L}$, and $(v, v') \in E$ iff cells v and v' overlap. In this setting, our problem translates to the classical minimum vertex cover problem [3]. Indeed, as $\mathbb{T}$ and $\mathbb{L}$ are also complete covers of Paris, each vertex in V has at least one edge (a tower cell is always overlapped by at least one IRIS cell, and vice-verse), and we want to compute the minimum cardinality subset of cells which cover all the overlapping areas between cells. Although the minimum vertex cover problem is NP-hard in general, our covering problem belongs to the special cases which can be efficiently solved.

Theorem 5 (Minimum cardinality cover) *Selecting the minimum cardinality subset of cells $\mathbb{C} \subseteq \mathbb{T} \cup \mathbb{L}$ such that $\mathbb{C}$ is a complete cover can be solved in $O(|\mathbb{T}||\mathbb{L}|\sqrt{|\mathbb{T}| + |\mathbb{L}|})$.*

PROOF. $G(V, E)$ is bipartite, since the IRIS cells as well as the tower cells are partitionings of Paris, i.e., there are no overlapping cells in any of the two sets. Hence, for all $(v, v') \in E$, either $v \in \mathbb{T}$ and $v' \in \mathbb{L}$, or $v' \in \mathbb{T}$ and $v \in \mathbb{L}$. For bipartite graphs, the minimum vertex cover problem is equivalent to the maximum matching problem based on König's theorem [3], which can be solved in polynomial time, e.g., with the Hopcroft-Karp algorithm [16] in $O(|E|\sqrt{|V|})$, where $|V| = |\mathbb{T}| + |\mathbb{L}|$ and $|E| \leq |\mathbb{T}||\mathbb{L}|$. □

[4]Tower cells are represented by their Voronoi polygons as it is depicted in Figure 1

Figure 2c shows the largest IRIS and tower cells covering Paris. We computed the mean count of each cell over the whole week which are illustrated by the cell colors. Apparently, the minimum cover contains cells with larger counts; the mean counts are 91 on average for the IRIS cells (Figure 2b) and 120 on average for the minimum cover (Figure 2c).

However, care must be taken before computing the cell counts in the minimum cover $\mathbb{C}$. Since $\mathbb{C}$ can contain both IRIS and tower cells which may overlap, the L_1-sensitivity of all the counts in $\mathbb{C}$ can be larger than ℓ. Indeed, if $\mathbb{C}$ contains a tower cell T and one of its overlapping IRIS cell L, then adding/removing a user who visited T at time t will change $\overline{X}_t^T$ with 1, and also $\overline{X}_t^L$ with a non-zero value. A trivial (but not optimal) solution is to modify the counts of all towers in $\mathbb{C}$ which have overlapping IRIS cells. For example, if T overlaps with IRIS cells $\{L_1, L_2, \ldots, L_c\}$, then $\overline{X}_t^T$ should be reduced by $\sum_{i=1}^{c} \overline{X}_t^{L_i} \times \frac{\mathsf{size}(L_i \cap T)}{\mathsf{size}(T)}$.

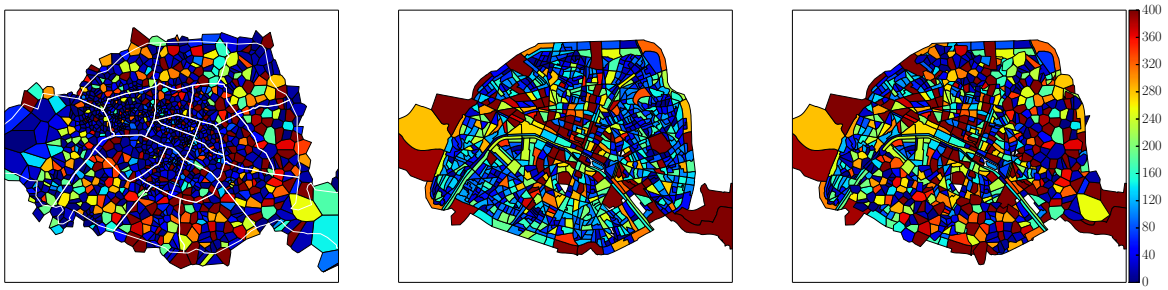
5.3 Perturbation

After identifying the largest covering cells, their time series (i.e., $\overline{\mathbf{X}}^{\mathbb{C}}$) can be perturbed by adding $\mathcal{L}(\ell/\varepsilon)$ to each count in all time series (see Theorem 2). Unfortunately, this naive method provides very poor results which is also illustrated in Figure 3a. Indeed, individual cells have much smaller counts than the magnitude of the injected noise; the standard deviation of the Laplacian noise is 141 with $\varepsilon = 0.3$, which is even larger than the mean count in the minimum cover.

