Differentially Private Release of Spatio-Temporal Data Statistics

Arvind Rameshwar
IUDX

based on joint works with

Anshoo Tandon — IUDX
Prajjwal Gupta — Northeastern U.
Novoneel Chakraborty — IUDX
Aditya Singh — IISc
Abhay Sharma — IUDX

CNI Networks Seminar 2024

Some background

- The release of even seemingly innocuous functions of a private dataset can leak information about identities of users/participants.

Sierra Williams

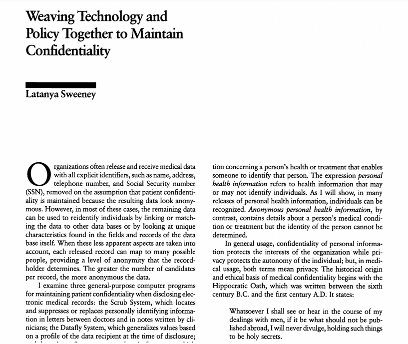
July 16th, 2014

On Taxis and Rainbow Tables: Lessons for researchers and governments from NYC's improperly anonymized taxi logs.

1 comment

Estimated reading time: 5 minutes

When New York City's Taxi and Limousine Commission made publicly available 20GB worth of trip and fare logs, many welcomed the vast trove of open data. Unfortunately, prior to being widely shared, the personally identifiable information had not been anonymized properly. Vijay Pandurangan *describes the structure of the*



Weaving Technology and Policy Together to Maintain Confidentiality

Latanya Sweeney

Organizations often release and receive medical data with all explicit identifiers, such as name, address, telephone number, and Social Security number (SSN), removed on the assumption that patient confidentiality is maintained because the resulting data look anonymous. However, in most of these cases, the remaining data can be used to reidentify individuals by linking or matching the data to other data bases or by looking at unique characteristics found in the fields and records of the data base itself. When these less apparent aspects are taken into account, each released record can map to many possible people, providing a level of anonymity that the record-holder determines. The greater the number of candidates per record, the more anonymous the data.

I examine three general-purpose computer programs for maintaining patient confidentiality when disclosing electronic medical records: the Scrub System, which locates and suppresses or replaces personally identifying information in letters between doctors and in notes written by clinicians; the Datafly System, which generalizes values based on a profile of the data recipient at the time of disclosure;

tion concerning a person's health or treatment that enables someone to identify that person. The expression *personal health information* refers to health information that may or may not identify individuals. As I will show, in many releases of personal health information, individuals can be recognized. *Anonymous personal health information*, by contrast, contains details about a person's medical condition or treatment but the identity of the person cannot be determined.

In general usage, confidentiality of personal information protects the interests of the organization while privacy protects the autonomy of the individual; but, in medical usage, both terms mean privacy. The historical origin and ethical basis of medical confidentiality begins with the Hippocratic Oath, which was written between the sixth century B.C. and the first century A.D. It states:

> Whatsoever I shall see or hear in the course of my dealings with men, if it be what should not be published abroad, I will never divulge, holding such things to be holy secrets.

- The framework of differential privacy (DP) was introduced in [Dwork et al. (2006)] for the design/analysis of mechanisms resilient to such attacks.

An explosion of works since then

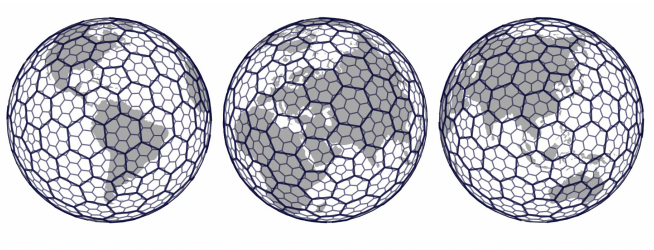
. . . and several more

Our interest: User-level DP

- Standard DP guarantees the privacy of a user when he/she contributes at most one data sample.
- However, most real-world applications, e.g., language/image recognition tasks, federated learning, traffic analysis, record multiple contributions from each user.
- Recent work [Levy et al. (2021), Cummings et al. (2022)] formally defined user-level DP that guarantees the privacy of any user who contributes potentially multiple samples, and provided explicit private mechanisms for mean estimation.
- Other works considered user-level privacy in the context of bounding user contributions in ML models [Amin et al. (2019)] and in private federated learning [Wang et al. (2019), McMahan et al. (2018)].

Basic setup

- Consider a city whose area is partitioned into grids/hexagons, e.g., using Uber's spatial indexing system H3.



Source: `https://www.uber.com/en-IN/blog/h3/`

- We quantize/bin the data records in each hexagon into fixed-duration timeslots.
- We seek to release user-level differentially private estimates of the sample mean of data values in a fixed Hexagon And Timeslot.

The dataset of interest

IUDX
INDIA URBAN DATA EXCHANGE

Central Catalogue

Realtime Bus Transit Information from Surat City

Publishes realtime information like bus position, ETA, planned trip schedules and bus occupancy level of public transit buses in Surat city.

Instance	Surat
Publisher	Surat Municipal Corporation
Resources	3

Resources

PRIVATE

Bus Position and ETA of Public Transit Buses in Surat City ★★★★★ 4.80

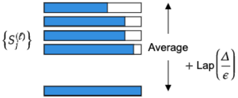
The bus position and ETA information of public transit buses at an interval of 10 seconds in Surat city.

Download sample data | View sample data | Query data | View Latest Data

[{"vehicle_label": "M22", "last_stop_id": "4022", "route_id": "11U", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": "08:29:25", "location": {"coordinates": [72.848299, 21.146356], "type": "Point"}, "speed": 17.0, "observationDateTime": "2021-05-08T08:29:45+05:30", "trip_id": "16004236", "license_plate": "GJ05BX3216", "trip_delay": 36, "actual_trip_start_time": "2021-05-08T08:20:46+05:30", "trip_direction": "UP"}, {"vehicle_label": "M28", "last_stop_id": "4028", "route_id": "11D", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": "08:29:21", "location": {"coordinates": [72.859182, 21.116843], "type": "Point"}, "speed": 18.0, "observationDateTime": "2021-05-08T08:29:51+05:30", "trip_id": "16004361", "license_plate": "GJ05BX3239", "trip_delay": 7, "actual_trip_start_time": "2021-05-08T08:26:51+05:30", "trip_direction": "DN"}, {"vehicle_label": "M49", "last_stop_id": "2041", "route_id": "11D", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": "08:28:39", "location": {"coordinates": [72.833627, 21.178609], "type": "Point"}, "speed": 17.0, "observationDateTime": "2021-05-08T08:29:50+05:30", "trip_id": "16004326", "license_plate": "GJ05BX3416", "trip_delay": 127, "actual_trip_start_time": "2021-05-08T08:12:48+05:30", "trip_direction": "DN"}, {"vehicle_label": "M53", "last_stop_id": "4027", "route_id": "11U", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": "08:24:55", "location": {"coordinates": [72.860109, 21.114961], "type": "Point"}, "speed": 29.0, "observationDateTime": "2021-05-08T08:26:29+05:30", "trip_id": "16004191", "license_plate": "GJ05BX3438", "trip_delay": 131, "actual_trip_start_time": "2021-05-08T08:09:07+05:30", "trip_direction": "UP"}, {"vehicle_label": "M60", "last_stop_id": "2047", "route_id": "11U", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": "08:29:02", "location": {"coordinates": [72.847256, 21.149716], "type": "Point"}, "speed": 38.0, "observationDateTime": "2021-05-08T08:29:47+05:30", "trip_id": "16004281", "license_plate": "GJ05BX3508", "trip_delay": 26, "actual_trip_start_time": "2021-05-08T08:21:28+05:30", "trip_direction": "UP"}, {"vehicle_label": "M69", "last_stop_id": "4031", "route_id": "11D", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": null, "location": {"coordinates": [72.865129, 21.103205], "type": "Point"}, "speed": 0.0, "observationDateTime": "2021-05-08T08:29:57+05:30", "trip_id": "16004343", "license_plate": "GJ05BX3862", "trip_delay": 1212, "actual_trip_start_time": "2021-05-08T08:18:34+05:30", "trip_direction": "DN"}, {"vehicle_label": "M22", "last_stop_id": "4022", "route_id": "11U", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": "08:29:25", "location": {"coordinates": [72.848551, 21.144395], "type": "Point"}, "speed": 13.0, "observationDateTime": "2021-05-08T08:30:10+05:30", "trip_id": "16004236", "license_plate": "GJ05BX3216", "trip_delay": 37, "actual_trip_start_time": "2021-05-08T08:20:46+05:30", "trip_direction": "UP"}, {"vehicle_label": "M28", "last_stop_id": "4028", "route_id": "11D", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": "08:29:21", "location": {"coordinates": [72.85876, 21.117956], "type": "Point"}, "speed": 34.0, "observationDateTime": "2021-05-08T08:30:11+05:30", "trip_id": "16004361", "license_plate": "GJ05BX3239", "trip_delay": 3, "actual_trip_start_time": "2021-05-08T08:26:51+05:30", "trip_direction": "DN"}, {"vehicle_label": "M49", "last_stop_id": "2041", "route_id": "11D", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": "08:28:39", "location": {"coordinates": [72.832844, 21.17985], "type": "Point"}, "speed": 36.0, "observationDateTime": "2021-05-08T08:30:10+05:30", "trip_id": "16004326", "license_plate": "GJ05BX3416", "trip_delay": 125, "actual_trip_start_time": "2021-05-08T08:12:48+05:30", "trip_direction": "DN"}, {"vehicle_label": "M53", "last_stop_id": "4027", "route_id": "11U", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": "08:24:55", "location": {"coordinates": [72.860109, 21.114961], "type": "Point"}, "speed": 29.0, "observationDateTime": "2021-05-08T08:26:29+05:30", "trip_id": "16004191", "license_plate": "GJ05BX3438", "trip_delay": 131, "actual_trip_start_time": "2021-05-08T08:09:07+05:30", "trip_direction": "UP"}, {"vehicle_label": "M60", "last_stop_id": "2047", "route_id": "11U", "id": "ee001002-ea73-48c4-bea0-453c4c956365", "last_stop_arrival_time": "08:29:02", "location": {"coordinates":