A better approach exploits (1) the similarity of geographically close time series, as well as (2) their periodic nature. In particular, we first cluster nearby less populated cells until their aggregated counts become sufficiently large to resist noise. The key observation is that the time series of close cells follow very similar trends, but their counts usually have different magnitudes. Hence, if we simply aggregate (i.e., sum up) all time series within such a cluster, the aggregated series will have a trend close to its individual components yet large enough counts to tolerate perturbation. To this end, we first accurately approximate the time series of individual cells by normalizing their aggregated time series (i.e., divide the aggregated count of each hour with the total number of visits inside the cluster), and scale back with the (noisy) total number of visits of individual cells.

In order to guarantee DP, we also need to perturb the aggregated time series before normalization. To do so, we exploit their periodic nature and apply a Fourier-based perturbation scheme [24, 1]: we add noise to the Fourier coefficients of the aggregated time series, and remove all high-frequency components that would be suppressed by the noise. As only low-frequency components are retained and perturbed, this method preserves the trends of the original data more faithfully than LPA.

The whole perturbation process is summarized in Algorithm 2. First, the noisy total number of visits of each cell in the minimum cover $\mathbb{C}$ is computed by adding noise $\mathcal{L}(2\ell/\varepsilon)$ to $\sum_{t=0}^{167} \overline{X}_t^i$ for cell i (Line 1). These noisy total counts are used to cluster similar cells in Line 2 by invoking Algorithm 3. When the clusters are created, their aggregated time series (i.e., the sum of all cells' time series within the cluster) is perturbed with a Fourier-based perturbation scheme in Line 5 (Algorithm 4). Finally, the perturbed time series of each cell i in cover $\mathbb{C}$ is computed in Line 7 by scaling back



(a) Tower cells, $|\mathbb{T}| = 1303$ (b) IRIS cells, $|\mathbb{L}| = 989$ (c) Minimum cardinality cover, $|\mathbb{C}| = 826$

Figure 2: Covering Paris with the largest tower and IRIS cells. Each cell is colored based on their mean count. Large cells tend to have large counts (closer to red), while small cells are less populated (closer to blue). The minimum cover (Figure 2c) includes large, more populated cells: IRIS cells from the city center, and tower cells around the perimeter.

Algorithm 2 Perturbation

Input: Minimum cover $\mathbb{C}$, Pre-sampled time series $\overline{\mathbf{X}}^{\mathbb{C}}$, Minimum total count τ, Privacy budget ε, δ, Sensitivity $\Delta_1(\overline{\mathbf{X}}^{\mathbb{C}}) = \ell$
Output: Noisy time series $\hat{\mathbf{X}}^{\mathbb{C}}$

1: $\hat{S}_i := \sum_{t=0}^{167} \overline{X}_t^i + \mathcal{L}(2\ell/\varepsilon)$ for each $i \in \mathbb{C}$
2: $\mathbb{E} :=$ Cluster$(\mathbb{C}, \tau, \hat{S})$ `//see Algorithm 3`
3: **for** each cluster $E \in \mathbb{E}$ **do**
4: $\quad \overline{\mathbf{X}}^E := \langle \sum_{i\in E} \overline{X}_0^i, \sum_{i\in E} \overline{X}_1^i, \ldots, \sum_{i\in E} \overline{X}_{167}^i \rangle$
5: $\quad \hat{\mathbf{X}}^E :=$ EFPAG$(\overline{\mathbf{X}}^E, \varepsilon/2, \delta)$ `//see Algorithm 4`
6: $\quad$ **for** each cell $i \in E$ **do**
7: $\quad\quad \hat{\mathbf{X}}^i := \hat{S}_i \cdot (\hat{\mathbf{X}}_t^E / ||\hat{\mathbf{X}}^E||_1)$
8: $\quad$ **end for**
9: **end for**

Algorithm 3 Cluster cells

Input: Minimum cover $\mathbb{C} = \{c_0, c_1, \ldots, c_{|\mathbb{C}|}\}$, Minimum total count τ, Noisy total counts $\hat{S}$ of cells in $\mathbb{C}$
Output: Cluster configuration $\mathbb{E} \subset 2^{|\mathbb{C}|}$

1: $\mathbb{E} := \{\{c_0\}, \{c_1\}, \{c_2\}, \ldots, \{c_{|\mathbb{C}|}\}\}$
2: $\hat{S}_E := \sum_{c_i \in E} \hat{S}_i$ for each cluster $E \in \mathbb{E}$
3: **while** $\exists E \in \mathbb{E}$ such that $\hat{S}_E < \tau$ **do**
4: $\quad E := \arg\min_{E\in\mathbb{E}} \hat{S}_E$
5: $\quad$ Let E' be the geographically closest neighbor of E
6: $\quad E := E \cup E'$
7: $\quad \hat{S}_E := \hat{S}_E + \hat{S}_{E'}$
8: $\quad$ Remove E' from $\mathbb{E}$
9: **end while**

the normalized aggregated time series with the noisy total count cell i (i.e., with $\hat{S}_i$). Since Line 1 guarantees $\varepsilon/2$-DP to the total counts $(\Delta_1(\overline{\mathbf{X}}^{\mathbb{C}}) = \ell)$, it follows from Theorem 1 that Algorithm 2 is (ε, δ)-DP if EFPAG is $(\varepsilon/2, \delta)$-DP.

5.3.1 Clustering cells

Algorithm 3 is a simple iterative process that, in each iteration, merges the least visited cluster with its geographically closest neighboring cluster until all clusters in the resultant configuration have a total count larger than a predefined threshold τ. Initially, each cluster is a singleton composed of an individual cell in the cover. Then, in Line 4, we select the cluster which has the smallest (noisy) total count, and merge with its closest neighboring cluster in Line 5-8. The distance between two clusters are measured with the physical distance between their cluster centers[5]. In each step, the noisy total count of each cluster is computed as the sum of all (noisy) total counts of each cell within the cluster (Line 7). Since the total counts of cells are noisy, Algorithm 3 preserves DP.

5.3.2 Perturbing aggregated time series

To perturb aggregated time series, we build on the Fourier Perturbation Algorithm (FPA) [24]:

[5]The cluster center is the centroid of its consitituent cell polygons.

1. Compute the Fourier coefficients $\mathbf{F} = \langle F_0, F_1, \ldots, F_{n-1} \rangle$ of the input aggregated time series $\overline{\mathbf{X}}$ with length n by discrete Fourier transform.

2. Remove the last $n - k$ coefficients from $\mathbf{F}$, which correspond to the high-frequency components in $\overline{\mathbf{X}}$, and retain only the first k elements of $\mathbf{F}$, denoted by $\mathbf{F}^k$. Note that k is an input to the algorithm.

3. Generate the noisy version of $\mathbf{F}^k$, denoted by $\hat{\mathbf{F}}^k$, by Laplace mechanism: add i.i.d Laplace noise $\mathcal{L}(\sqrt{k}/\varepsilon)$ to each coefficient in $\mathbf{F}^k$.

4. Pad $\hat{\mathbf{F}}^k$ to be a n-dimensional vector by appending $n - k$ zeros, which is denoted by $\text{PAD}^n(\hat{\mathbf{F}}^k)$. Finally, the inverse DFT is applied to $\text{PAD}^n(\hat{\mathbf{F}}^k)$ to obtain a noisy version of $\overline{\mathbf{X}}$.

FPA provably guarantees ε-DP [24]. Enhanced FPA [1] improves basic FPA by selecting the coefficients to be removed more effectively. Specifically, in Step 2, EFPA chooses k probabilistically using the exponential mechanism such that the values of k which minimize the root-sum-squared error $\mathbb{E}||\overline{\mathbf{X}} - \hat{\mathbf{X}}||_2 \leq \sqrt{\sum_{i=k+1}^{n} |F_{i-1}|^2 + \frac{2k\Delta_2(\mathbf{X})}{\varepsilon}}$ (RSSE) have exponentially larger probability to be selected. In this paper, we improve the accuracy of EFPA in two ways. First, instead of the Discrete Fourier Transform (DFT), we

Algorithm 4 EFPAG

Input: Truncated time series $\overline{\mathbf{X}}$ with length n, Privacy budget ε, δ, L_2-sensitivity of $\overline{\mathbf{X}}$: $\Delta_2(\overline{\mathbf{X}})$

Output: Noisy time series $\hat{\mathbf{X}}$ with length n

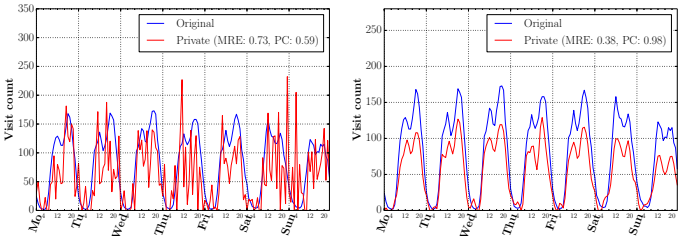
1: $\mathbf{F} := \mathrm{DCT}(\overline{\mathbf{X}})$ `// Discrete Cosine Transform`

2: Compute $u_G(\overline{\mathbf{X}}, k) = \sqrt{\sum_{i=k+1}^{n} |F_{i-1}|^2} + \frac{\sqrt{2}\Delta_2(\overline{\mathbf{X}})\sqrt{k}\ln^{1/2}(4/\delta)}{\varepsilon}$ for all $1 \le k \le n$

3: Select k with probability $\propto \exp\left(-\frac{\varepsilon \cdot u_G(\overline{\mathbf{X}},k)}{4\Delta_2(\overline{\mathbf{X}})}\right)$

4: $\hat{\mathbf{F}}^k := \mathbf{F}^k + \langle \mathcal{G}(\sqrt{2}\Delta_2(\overline{\mathbf{X}})\ln^{1/2}(4/\delta)/\varepsilon)\rangle^k$

5: **return** $\hat{\mathbf{X}} = \mathrm{IDCT}(\mathrm{PAD}^n(\hat{\mathbf{F}}^k))$ `//Inverse DCT`