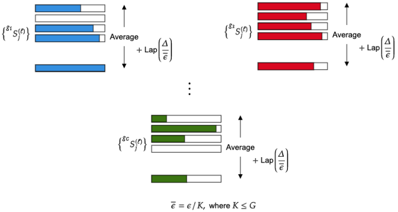
Our contributions

Single Grid Mean Release



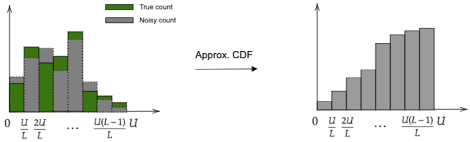
Algorithms; worst-case error

Multiple Grid (Mean,Var) Release



Gains in $K(< G)$ for fixed worst-case estimation error

Approximate CDF Release



Optimal tree-based mechanisms; optimal post-processing for consistency

Differentially Private Sample Mean Release for a Single Grid/HAT

Preliminaries: Single grid/HAT

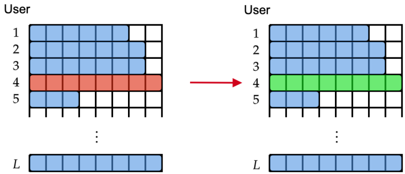
- Let L be the number of users in the HAT and let $\{m_\ell : \ell \in [L]\}$ be the collection of numbers of user contributions.
- We define $m_\star := \min_\ell m_\ell$ and $m^\star := \max_\ell m_\ell$.
- Each user ℓ contributes speed samples $S^{(\ell)} := \left\{S_j^{(\ell)} : j \in [m_\ell]\right\}$, where each sample lies in $[0, U]$; for us, $U = 65$ km/hr.
- Our dataset hence is $\mathcal{D} = \{(\ell, S^{(\ell)}) : \ell \in [L]\}$.
- We wish to release the sample mean

$$\mu(\mathcal{D}) := \frac{1}{\sum_\ell m_\ell} \cdot \sum_\ell \sum_{j=1}^{m_\ell} S_j^{(\ell)}$$

in a user-level differentially private manner.

User-level DP

- We say that two datasets $\mathcal{D}_1, \mathcal{D}_2$ are user-level neighbours if they differ in the sample values of a single user.

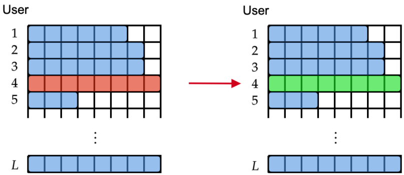


- A mechanism M is user-level ε-DP if for every pair of datasets $\mathcal{D}_1, \mathcal{D}_2$ that are user-level neighbours, and for every (measurable) Y,

$$e^{-\varepsilon}\Pr[M(\mathcal{D}_2) \in Y] \leq \Pr[M(\mathcal{D}_1) \in Y] \leq e^{\varepsilon}\Pr[M(\mathcal{D}_2) \in Y].$$

User-level DP

- We say that two datasets $\mathcal{D}_1, \mathcal{D}_2$ are user-level neighbours if they differ in the sample values of a single user.



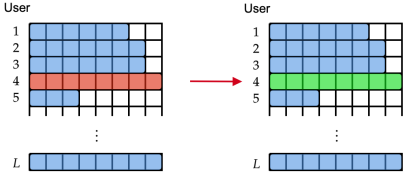
- A mechanism M is user-level ε-DP if for every pair of datasets $\mathcal{D}_1, \mathcal{D}_2$ that are user-level neighbours, and for every (measurable) Y,

$$e^{-\varepsilon} \Pr[M(\mathcal{D}_2) \in Y] \leq \Pr[M(\mathcal{D}_1) \in Y] \leq e^{\varepsilon} \Pr[M(\mathcal{D}_2) \in Y].$$

... Think of $e^{\varepsilon} \approx 1 + \varepsilon$, for $\varepsilon > 0$ small

User-level DP

- We say that two datasets $\mathcal{D}_1, \mathcal{D}_2$ are user-level neighbours if they differ in the sample values of a single user.



- A mechanism M is user-level ε-DP if for every pair of datasets $\mathcal{D}_1, \mathcal{D}_2$ that are user-level neighbours, and for every (measurable) Y,

$$e^{-\varepsilon} \Pr[M(\mathcal{D}_2) \in Y] \leq \Pr[M(\mathcal{D}_1) \in Y] \leq e^{\varepsilon} \Pr[M(\mathcal{D}_2) \in Y].$$

- Informally, a user-level DP mechanism ensures statistical indistinguishability of its outputs when a single user changes his/her samples.

Achieving user-level DP: the Laplace mechanism - I

- Given a function $f : \mathcal{D} \to \mathbb{R}$ (say, the sample mean), we define its user-level sensitivity to be

$$\Delta_f := \max_{\mathcal{D}_1, \mathcal{D}_2 \text{ u-l nbrs.}} |f(\mathcal{D})_1 - f(\mathcal{D}_2)|.$$

As an example, for our dataset $\mathcal{D}$,

$$\Delta_\mu = \frac{Um^\star}{\sum_\ell m_\ell}.$$

Achieving user-level DP: the Laplace mechanism - I

- Given a function $f : \mathcal{D} \to \mathbb{R}$ (say, the sample mean), we define its user-level sensitivity to be

$$\Delta_f := \max_{\mathcal{D}_1, \mathcal{D}_2 \text{ u-l nbrs.}} |f(\mathcal{D})_1 - f(\mathcal{D}_2)|.$$

As an example, for our dataset $\mathcal{D}$,

$$\Delta_\mu = \frac{Um^\star}{\sum_\ell m_\ell}.$$

- The Laplace mechanism simply adds Laplacian noise (of the right std. dev.) to the function of interest:

$$M^{\text{Lap}}(\mathcal{D}) = f(\mathcal{D}) + Z,$$

where $Z \sim \text{Lap}(\Delta_f/\varepsilon)$.

For $X \sim \text{Lap}(b)$, $b > 0$, we have $f_X(x) = \frac{1}{2b}e^{-|x|/b}$, $x \in \mathbb{R}$.

Achieving user-level DP: the Laplace mechanism - II

The following theorem is well-known:

Theorem

The mechanism M^{Lap} is user-level ε-DP.

Achieving user-level DP: the Laplace mechanism - II

The following theorem is well-known:

Theorem
The mechanism M^{Lap} is user-level ε-DP.

The following "utility" guarantee holds, via Laplacian tail bounds:

Theorem
For any $\mathcal{D}$ and any $\delta \in (0,1)$, we have

$$\Pr\left[\left|M^{Lap}(\mathcal{D}) - f(\mathcal{D})\right| \leq \frac{\Delta_f \ln(1/\delta)}{\varepsilon}\right] \geq 1 - \delta.$$

Achieving user-level DP: the Laplace mechanism - II

The following theorem is well-known:

Theorem
The mechanism M^{Lap} is user-level ε-DP.

- However, for real-world datasets, when $f = \mu$, the std. dev. of noise $Z \sim \text{Lap}(\Delta_\mu/\varepsilon)$ added is

$$\sigma_Z = \frac{\sqrt{2}\Delta_\mu}{\varepsilon} = \frac{\sqrt{2}Um^\star}{\varepsilon \cdot \sum_\ell m_\ell},$$

which is large when either U or $m^\star$ is large.

Achieving user-level DP: the Laplace mechanism - II

The following theorem is well-known:

Theorem
The mechanism M^{Lap} is user-level ε-DP.

- However, for real-world datasets, when $f = \mu$, the std. dev. of noise $Z \sim \text{Lap}(\Delta_\mu/\varepsilon)$ added is

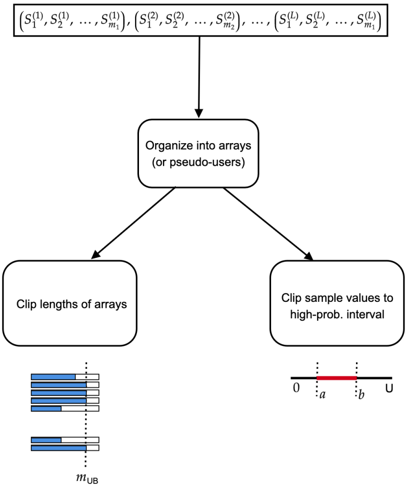
$$\sigma_Z = \frac{\sqrt{2}\Delta_\mu}{\varepsilon} = \frac{\sqrt{2}Um^\star}{\varepsilon \cdot \sum_\ell m_\ell},$$

which is large when either U or $m^\star$ is large.

We attempt to reduce σ_Z by fine-tuning mechanisms from the literature and by introducing novel choices of subroutines.

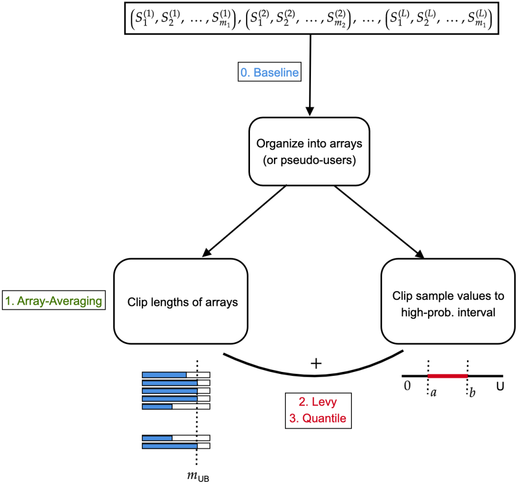
Our approach

We design three ε-DP mechanisms that perform two kinds of operations:

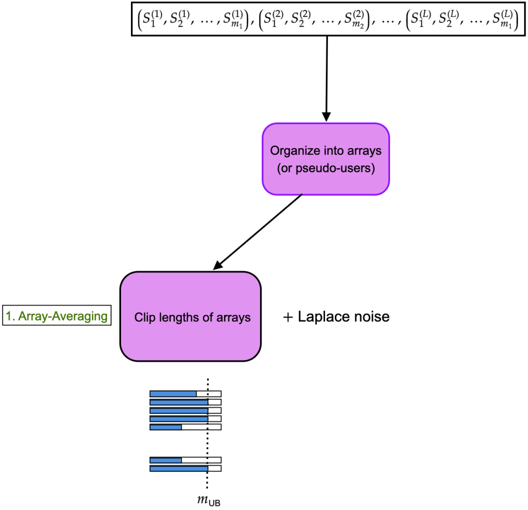


Our approach

We design three ε-DP mechanisms that perform two kinds of operations:



Prelude: Strategies for creation of pseudo-users



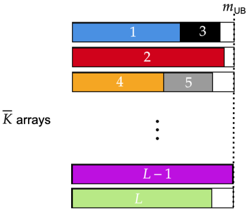
Prelude: Strategies for creation of pseudo-users

We first organize the speed samples into arrays/pseudo-users via a natural grouping strategy, called BestFit. Fix $m_{\mathrm{UB}} \in [m_\star, m^\star]$.

Prelude: Strategies for creation of pseudo-users

We first organize the speed samples into arrays/pseudo-users via a natural grouping strategy, called BestFit. Fix $m_{\mathrm{UB}} \in [m_\star, m^\star]$.

- BestFit: The first $\min\{m_\ell, m_{\mathrm{UB}}\}$ samples from each user $\ell \in \mathcal{L}$ are filled into that array of length m_{UB} that is filled the most.



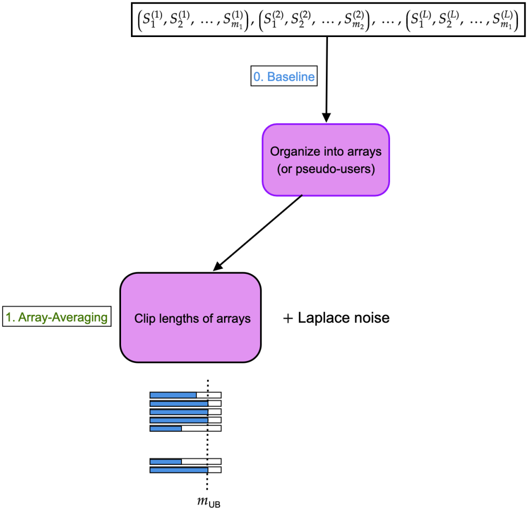
- The number of arrays created is

$$\overline{K} \geq K = \left\lfloor \frac{\sum_\ell \min\{m_\ell, m_{\mathrm{UB}}\}}{m_{\mathrm{UB}}} \right\rfloor.$$

- Each user "occupies" at most 1 array.

Array-Averaging

Array-Averaging adds suitable Laplace noise to the array means.



Array-Averaging

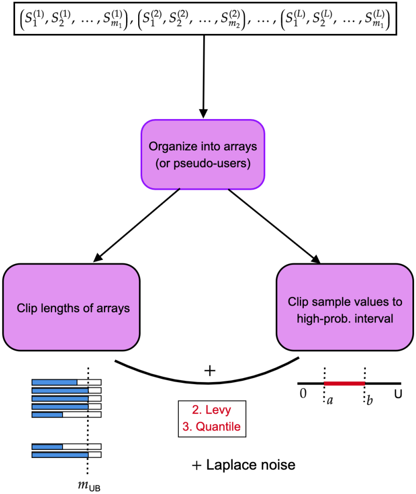
1. Group the samples in pseudo-users using BestFit.
2. Compute the means $\overline{A}_i$ of the sample values in each array A_i.
3. Return

$$M_{\text{arr,best}}(\mathcal{D}) = \frac{1}{\overline{K}} \sum_{i=1}^{\overline{K}} \overline{A}_i + \text{Lap}\left(\frac{U}{\overline{K}\varepsilon}\right).$$

Choosing $m_{\text{UB}} = \text{median}(\{m_\ell\})$ gives a factor-of-2 approximation of the lowest σ_Z to be added, under some regularity conditions.

Levy and Quantile

Levy and Quantile first clip the array means and then add Laplace noise.