(a) LPA (b) Our approach

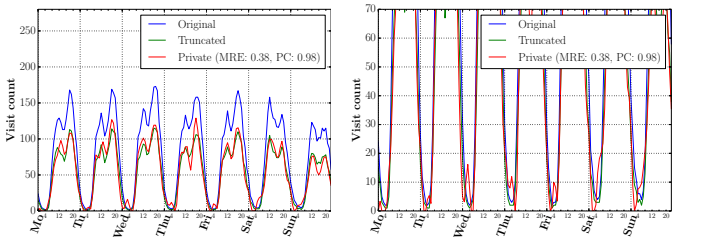
Figure 3: Noisy time series of an IRIS cell ($\varepsilon = 0.3$, $\ell = 30$)

apply the orthonormal version of the Discrete Cosine Transform (DCT), which tends to provide smaller high frequency components due to its different boundary conditions [26]. This can result in smaller RSSE when these components are removed in Step 2. Moreover, since we use orthonormal DCT, the resultant scheme also preserves ε-DP [1].

Second, instead of adding Laplace noise, we add properly calibrated Gaussian noise to the first k Fourier coefficients of $\overline{\mathbf{X}}$ thereby providing larger accuracy at the cost of weaker privacy. More specifically, when $\hat{\mathbf{F}}^k$ is generated in Step 3, we employ the Gaussian mechanism instead of LPA and add i.i.d Gaussian noise $\mathcal{G}(\sigma)$ to each coefficient in $\mathbf{F}^k$, where $\sigma = \sqrt{2}\Delta_2(\overline{\mathbf{X}})\ln^{1/2}(4/\delta)/\varepsilon$ to provide (ε, δ)-probabilistic DP based on Theorem 3. In addition, we select k with probability $\propto \exp\left(-\frac{\varepsilon \cdot u_G(\overline{\mathbf{X}},k)}{4}\right)$ in Step 2, where $u_G(\overline{\mathbf{X}}, k) = \sqrt{\sum_{i=k+1}^{n} |F_{i-1}|^2} + \frac{\sqrt{2}\Delta_2(\overline{\mathbf{X}})\sqrt{k}\ln^{1/2}(4/\delta)}{\varepsilon}$ which follows from Theorem 5 in [1] and Theorem 3. When we use GPA in Step 3, the new scheme (with DCT) is denoted by EFPAG in the rest.

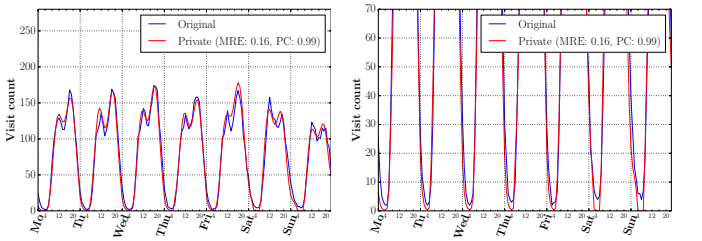
Since Gaussian noise has smaller variance than Laplacian, EFPAG provides better accuracy than EFPA. Specifically, the variance of the Gaussian noise added to the Fourier coefficients is $8\Delta_2(\overline{\mathbf{X}})^2\ln(4/\delta)/\varepsilon^2$, which is independent of the number of retained coefficients (i.e., k). By contrast, the variance of the Laplace noise added to the coefficients in EFPA is $8\Delta_2(\overline{\mathbf{X}})^2 k/\varepsilon^2$ which linearly increases with the number of retained coefficients. However, this improvement also leads to some privacy degradation which is measured by δ. In the sequel, we fix δ to $2 \cdot 10^{-6} (\le 1/N)$.

EFPAG is summarized in Algorithm 4, where the total budget ε is uniformly divided between GPA (Line 4) and exponential mechanism (Line 2), therefore, EFPAG is (ε, δ)-probabilistically DP due to Theorem 1.



(a) Large counts (b) Small counts around local minimas

Figure 4: Our scheme before improvements ($\varepsilon = 0.3$, $\delta = 2 \cdot 10^{-6}$, $\ell = 30$).



(a) Scaling (b) Smoothing

Figure 5: Our scheme after improvements ($\varepsilon = 0.3, \delta = 2 \cdot 10^{-6}, \ell = 30$)

Finally, in order to employ EFPA(G), we need to compute the L_2-sensitivity of the counts in the cover $\mathbb{C}$, i.e., $\Delta_2(\overline{\mathbf{X}}^{\mathbb{C}})$. Indeed, since $\mathbb{E}$ is a partitioning of $\mathbb{C}$, $\Delta_2(\overline{\mathbf{X}}^{\mathbb{E}}) = \Delta_2(\overline{\mathbf{X}}^{\mathbb{C}})$. Recall that, as a result of pre-sampling, at most a single visit of a user is retained in any slot, and at most ℓ visits per user over the whole week. This means that, for any t, there is only a single tower whose count can change (by at most 1) by modifying a single user's data. From Formula (1), it follows that the total change of all IRIS cell counts is at most 1 at any t, and hence $\Delta_2(\overline{\mathbf{X}}^{\mathbb{L}}) \le \Delta_2(\overline{\mathbf{X}}^{\mathbb{T}}) = \sqrt{\ell}$ based on the definition of L_2-norm. Since $\mathbb{C} \subseteq \mathbb{T} \cup \mathbb{L}$, $\Delta_2(\overline{\mathbf{X}}^{\mathbb{C}}) \le \sqrt{\ell}$. Figure 3 illustrates the improvement of our approach (Clustering + EFPAG) over simple LPA.