Levy

1. Group the speed samples into pseudo-users using BestFit.
2. Privately estimate (with budget $\varepsilon/2$) a high-probability interval $[a, b]$ that is the $\left(\frac{1}{4}, \frac{3}{4}\right)$-interquantile interval [Levy et al. (2021)].
3. Project the array means $\overline{A}_i$ into the interval $[a, b]$.
4. Return

$$M_{\mathsf{Levy}}(\mathcal{D}) = \underbrace{\frac{1}{\overline{K}} \sum_{i=1}^{\overline{K}} \Pi_{[a,b]}(\overline{A}_i)}_{\mu_{\mathsf{Levy}}} + \mathsf{Lap}\left(\frac{2\Delta_{\mu_{\mathsf{Levy}}}}{\varepsilon}\right).$$

Levy

1. Group the speed samples into pseudo-users using BestFit.
2. Privately estimate (with budget $\varepsilon/2$) a high-probability interval $[a, b]$ that is the $\left(\frac{1}{4}, \frac{3}{4}\right)$-interquantile interval [Levy et al. (2021)].
3. Project the array means $\overline{A}_i$ into the interval $[a, b]$.
4. Return

$$M_{\text{Levy}}(\mathcal{D}) = \underbrace{\frac{1}{\overline{K}} \sum_{i=1}^{\overline{K}} \Pi_{[a,b]}(\overline{A}_i)}_{\mu_{\text{Levy}}} + \text{Lap}\left(\frac{2\Delta_{\mu_{\text{Levy}}}}{\varepsilon}\right).$$

Here, $\sigma_{Z,\text{Levy}} = \min\left\{\Theta\left(\frac{U}{\overline{K}\varepsilon}\sqrt{\frac{\log(\overline{K})}{m_{\text{UB}}}}\right), \frac{2\sqrt{2}U}{\overline{K}\varepsilon}\right\} \overset{\text{(potentially)}}{\leq} \sigma_{Z,\text{Arr}}.$

In our experiments, we attempt a heuristic minimization of the first term above by maximizing $\overline{K}\sqrt{m_{\text{UB}}}$ over m_{UB}.

Quantile

1. Group the speed samples into pseudo-users using BestFit.
2. Privately estimate (with budget $\varepsilon/2$) a high-probability interval $[a', b']$ that is either
   - the $(\frac{1}{10}, \frac{9}{10})$-interquantile interval [Smith (2011)] (FixedQuantile) or
   - an "optimized" ε-dependent interval [Amin et al. (2019)] (ε-DependentQuantile).
3. Project the array means $\overline{A}_i$ into the interval $[a', b']$.
4. Return

$$M_{\text{Levy}}(\mathcal{D}) = \underbrace{\frac{1}{\overline{K}} \sum_{i=1}^{\overline{K}} \Pi_{[a',b']}(\overline{A}_i)}_{f_{\text{Quantile}}} + \text{Lap}\left(\frac{2\Delta_{f_{\text{Quantile}}}}{\varepsilon}\right).$$

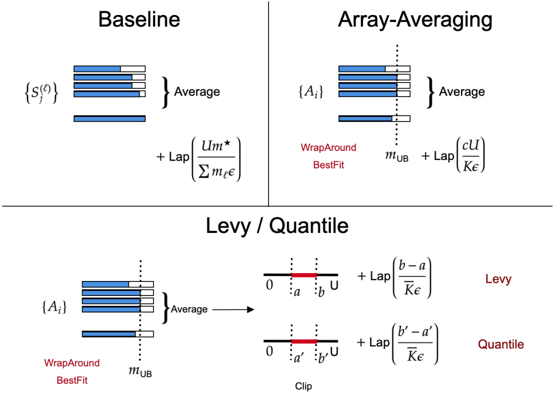
Quantile

1. Group the speed samples into pseudo-users using BestFit.
2. Privately estimate (with budget $\varepsilon/2$) a high-probability interval $[a', b']$ that is either
   - the $(\frac{1}{10}, \frac{9}{10})$-interquantile interval [Smith (2011)] (FixedQuantile) or
   - an "optimized" ε-dependent interval [Amin et al. (2019)] (ε-DependentQuantile).
3. Project the array means $\overline{A}_i$ into the interval $[a', b']$.
4. Return

$$M_{\text{Levy}}(\mathcal{D}) = \underbrace{\frac{1}{\overline{K}} \sum_{i=1}^{\overline{K}} \Pi_{[a',b']}(\overline{A}_i)}_{f_{\text{Quantile}}} + \text{Lap}\left(\frac{2\Delta_{f_{\text{Quantile}}}}{\varepsilon}\right).$$

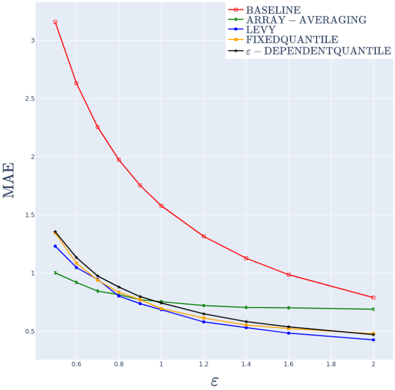
Here, $\sigma_{Z,\text{Quantile}} = \frac{2\sqrt{2}(b'-a')}{\overline{K}\varepsilon}$.

A quick recap



Experimental results I: Real-world data

- We evaluated the performance of our algorithms on real-world ITMS traffic data from an Indian city.
- We compare the mean absolute error (MAE) of our private algorithms vis-á-vis the true sample mean.



Experimental results II: Synthetic data

We then generate a synthetic dataset as follows. Fix a (large) integer λ.

1. **User contributions**:
   - **Sample scaling**: Set $\widehat{L} = L$ and $\widehat{m}_\ell = \lambda \cdot m_\ell$, for each $\ell \in \mathcal{L}$.
   - **User scaling**: Set $\widehat{L} = \lambda L$ and $\widehat{m}_{\lambda(\ell-1)+i} = m_\ell$, for $i \in [\lambda]$ and $\ell \in \mathcal{L}$.
2. **Data samples**:

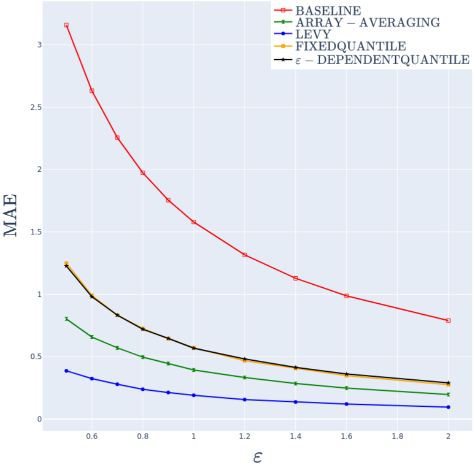
   Generate i.i.d. speed samples $\{\widehat{S}_j^{(\ell)} : \ell \in [\widehat{L}], j \in [\widehat{m}_\ell]\}$ such that

   $$\widehat{S}_j^{(\ell)} \sim \Pi_{[0,U]}(Z), \text{ where } Z \sim \mathcal{N}(\mu, \sigma^2),$$

   where μ, σ^2 are the (true) mean and variance of the ITMS samples.

Experimental results II: Synthetic data

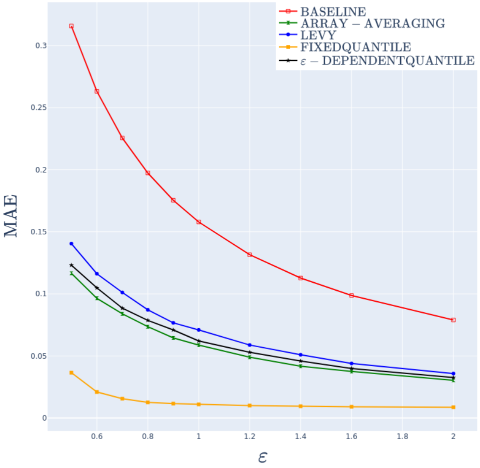
We compare the mean absolute error (MAE) of our private algorithms vis-á-vis the true sample mean.



Sample scaling

Experimental results II: Synthetic data

We compare the mean absolute error (MAE) of our private algorithms vis-á-vis the true sample mean.



User scaling

Some theoretical justification of performance trends

From our simulations, we see that

$$
\begin{aligned}
\text{Levy} &\succ \text{other alg.} \qquad (\text{Sample scaling}) \\
\text{(Fixed-)Quantile} &\succ \text{other alg.} \qquad (\text{User scaling})
\end{aligned}
$$

Some theoretical justification of performance trends

From our simulations, we see that

$$\text{Levy} \succ \text{other alg.} \qquad (\text{Sample scaling})$$
$$\text{(Fixed-)Quantile} \succ \text{other alg.} \qquad (\text{User scaling})$$

Theorem

Under sample scaling, using our choices of m_{UB} (median/heuristically optimized),

$$\sigma^{(s)}_{Z,Base} = \sigma_{Z,Base},\ \ \sigma^{(s)}_{Z,Arr} = \sigma_{Z,Arr},$$

and

$$\sigma^{(s)}_{Z,Levy} = \frac{1}{\sqrt{\lambda}} \cdot \sigma_{Z,Levy}.$$

Some theoretical justification of performance trends

From our simulations, we see that

$$\text{Levy} \succ \text{other alg.} \qquad (\text{Sample scaling})$$
$$\text{(Fixed-)Quantile} \succ \text{other alg.} \qquad (\text{User scaling})$$

Theorem

Under user scaling, using our choices of m_{UB} (median/heuristically optimized), for large enough scaling λ,

$$\sigma^{(u)}_{Z,Arr} < \min\left\{\sigma^{(u)}_{Z,Levy}, \sigma^{(u)}_{Z,\varepsilon\text{-}Dep.\text{-}Quantile}\right\} \quad w.h.p.,$$

if the exact sample-dependent quantiles are employed.

A second look at Array-Averaging: Error bounds

- We attempt to characterize exactly a measure of the total estimation error (clipping+privacy loss) in Array-Averaging.
- Since our real-world datasets $\mathcal{D}$ contain non-i.i.d. samples, we define a notion of the worst-case error, for a fixed $m = m_{\mathsf{UB}}$:

$$E^{(\varepsilon)}(m) := \max_{\mathcal{D}} E^{(\varepsilon)}(\mathcal{D}, m),$$

where

$$E^{(\varepsilon)}(\mathcal{D}, m) = \underbrace{|f_{\mathsf{Arr}}(\mathcal{D}, m) - f(\mathcal{D})|}_{\text{Clipping}} + \underbrace{\frac{\tilde{\Delta}_{f_{\mathsf{Arr}}}}{\varepsilon}}_{\text{Privacy}}.$$

Theorem

$$\max_{\mathcal{D}} |f_{Arr}(\mathcal{D}, m) - f(\mathcal{D})| = U \cdot \left(1 - \frac{\sum_\ell \min\{m_\ell, m\}}{\sum_\ell m_\ell}\right).$$

A second look at Array-Averaging: Error bounds

Let $\Gamma_\ell := \min\{m_\ell, m\}$. We then set

$$E^{(\varepsilon)} = \min_{m_\star \leq m \leq m^\star} E^{(\varepsilon)}(m)$$
$$= \min_{m_\star \leq m \leq m^\star} \left(U \cdot \left(1 - \frac{\sum_\ell \Gamma_\ell}{\sum_\ell m_\ell}\right) + \frac{Um}{\varepsilon \cdot \sum_{\ell=1}^{L} \Gamma_\ell} \right). \qquad (1)$$

Since

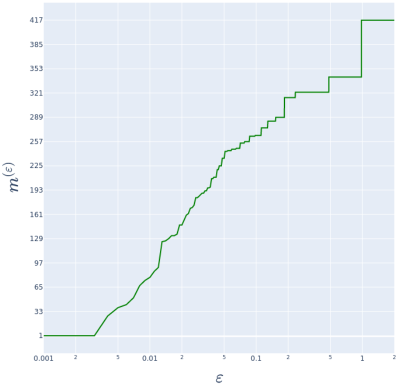
$$E^{(\varepsilon)} \geq \max_{\mathcal{D}'} \min_{m_\star \leq m \leq m^\star} E(\mathcal{D}', m) \geq \min_{m_\star \leq m \leq m^\star} E(\overline{\mathcal{D}}, m),$$

we have that $E^{(\varepsilon)}$ is an upper bound on the smallest error of Array-Averaging on any dataset $\overline{\mathcal{D}}$.

The optimization problem in (1) is non-convex in m. Let $m^{(\varepsilon)}$ be an optimizer.

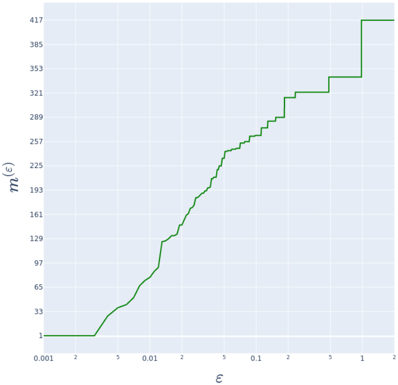
Some properties of $m^{(\varepsilon)}$

1. There exists
$$m^{(\varepsilon)} \in \{m_1, \ldots, m_L\}.$$
2. $m^{(\varepsilon)}$ is non-decreasing in ε.

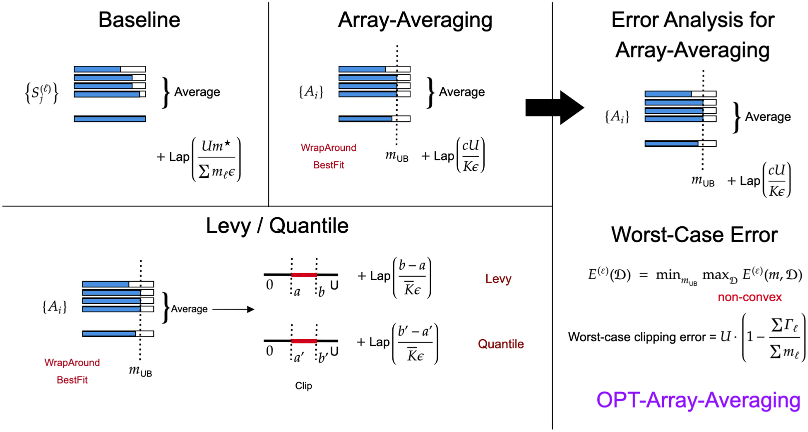


Some properties of $m^{(\varepsilon)}$

1. There exists
$$m^{(\varepsilon)} \in \{m_1, \dots, m_L\}.$$
2. $m^{(\varepsilon)}$ is non-decreasing in ε.



3. Let $\varepsilon_{\min} := \frac{m_\star}{L \cdot \sum_\ell m_\ell}$ and $\varepsilon_{\max} := \left(\frac{\sum_\ell m_\ell}{L m_\star}\right)^2$. Then, for $\varepsilon \le \varepsilon_{\min}$, we have $m^{(\varepsilon)} = m_\star$, and for $\varepsilon \ge \varepsilon_{\max}$, we have $m^{(\varepsilon)} = m^\star$.



All mechanisms in one figure

Differentially Private Sample Mean and Variance Release for Multiple Grids/HATs

Preliminaries: Multiple grids/HATs

- Let L be the total number of users and G be the total number of disjoint grids.

$$\{{}^{g}m_\ell : \ell \in [L], g \in [G]\} \leftarrow \text{numbers of user contributions.}$$

- Further, let

$${}^{g}\mathcal{L} = \{\ell : {}^{g}m_\ell > 0\} \text{ and } \mathcal{G}_\ell = \{g : {}^{g}m_\ell > 0\}$$

  and let ${}^{g}L$ and G_ℓ be their cardinalities.

- Analogous to the case earlier, let ${}^{g}S^{(\ell)} := \left\{{}^{g}S_j^{(\ell)} : j \in [{}^{g}m_\ell]\right\}$, be the data samples, all of which lie in $[0, U]$.

Preliminaries: Multiple grids/HATs

- Let L be the total number of users and G be the total number of disjoint grids.

$$\{{}^g m_\ell : \ell \in [L], g \in [G]\} \leftarrow \text{numbers of user contributions.}$$

- Further, let

$${}^g\mathcal{L} = \{\ell : \ {}^g m_\ell > 0\} \text{ and } \mathcal{G}_\ell = \{g : \ {}^g m_\ell > 0\}$$

  and let ${}^g L$ and G_ℓ be their cardinalities.