5.4 Further improvements

Although our approach is clearly superior to LPA, Figure 4 still suggests a large error on average. This difference between $\hat{\mathbf{X}}$ and $\mathbf{X}$ is the result of two errors: the sampling error (between $\overline{\mathbf{X}}$ and $\mathbf{X}$) is attributed to pre-sampling, whereas the perturbation error (between $\hat{\mathbf{X}}$ and $\overline{\mathbf{X}}$) is due to our perturbation scheme.

As illustrated by Figure 4a, sampling error mainly distorts large counts: although the noisy counts are close to the counts of the *truncated* (pre-sampled) time series between 9:00 AM and 11:00 PM, it is still far from the original count values. This significantly increases MRE.

In addition, as Figure 4b also shows, noisy counts also deviate from pre-sampled as well as from original counts around the local minimas (close to 4:00 AM every day), which further deteriorates MRE. This perturbation error is caused by the higher frequency components that are retained and perturbed by EFPA(G).

To alleviate these errors, we propose two further improvements: first, we improve the perturbation of total cell counts (Line 1 in Algorithm 2), which is used in cell clustering (Algorithm 3) and scaling (Line 6 in Algorithm 2). Then, as a post-processing step, we smooth out small counts (i.e., between 0:00 and 6:00 AM) through non-linear least-square fitting to diminish perturbation error.

5.4.1 Improved scaling

Recall that we scale back the normalized aggregated time series with $\hat{S}_i$ in Line 6 (Algorithm 2), where $\hat{S}_i = \sum_{t=0}^{167} \overline{X}_t^i + \mathcal{L}(2\ell/\varepsilon)$. Since $\overline{\mathbf{X}}^i$ is the pre-sampled time series of cell i, $\hat{\mathbf{X}}^i$ (Line 6) will be a scaled down version of the original time series $\mathbf{X}^i$ due to the fact that the ℓ visits per individual are sampled *uniformly* at random. Also, as we have discussed in Section 5.1, adding Laplace noise directly to the original total count $\sum_{t=0}^{167} X_t^i$ is very inaccurate, as the sensitivity of $\sum_{t=0}^{167} X_t^i$ is $\Delta_1(\mathbf{X})$ which is too large.

We rather perturb the *original* total count $\sum_{t=0}^{167} X_t^i$ using a different approach: we first approximate the relative frequencies of each *tower* by another constraint sampling, and scale back these frequencies to count values with the (noisy) total number of visits in the dataset. The main idea is that sampling requires only a small amount of noise to guarantee privacy, while the total number of all visits is so large that it tolerates a large noise magnitude.

In particular, we estimate the histogram H where a bin H_j represents the frequency of visits at tower j, i.e., $H_j = \sum_{t=0}^{167} X_t^j / K$, where K denotes the total number of tower visits in the dataset ($K = \sum_{t=0}^{167} \sum_{T \in \mathbb{T}} X_t^T$). To do so, we sample a single visit per user uniformly at random, and create a new histogram $\tilde{H}$ from the sampled visits (with size N). Using this approximative histogram $\tilde{H}$, the total number of visits $\sum_{t=0}^{167} X_t^j$ of a tower j is computed as $(\tilde{H}_j + \mathcal{L}(2/\varepsilon')) \times \hat{K}$, where $\hat{K} = K + \mathcal{L}(2\Delta_1(\mathbf{X})/\varepsilon')$ and $\Delta_1(\mathbf{X})$ is universally fixed for all input dataset[6]. Finally, having the noisy $\sum_{t=0}^{167} X_t^j$ for each tower cell j, we can also compute the noisy $\sum_{t=0}^{167} X_t^L$ for any IRIS cell L (using Formula (1)) and calculate $\hat{S}_i$ for all cell i in $\mathbb{C}$. This technique is $2 \cdot (\varepsilon'/2) = \varepsilon'$ differential private based on Theorem 1.

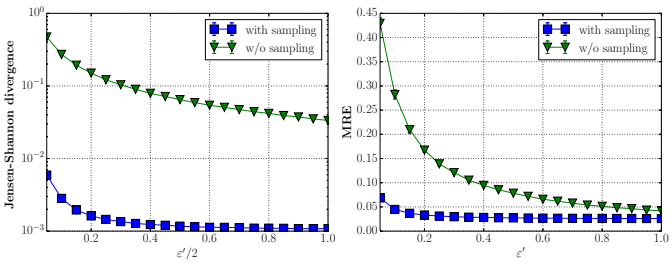
Figure 6a compares the accuracy of our sampling approach to perturb the relative frequencies of each tower (i.e., sampling is followed by adding $\mathcal{L}(2/\varepsilon')$ to each H_j) with the naive Laplace approach (i.e., $\mathcal{L}(2\Delta_1(\mathbf{X})/\varepsilon')$ is added to each H_j without sampling). Sampling clearly boosts the accuracy of histogram perturbation (Figure 6a) especially for smaller values of ε', and eventually yields significantly more accurate estimation of $\sum_{t=0}^{167} X_t^T$ for all towers T (Figure 6b).