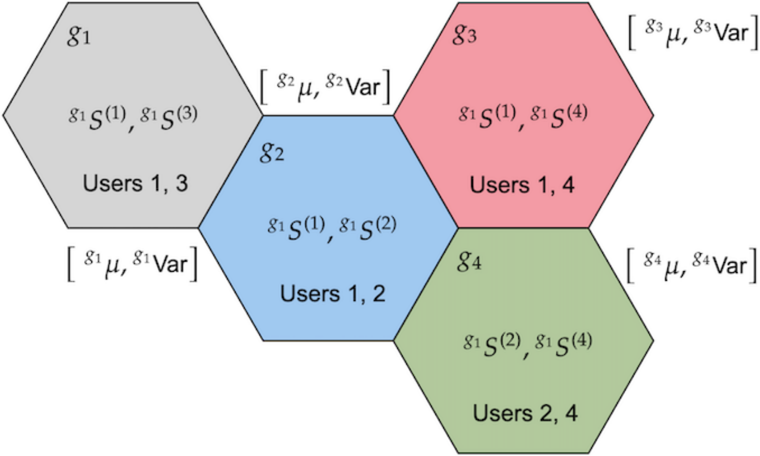
- Analogous to the case earlier, let ${}^g S^{(\ell)} := \left\{ {}^g S_j^{(\ell)} : j \in [{}^g m_\ell] \right\}$, be the data samples, all of which lie in $[0, U]$.

- We wish to release

$$f(\mathcal{D}) := \left({}^g f(\mathcal{D})\right)_g, \text{ where } {}^g f(\mathcal{D}) = \left[\underbrace{{}^g\mu(\mathcal{D})}_{\text{Mean in grid } g}, \underbrace{{}^g\mathrm{Var}(\mathcal{D})}_{\text{Var. in grid } g} \right]$$

  in a user-level ε-DP manner.

A pictorial depiction



Achieving user-level DP: the Laplace mechanism again

- As earlier, one can define an ε-user-level DP Laplace mechanism

$$M^{\mathrm{Lap}}(\mathcal{D}) = f(\mathcal{D}) + Z,$$

  where $Z = (Z_1, \ldots, Z_G)$, with $Z_i \overset{\text{i.i.d.}}{\sim} \mathrm{Lap}(\Delta_f/\varepsilon)$.

- Explicitly characterizing Δ_f is hard, owing to the contributions of users across grids.

Achieving user-level DP: the Laplace mechanism again

- As earlier, one can define an ε-user-level DP Laplace mechanism

$$M^{\mathsf{Lap}}(\mathcal{D}) = f(\mathcal{D}) + Z,$$

where $Z = (Z_1, \ldots, Z_G)$, with $Z_i \overset{\text{i.i.d.}}{\sim} \mathsf{Lap}(\Delta_f/\varepsilon)$.

- Explicitly characterizing Δ_f is hard, owing to the contributions of users across grids.

- A simple (and practical) solution: allocate a "privacy budget" ε to each grid, with

$$^{g}M_{\mu}^{\mathsf{Lap}}(\mathcal{D}) = {}^{g}\mu(\mathcal{D}) + {}^{g}Z_1,\ {}^{g}M_{\mathsf{Var}}^{\mathsf{Lap}}(\mathcal{D}) = {}^{g}\mathsf{Var}(\mathcal{D}) + {}^{g}Z_2.$$

Here, ${}^{g}Z_1 \sim \mathsf{Lap}(2\Delta_{{}^{g}\mu}/\varepsilon)$ and ${}^{g}Z_2 \sim \mathsf{Lap}(2\Delta_{{}^{g}\mathsf{Var}}/\varepsilon)$.

Achieving user-level DP: the Laplace mechanism again

- As earlier, one can define an ε-user-level DP Laplace mechanism

$$M^{\mathsf{Lap}}(\mathcal{D}) = f(\mathcal{D}) + Z,$$

  where $Z = (Z_1, \ldots, Z_G)$, with $Z_i \overset{\text{i.i.d.}}{\sim} \mathsf{Lap}(\Delta_f/\varepsilon)$.

- Explicitly characterizing Δ_f is hard, owing to the contributions of users across grids.

- A simple (and practical) solution: allocate a "privacy budget" ε to each grid, with

$${}^{g}M_{\mu}^{\mathsf{Lap}}(\mathcal{D}) = {}^{g}\mu(\mathcal{D}) + {}^{g}Z_1,\ {}^{g}M_{\mathsf{Var}}^{\mathsf{Lap}}(\mathcal{D}) = {}^{g}\mathsf{Var}(\mathcal{D}) + {}^{g}Z_2.$$

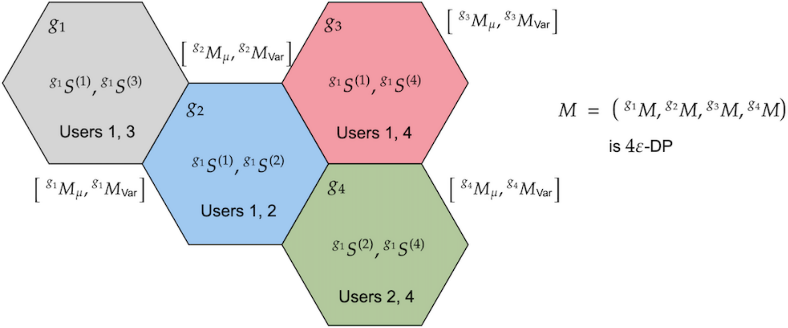
  Here, ${}^{g}Z_1 \sim \mathsf{Lap}(2\Delta_{{}^{g}\mu}/\varepsilon)$ and ${}^{g}Z_2 \sim \mathsf{Lap}(2\Delta_{{}^{g}\mathsf{Var}}/\varepsilon)$.

- By the Basic Composition Thm., the mechanism

$$M = \left(\left({}^{g}M_{\mu}^{\mathsf{Lap}}(\mathcal{D}), {}^{g}M_{\mathsf{Var}}^{\mathsf{Lap}}(\mathcal{D})\right) :\ g \in [G]\right)$$

  is $G\varepsilon$-user-level DP.

A vanilla bound in a picture



. . . but can we do better?

A simple observation

Indeed, in our problem setting, if ${}^{g}M$ are ε-user-level DP mechanisms for each grid g,

Theorem

The mechanism $M = ({}^{g}M : \; g \in [G])$ is user-level $\varepsilon \cdot \max_\ell G_\ell$-DP.

A simple observation

Indeed, in our problem setting, if ${}^{g}M$ are ε-user-level DP mechanisms for each grid g,

Theorem
The mechanism $M = ({}^{g}M : \ g \in [G])$ is user-level $\varepsilon \cdot \max_{\ell} G_{\ell}$-DP.

- We hence seek to reduce $\max_{\ell} G_{\ell}$, i.e., the largest number of grids any user "occupies".
- This is accomplished by completely suppressing contributions of selected users in selected grids, while maintaining the same worst-case error.

…but how is the error computed?

A notion of a worst-case error

- Suppose that we have mechanisms ${}^{g}M_{\theta} : \mathcal{D} \to \mathbb{R}^{d}$ for each grid g, to privately release statistics ${}^{g}\theta$, where

$$ {}^{g}M_{\theta}(\mathcal{D}) = {}^{g}\overline{\theta}(\mathcal{D}) + \overline{Z}, $$

with $\overline{Z} \sim \mathrm{Lap}^{\otimes d}\left(\Delta_{{}^{g}\overline{\theta}}/\varepsilon\right)$, for some estimator ${}^{g}\overline{\theta}$ of the true statistic ${}^{g}\theta$.

- We define the *worst-case* estimation error of ${}^{g}M_{\theta}$ as

$$ {}^{g}E := \sum_{i \in [d]} \underbrace{\max_{\mathcal{D} \in \mathrm{D}} \left| {}^{g}\theta_i(\mathcal{D}) - {}^{g}\overline{\theta}_i(\mathcal{D}) \right|}_{\text{Worst-case bias}} + \underbrace{\mathbb{E}[\|\overline{Z}\|]}_{\text{Privacy loss}} . $$

- Finally, we define the error metric E of $M_{\theta} = ({}^{g}M_{\theta} : \; g \in [G])$ as

$$ E := \max_{g \in [G]} {}^{g}E. $$

- We treat the error threshold of a dataset-unaware client as precisely this worst-case error E.

Exact error characterizations I: Sensitivities

- Focus on a single grid g.
- Consider estimators of sample mean and variance that are obtained by (arbitrarily) clipping user contributions.
- Fix a strategy Clip that retains any $\Gamma_\ell \in [0 : m_\ell]$ contributions of each user ℓ. Let $\Gamma^\star := \max_\ell \Gamma_\ell$.

Theorem

We have

$$\Delta_{\mu_{Clip}} = \frac{U\ \Gamma^\star}{\sum_{\ell=1}^{L} \Gamma_\ell} \quad \textit{and}$$

$$\Delta_{Var_{Clip}} = \begin{cases} \frac{U^2\ \Gamma^\star_\ell(\sum_\ell \Gamma_\ell - \Gamma^\star_\ell)}{\left(\sum_\ell \Gamma_\ell\right)^2}, & \textit{if } \sum_\ell \Gamma_\ell > 2\Gamma^\star, \\ \frac{U^2}{4}, & \textit{if } \sum_\ell \Gamma_\ell \leq 2\Gamma^\star \textit{ and } \sum_\ell \Gamma_\ell \textit{ is even}, \\ \frac{U^2}{4} \cdot \left(1 - \frac{1}{(\sum_\ell \Gamma_\ell)^2}\right), & \textit{if } \sum_\ell \Gamma_\ell \leq 2\Gamma^\star \textit{ and } \sum_\ell \Gamma_\ell \textit{ is odd.} \end{cases}$$

Exact error characterizations II: Clipping/Bias errors

Let E_μ and E_{Var} be the clipping errors via Clip.

Theorem

We have

$$E_\mu = U \cdot \left(1 - \frac{\sum_\ell \Gamma_\ell}{\sum_\ell m_\ell}\right).$$

Theorem

$E_{Var} = 0$ if $\Gamma_\ell = m_\ell$, for all $\ell \in [L]$. Furthermore, if $\sum_\ell \Gamma_\ell < \sum_\ell m_\ell$, we have

$$E_{Var} = \begin{cases} \frac{U^2 \cdot \sum_\ell \Gamma_\ell \cdot \sum_{\ell'} (m_{\ell'} - \Gamma_{\ell'})}{\left(\sum_\ell m_\ell\right)^2}, & \text{if } \sum_\ell m_\ell > 2 \sum_\ell \Gamma_\ell, \\ \frac{U^2}{4}, & \text{if } \sum_\ell m_\ell \leq 2 \sum_\ell \Gamma_\ell \text{ and } \sum_\ell m_\ell \text{ is even}, \\ \frac{U^2}{4} \cdot \left(1 - \frac{1}{(\sum_\ell m_\ell)^2}\right), & \text{otherwise.} \end{cases}$$

Some remarks

- There is a close relationship between the proofs for sensitivity and for the worst-case clipping error.
- The user-level sensitivity Δ_{Var} gives as a corollary the item-level sensitivity

  $$\Delta_{\mathrm{Var,item}} = \frac{U^2(L-1)}{L},$$

  obtained in [D'Orazio, Honaker, King (2015)].

  The techniques for computing Δ_{Var} are however much more involved.
- Via the worst-case bias and sensitivity expressions, we obtain expressions for the worst-case errors ${}^{g}E$:

  $${}^{g}E := \sum_{i\in[d]} \underbrace{\max_{\mathcal{D}\in \mathrm{D}} \left| {}^{g}\theta_i(\mathcal{D}) - {}^{g}\overline{\theta}_i(\mathcal{D}) \right|}_{\text{Worst-case bias}} + \underbrace{\mathbb{E}[\|\overline{Z}\|]}_{\text{Privacy loss}}.$$

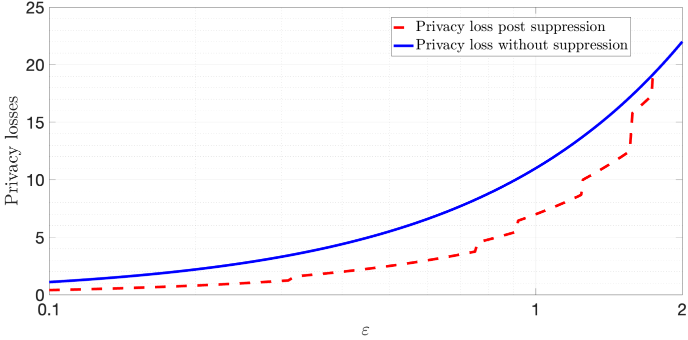
Goal: Can we reduce $\max_\ell G_\ell$ without hurting $E = \max_g {}^{g}E$?

The Chop-User algorithm for suppression

- For each grid g, compute the initial privacy loss errors $\mathbb{E}[\|{}^{g}\overline{Z}\|]$.
- Set $E_{\text{thresh}} = \max_g \mathbb{E}[\|{}^{g}\overline{Z}\|]$.
- Iterate the following until Stop:
  - For each user ℓ, identify the grid
    $$g(\ell) = \min_{g \in \mathcal{G}_\ell} {}^{g}E^{\text{post}},$$
    where ${}^{g}E^{\text{post}}$ is the error obtained by (potentially) suppressing ℓ in g, i.e., by setting ${}^{g}\Gamma_\ell = 0$ and ${}^{g}\Gamma_{\ell'} = {}^{g}m_{\ell'}$, for all $\ell' \in \mathcal{L}_g$.
  - If ${}^{g(\ell)}E^{\text{post}} > E_{\text{thresh}}$, then Stop.
  - Else, update $\mathcal{G}_\ell \leftarrow \mathcal{G}_\ell \setminus \{g(\ell)\}$ and ${}^{g}\mathcal{L} \leftarrow {}^{g}\mathcal{L} \setminus \{\ell\}$.
- Return $K = \max_\ell G_\ell$.

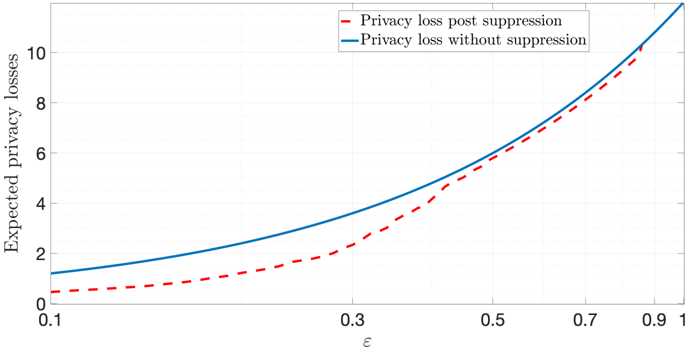
(!) Such a suppression-based approach will not work in the item-level DP setting.

Experimental results I: Real-world data



Plot of privacy loss under composition $K\varepsilon$ after execution of CLIP-USER on the real-world ITMS dataset, against the original privacy loss $\varepsilon \cdot \max_\ell G_\ell = 11\varepsilon$.

Experimental results II: Synthetic data



Plot of privacy loss under composition $K\varepsilon$ after execution of Clip-User on a synthetic dataset with a single heavy-hitter user, against the original privacy loss $\varepsilon \cdot \max_\ell G_\ell = 12\varepsilon$.

Clear gains in composition privacy loss are to be had for small (high-privacy) ε!

Summary

- Vanilla user-level DP mechanisms can be beaten by clipping-based mechanisms, with some fine-tuning.
- The simple Array-Averaging mechanism can be rigorously analyzed for worst-case error.
- Using exact expressions for worst-case errors, it is possible in practice to improve the composition privacy loss of mechanisms on disjoint grids, via suppression.