The effect of scaling is illustrated in Figure 5a. Recall that the full privacy budget ε is divided equally between EFPA(G) and scaling (see Algorithm 1). Hence, our sanitization scheme is ε-DP (or (ε, δ)-prob. DP with EFPAG) based on Theorem 1.

5.4.2 Smoothing

In order to smooth out low (noisy) counts around the local minimas (around 4:00 AM each day), we fit an exponential curve to the noisy counts between 0:00 AM and 4:00 AM where the counts are exponentially decreasing, and another exponential curve between 4:00 AM and 6:00 AM, where the counts are exponentially increasing. In particular, we fit function $g(x, a, b) = a \cdot \exp(b \cdot x)$ to the noisy counts, i.e., compute parameters a, b such that the error $\sum_i (\hat{X}_i - g(x_i, a, b))^2$ is minimized where x_i runs over the hours of the given time intervals for each day, and then replace the noisy counts with the values of the fitted function. This is a standard non-linear least square fitting problem which can be approximated with any numerical minimization method (e.g., Levenberg-Marquardt algorithm [18]). Since this operation is performed on the noisy time series, it is already private. The effect of smoothing in illustrated in Figure 5b.

[6] $\Delta_1(\mathbf{X})$ is fixed to 732 in this paper. Notice that although $\Delta_1(\mathbf{X})$ is large so is K: $K = 137,255,052$ in our dataset, and $|K - \hat{K}|/K$ is less than 10^{-5} on average



(a) Perturbing H with sampling and without sampling.

(b) MRE of the noisy $\sum_{t=0}^{167} X_t^T$ over all towers T.

Figure 6: Perturbing the total visits $\sum_{t=0}^{167} X_t^T$ of each tower cell T.

6. PERFORMANCE EVALUATION

We evaluate the utility of our scheme depending on the guaranteed privacy (i.e., ε) with EFPA and EFPAG, where $\delta = 2 \cdot 10^{-6} < 1/N$. The minimum total count τ used in Algorithm 3 is adjusted such that the expected RSSE is less than 1% of the total count when all coefficients are retained in EFPA(G). That is, $\tau = \sqrt{168} \cdot \sigma / 0.01$, where σ^2 is the variance of noise added to each coefficient. We compare our approaches to the naive Laplace mechanism (LPA) that adds $\mathcal{L}(\ell/\varepsilon)$ noise to each count of each time series in cover $\mathbb{C}$. We use the CDR dataset described in Section 4.

First, we analyze the utility depending on the pre-sampling size ℓ. Then, we show how pre-sampling combined with the improved scaling and smoothing boost accuracy, and also report the error distribution among individual IRIS cells. Finally, we measure the Earth Mover's Distance (EMD) which captures the error between spatial distributions in terms of geographical distances.

6.1 Utility depending on the pre-sample size

Recall that the number of visits retained per user (i.e., ℓ) determines the injected noise in the perturbation phase. In general, larger values of ℓ imply larger noise, which degrades utility. On the other hand, smaller values of ℓ preserve more information about individuals which results in a more accurate representation of the original dataset. The goal is to select a value of ℓ which yields the best trade-off. Nevertheless, we experimentally show next that our scheme exhibits stable performance with quite different values of ℓ.

In Figure 7, we report the utility of our scheme with EFPAG for three values of ℓ: 10, 30 and 168. In each case, perturbation is followed by the improvements described in

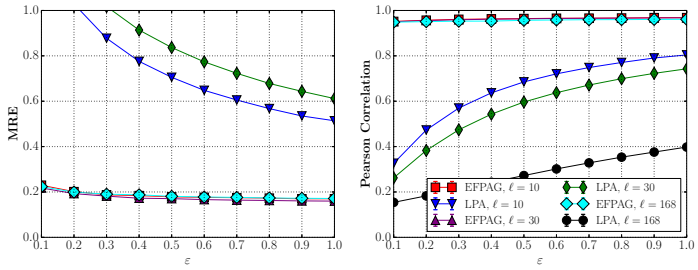


Figure 7: Utility depending on the pre-sampling size ℓ.

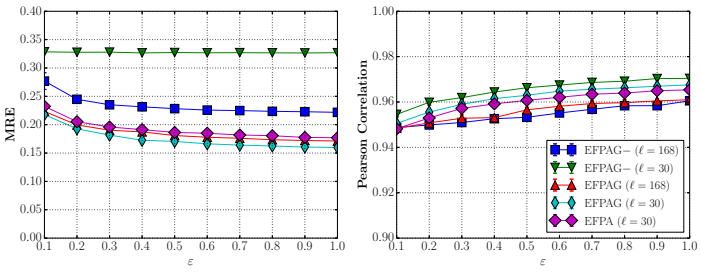


Figure 8: Utility of our scheme with different perturbation techniques.