Our works

Mean Estimation with User-Level Privacy for Spatio-Temporal IoT Datasets

V. Arvind Rameshwar, Anshoo Tandon, Prajjwal Gupta, Aditya Vikram Singh, Novoneel Chakraborty, and Abhay Sharma

***Abstract*—This paper considers the problem of the private release of sample means of speed values from traffic datasets. Our key contribution is the development of user-level differentially private algorithms that incorporate carefully chosen parameter values to ensure low estimation errors on real-world datasets, while ensuring privacy. We test our algorithms on ITMS (Intelligent Traffic Management System) data from an Indian city, where the speeds of different buses are drawn in a potentially non-i.i.d. manner from an unknown distribution, and where the number of speed samples contributed by different buses is potentially different. We then apply our algorithms to large synthetic datasets, generated based on the ITMS data. Here, we provide theoretical justification for the observed performance trends, and also provide recommendations for the choices of algorithm subroutines that result in low estimation errors. Finally, we characterize the best performance of pseudo-user creation-based algorithms on worst-case datasets via a minimax approach; this then gives rise to a novel procedure for the creation of pseudo-users, which optimizes the worst-case total estimation error. The algorithms discussed in the paper are readily applicable to general spatio-temporal IoT datasets for releasing a differentially private mean of a desired value.**

I. Introduction

It is now well-understood that the release of even seemingly innocuous functions of a dataset that is not publicly available can result in the reconstruction of the identities of individuals (or users) in the dataset with alarming levels of accuracy (see, e.g., [1], [2]). A notable such reconstruction attack involved a somewhat naïvely anonymized database of taxi data, released by the Taxi and Limousine Commission of New York City [3], which was succesfully deanonymized [4], thereby revealing sensitive information about the taxi drivers. To alleviate concerns over such attacks, the framework of differential privacy (DP) was introduced in [5], which, informally speaking, guarantees the privacy of a *single* data sample, or equivalently, of users when each user contributes at most one sample. However, most real-world datasets, such as traffic databases, record multiple contributions from every user; a straightforward application of standard DP techniques achieves poor estimation errors, owing to the addition of a large amount of noise to guarantee privacy. Recent work on "user-level privacy" [6] however demonstrates the effectiveness of some new algorithms that guarantee much improved estimation error due to the additional privacy requirement for (a fixed) $m > 1$ samples per user.

In this paper, we provide algorithms, which draw on the research in [6], for ensuring user-level privacy in the context of releasing the sample means of speed records in traffic datasets. Clearly, it is desirable to keep the speed values of vehicles private, because they indirectly reflect the individual driving behaviour and might affect vehicle insurance premiums. Our algorithms for estimating the sample means of the data crucially rely on carefully chosen procedures that first create *pseudo-users*, or arrays, following [7], and then clip the number of speed samples contributed by each user and clip each speed sample to lie in a high-probability interval. These procedures are designed with the objective of controlling the "user-level sensitivity" of the sample mean that we are interested in.

We first emperically evaluate the performance of such algorithms (via their estimation errors) on real-world speed values from ITMS (Intelligent Traffic Management System) traffic data, supplied by IoT devices deployed in an Indian city. Here, the speeds of different buses are drawn in a potentially non-i.i.d. manner from an unknown distribution, and the number of speed samples contributed by different buses is potentially different. Next, we artificially generate a "large" synthetic dataset, using the statistics of the real-world ITMS data, with either a large number of users or a large number of samples contributed per user. We demonstrate, via extensive experiments, the effectiveness or the relative poor performance of the different algorithms we employ, in each case. In addition, we provide theoretical justification for the performance trends that we observe and recommendations for the choice of algorithm to be used on large real-world datasets. We mention that the results presented in this paper can be directly applied to Floating Car Data (FCD) (see, e.g., [8]) for

Presented in ISIT IT-TML, SPCOM;
arXiv: 2401.15906

Improving the Privacy Loss Under User-Level DP Composition for Fixed Estimation Error

V. Arvind Rameshwar, *Graduate Student Member, IEEE,* Anshoo Tandon, *Senior Member, IEEE*

Abstract

This paper considers the private release of statistics of several disjoint subsets of a datasets. In particular, we consider the ϵ-user-level differentially private release of sample means and variances of sample values in disjoint subsets of a dataset, in a potentially sequential manner. Traditional analysis of the privacy loss under user-level privacy due to the composition of queries to the disjoint subsets necessitates a privacy loss degradation by the total number of disjoint subsets. Our main contribution is an iterative algorithm, based on suppressing user contributions, which seeks to reduce the overall privacy loss degradation under a canonical Laplace mechanism, while not increasing the worst estimation error among the subsets. Important components of this analysis are our exact, analytical characterizations of the sensitivities and the worst-case bias errors of estimators of the sample mean and variance, which are obtained by clipping or suppressing user contributions. We test the performance of our algorithm on real-world and synthetic datasets and demonstrate improvements in the privacy loss degradation factor, for fixed estimation error. We also show improvements in the worst-case error across subsets, via a natural optimization procedure, for fixed numbers of users contributing to each subset.

Index Terms

User-level differential privacy, minimax error, composition, traffic datasets

Submitted to IEEE T-IT;
arXiv: 2405.06261

Our works

Optimal Tree-Based Mechanisms for Differentially Private Approximate CDFs

V. Arvind Rameshwar, Anshoo Tandon, and Abhay Sharma

***Abstract*—This paper considers the ε-differentially private (DP) release of an approximate cumulative distribution function (CDF) of the samples in a dataset. We assume that the true (approximate) CDF is obtained after lumping the data samples into a fixed number K of bins. In this work, we extend the well-known binary tree mechanism to the class of *level-uniform tree-based* mechanisms and identify ε-DP mechanisms that have a small ℓ_2-error. We identify optimal or close-to-optimal tree structures when either of the parameters, which are the branching factors or the privacy budgets at each tree level, are given, and when the algorithm designer is free to choose both sets of parameters. Interestingly, when we allow the branching factors to take on real values, under certain mild restrictions, the optimal level-uniform tree-based mechanism is obtained by choosing equal branching factors *independent* of K, and equal privacy budgets at all levels. Furthermore, for selected K values, we explicitly identify the optimal *integer* branching factors and tree height, assuming equal privacy budgets at all levels. Finally, we describe general strategies for improving the private CDF estimates further, by combining multiple noisy estimates and by post-processing the estimates for consistency.**

I. Introduction

It is now well-understood that the release of even seemingly innocuous functions of a dataset that is not publicly available can result in the reconstruction of the identities of individuals (or users) in the dataset with alarming levels of accuracy (see, e.g., [1], [2]). To alleviate concerns over such attacks, the framework of differential privacy (DP) was introduced in [3], which guarantees the privacy of any single sample. Subsequently, several works (see the surveys [4], [5] for references) have sought to design DP mechanisms or algorithms for the provably private release of statistics such as the mean, variance, counts, and histograms, resulting in the widespread adoption of DP for private data mining and analysis [6], [7].

In this work, we consider the fundamental problem of the DP release of (approximate) cumulative distribution functions been only few works that seek to optimize the parameters of such mechanisms to achieve low errors. In particular, the works [12], [13] consider variants of the well-known binary tree mechanism and suggest choices of the tree branching factor that achieve low errors using somewhat unnatural error metrics and asymptotic analysis. On the other hand, in the context of continual counting, given a fixed choice of parameters of (variants of) the binary tree mechanism, the works [14], [15] suggest techniques to optimally process the information in the nodes of the tree and use multiple noisy estimates of the same counts to obtain low-variance estimates of interval queries. We mention also that there have been several works (see, e.g., [16]–[18] and references therein) on matrix factorization-based mechanisms that result in the overall optimal error for general "linear queries" (see [5, Sec. 1.5] for the definition). In this work, we concentrate on the class of tree-based mechanisms and seek to optimize their parameters for low ℓ_2-error.

We first revisit some simple mechanisms for differentially private CDF release, via direct interval queries or histogram-based approaches, and explicitly characterize their ℓ_2-errors. While such results are well-known (see, e.g., [10]), they allow for comparisons with the errors of the broad class of "level-uniform tree-based" mechanisms – a class that we define in this work – that subsumes the binary tree mechanism and its previously studied variants [12], [13], [15]. First, by relaxing the integer constraint on the branching factors of the tree, we identify the optimal mechanism within this class, which turns out to be a simple tree-based mechanism with equal branching factors and privacy budgets at all levels. Furthermore, for sufficiently large K, the optimal branching factor, under a mild restriction on the branching factors, is a *constant* – roughly 17. We mention that, interestingly, [12] reports the optimal branching factor in a *subclass* of

Submitted to IEEE TIFS; arXiv: 2409.18573

Ongoing and future research directions

- Exploring the release of user-level DP data cluster centers for telecom inference tasks
- Investigating user-level DP mechanisms for general machine learning tasks
- Deriving exact expressions for the worst-case clipping errors and user-level sensitivities for other statistics of interest

Thank You!