Section 5.4. Specifically, we computed the average of MRE and Pearson correlation over all cells. We repeated the whole process 20 times and plotted the mean and standard deviation of the average MRE and PC over all executions.

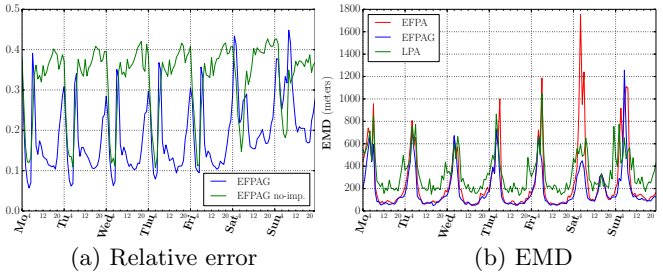
LPA has significantly larger error for all values of ℓ, and provides especially poor results for smaller values of ε. By contrast, our scheme does not only provide practical utility even for stringent privacy guarantee, but also has stable performance for different ℓ. For instance, for $\varepsilon = 0.3$, MRE is less than 20%, while PC is larger than 0.95. Therefore, the output of our scheme has almost perfect linear correlation with the original data thanks to the combination of clustering and the Fourier perturbation approach. Moreover, the variations of these values are very moderate: 0.2 ± 0.03 and 0.95 ± 0.03, respectively, for different values of ℓ. In the rest of the paper, we fix ℓ to 30.

EFPAG and EFPA[7] are compared in Figure 8. EFPAG outperforms EFPA especially for smaller ε: MRE is reduced by 0.03 and PC is increased by 0.02 on average.

6.2 Pre-sampling with scaling

The distortion effect of pre-sampling is mitigated by the improved scaling step detailed in Section 5.4.1. Our aim now is to show that scaling and smoothing indeed results in better utility. Figure 8 depicts a variation of our scheme, denoted by "EFPA−" when the improvements described in Section 5.4 are not employed after perturbation. The results show that MRE is reduced by 0.07 on average when the pre-sample size is diminished to $\ell = 30$ and improvements are employed. By contrast, PC is increased only by about 0.01 for smaller values of ε; the change is not so significant due to the fact that scaling does not influence linear correlation,

[7]Recall that EFPAG adds Gaussian noise whereas EFPA adds Laplacian noise to the retained Fourier coefficients.



(a) Relative error (b) EMD

Figure 9: Error depending on time with EFPAG ($\varepsilon = 0.3, \ell = 30$)

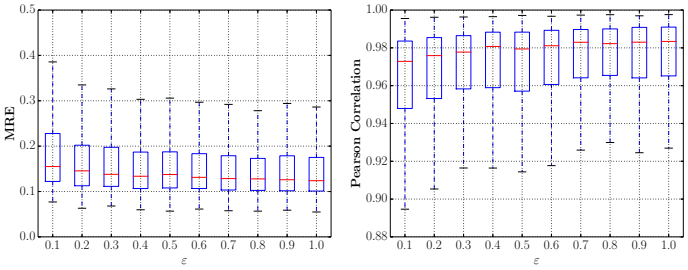


Figure 10: Error and Pearson correlation on IRIS cells with EFPAG, $\ell = 30$. The box extends from the lower to upper quartile values of the cell errors, with a red line at the median.

and smoothing modifies relatively small number of counts in general.

In Figure 9, we also plotted the average relative error depending on the time for our scheme with and without improvements. In particular, we computed the relative error and took the average over all cells in each hour. Figure 9a confirms that scaling significantly diminishes the relative error in daylight when counts are larger. The improvement can be almost a factor of 4. This has particular importance in practice, as location counts in daylight are usually more important than at night.

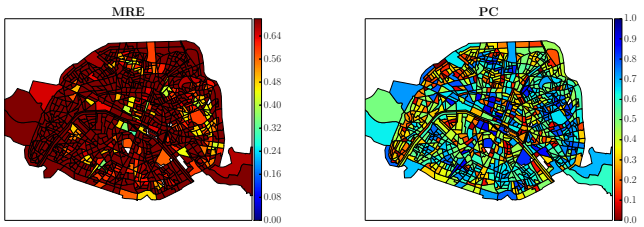
6.3 Error variation among IRIS cells

Figure 10 shows through box plots how MRE and Pearson correlation change among IRIS cells; we compute these metrics for each IRIS cell, and compute the corresponding box plot over the metric values of all cells. Although medians do not change significantly for different values of ε, MRE has larger variation for smaller ε, i.e., there are more cells which have larger error.

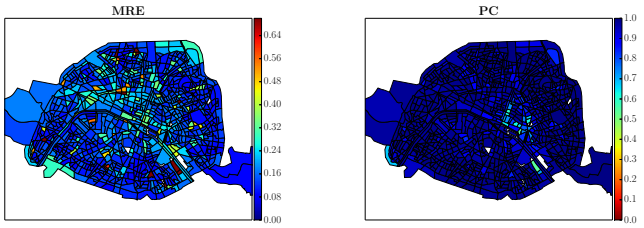
The MRE and PC of individual IRIS cells are also illustrated by color maps in Figure 11. This figure shows that our scheme can provide practical utility for most cells with strong privacy guarantee. Specifically, the average MRE over all cells is only 0.17 with $\varepsilon = 0.3$.

6.4 Earth Mover's Distance

In order to compare the sanitized spatial probability distribution with the original one at a given time, we use Earth Mover's Distance (EMD) [25]. EMD measures the "amount of energy" (or cost) needed to transform one distribution to another, and is a metric for probability distributions (i.e., location counts have to be normalized). Formally, for any



(a) LPA (Avg. MRE: 1.01, PC: 0.47)



(b) Our scheme with EFPAG (Avg. MRE: 0.17, PC: 0.96)

Figure 11: MRE and PC of each IRIS cell ($\ell = 30$, $\varepsilon = 0.3$)

t, $\text{EMD}(\mathbf{X}_t^{\mathbb{L}}, \hat{\mathbf{X}}_t^{\mathbb{L}}) = \min_{\{f_{ij}\}} \sum_{i,j} f_{ij} d_{ij}$ such that $f_{ij} \geq 0$, $\sum_j f_{ij} \leq X_t^i / \sum_k X_t^k$, $\sum_i f_{ij} \leq \hat{X}_t^j / \sum_k \hat{X}_t^k$, where $\{f_{ij}\}$ denotes the set of all possible flows (each f_{ij} represents the amount of probability mass transported from IRIS cell i to j), and d_{ij} is the geographical distance between the centers of cells i and j, resp. Intuitively, EMD measures the meters of error between two spatial density maps. Figure 9b reports the EMD depending on the time. The mean EMD over the whole week is 258 meters for EFPA, 188 meters for EFPAG, and 341 meters for LPA.

7. CONCLUSIONS

The goal of this work is to demonstrate through a real-world application that differential privacy can be a practical model for data anonymization, even if the input dataset has large dimension and/or is highly sensitive. We showed that, in order to achieve meaningful accuracy, the sanitization process has to be carefully customized to the application and public characteristics of the dataset. We strongly believe that there are no "universal" sanitization solutions that fit all applications, i.e., provide good accuracy in all scenarios. In particular, achieving the best performance requires to find the most faithful and concise representation of the data, such that it withstands perturbation. In our application (i.e., spatio-temporal density), clustering and sampling with Fourier-based perturbation are seemingly the best choices due to the periodic nature and large sensitivity of location counts. We experimentally showed that our scheme can provide practical utility and strong privacy guarantee.

8. REFERENCES

[1] G. Ács, R. Chen, and C. Castelluccia. Differentially private histogram publishing through lossy compression. In *ICDM*, 2012.

[2] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *ACM CCS*, pages 901–914. ACM, 2013.

[3] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. Elsevier, New York, 1976.

[4] T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3):26:1–26:24, Nov. 2011.

[5] R. Chen, G. Ács, and C. Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *ACM CCS*, pages 638–649, 2012.

[6] C.-Y. Chow and M. F. Mokbel. Trajectory privacy in location-based services and data publication. *SIGKDD Explor. Newsletter*, 13(1):19–29, Aug. 2011.

[7] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu. Differentially private spatial decompositions. In *ICDE*, pages 20–31, 2012.

[8] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports, Nature*, March 2013.

[9] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[11] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *ACM STOC*, pages 715–724, 2010.

[12] L. Fan and L. Xiong. Real-time aggregate monitoring with differential privacy. In *ACM CIKM*, pages 2169–2173, 2012.

[13] P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *Percom*, pages 390–397, 2009.

[14] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453, 2008.

[15] M. Götz. *On User Privacy in Personalized Mobile Services*. PhD thesis, Cornell University, May 2012.

[16] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4), 1973.

[17] G. Kellaris and S. Papadopoulos. Practical differential privacy via grouping and smoothing. In *VLDB*, pages 301–312, 2013.

[18] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.

[19] N. Li, W. Yang, and W. Qardaji. Differentially private grids for geospatial data. In *ICDE*, pages 757–768, 2013.

[20] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *ICDE*, 2008.

[21] F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD*, pages 19–30, 2009.

[22] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, 2007.

[23] D. J. Mir, S. Isaacman, R. Cáceres, M. Martonosi, and R. N. Wright. Dp-where: Differentially private modeling of human mobility. In *BigData Conference*, pages 580–588, 2013.

[24] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *SIGMOD*, 2010.

[25] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, Nov. 2000.

[26] G. Strang. The discrete cosine transform. *SIAM Rev.*, 41(1):135–147, Mar. 1999.

[27] X. Xiao, G. Bender, M. Hay, and J. Gehrke. iReduct: Differential privacy with reduced relative errors. In *SIGMOD*, pages 229–240, 2011.

[28] H. Zang and J. Bolot. Anonymization of location data does not work: A large-scale measurement study. In *Mobicom*, pages 145–156, 2011.