AN INTRODUCTION TO DIFFERENTIALLY PRIVATE DATA ANALYSIS

Aurélien Bellet (Inria)

Séminaire de Probabilités et Statistiques, IMAG Montpellier
May 10, 2021
1. Context & motivation

2. Differential Privacy (DP)

3. Designing DP algorithms

4. DP without a trusted curator

5. Wrapping up
CONTEXT & MOTIVATION
Ability of an individual to **seclude themselves** or to **withhold information about themselves**

(“right to be let alone”)
**PRIVACY IN THE BIG DATA ERA**

- **Massive collection of personal data** by companies and public organizations, driven by the progress of data science and AI

- Data is **increasingly sensitive and detailed**: browsing history, purchase history, social network posts, speech, geolocation, health...

- It is sometimes **shared unknowingly and without a clear understanding of the risks**
• Improper disclosure of data can have adverse consequences for individuals:
  • Credentials
    • Examples: credit card number, home access code, passwords
    • Risks: stealing personal property
  • Identification information
    • Examples: name, bank information, biometric data
    • Risks: identity theft
  • Information about an individual
    • Examples: medical status, religious beliefs, political opinions, sexual preferences
    • Risks: discrimination, blackmailing, unsolicited micro-targeting, public shame...

• Some of these risks can affect anyone (even if they think they have “nothing to hide”) and without individuals knowing it (cf. Cambridge Analytica scandal)
• There is increasing regulation to address privacy-related harms related to the collection, use and release of personal data
  • General regulations (e.g., adoption of GDPR by the EU in 2018)
  • Sector- and context-specific regulations, e.g. in health, education, research, finance...

• Privacy has a cost on the utility of the analysis, but ideally it should not destroy it

• One of the main goals of privacy research is to find good trade-offs between utility and privacy so we can better protect individuals and unlock new applications
- **Goal:** achieve utility while preserving privacy (conflicting objectives!)
- This is separate from security concerns (e.g., unauthorized access to the system)
DATA “ANONYMIZATION” IS NOT SAFE

<table>
<thead>
<tr>
<th>Name</th>
<th>Birth date</th>
<th>Zip code</th>
<th>Gender</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ewen Jordan</td>
<td>1993-09-15</td>
<td>13741</td>
<td>M</td>
<td>Asthma</td>
</tr>
<tr>
<td>Lea Yang</td>
<td>1999-11-07</td>
<td>13440</td>
<td>F</td>
<td>Type-1 diabetes</td>
</tr>
<tr>
<td>William Weld</td>
<td>1945-07-31</td>
<td>02110</td>
<td>M</td>
<td>Cancer</td>
</tr>
<tr>
<td>Clarice Mueller</td>
<td>1950-03-13</td>
<td>02061</td>
<td>F</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

- **Anonymization**: removing **personally identifiable information** before publishing data
- First solution: **strip attributes that uniquely identify an individual** (e.g., name, social security number...)
DATA “ANONYMIZATION” IS NOT SAFE

<table>
<thead>
<tr>
<th>Name</th>
<th>Birth date</th>
<th>Zip code</th>
<th>Gender</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ewen Jordan</td>
<td>1993-09-15</td>
<td>13741</td>
<td>M</td>
<td>Asthma</td>
</tr>
<tr>
<td></td>
<td>1999-11-07</td>
<td>13440</td>
<td>F</td>
<td>Type-1 diabetes</td>
</tr>
<tr>
<td></td>
<td>1945-07-31</td>
<td>02110</td>
<td>M</td>
<td>Cancer</td>
</tr>
<tr>
<td></td>
<td>1950-03-13</td>
<td>02061</td>
<td>F</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

• **Anonymization**: removing personally identifiable information before publishing data

• First solution: strip attributes that uniquely identify an individual (e.g., name, social security number...)

• Now we cannot know that William Weld has cancer!

• Or can we?
**DATA “ANONYMIZATION” IS NOT SAFE**

- **Problem**: susceptible to **linkage attacks**, i.e. uniquely linking a record in the anonymized dataset to an identified record in a public dataset

- For instance, an estimated 87% of the US population is uniquely identified by the combination of their gender, birthdate and zip code

- In the late 90s, L. Sweeney managed to re-identify the medical record of the governor of Massachusetts using a public voters list
DATA “ANONYMIZATION” IS NOT SAFE

| Name          | Birth date     | Zip code | Gender | Diagnosis       |...
|---------------|----------------|----------|--------|-----------------|------------------------|
|               | 1993-09-15     | 13741    | M      | Asthma          |...
|               | 1999-11-07     | 13440    | F      | Type-1 diabetes |...
|               | 1945-07-31     | 02110    | M      | Cancer          |...
|               | 1950-03-13     | 02061    | F      | Cancer          |...

- Second solution: \(k\)-anonymity [Sweeney, 2002]
  1. Define a set of attributes as quasi-identifiers (QIs)
  2. Suppress/generalize attributes and/or add dummy records to make every record in the dataset indistinguishable from at least \(k - 1\) other records with respect to QIs
DATA “ANONYMIZATION” IS NOT SAFE

<table>
<thead>
<tr>
<th>Quasi identifiers</th>
<th>Sensitive attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Age</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Ewen Jordan</td>
<td>20-30</td>
</tr>
<tr>
<td></td>
<td>20-30</td>
</tr>
<tr>
<td></td>
<td>70-80</td>
</tr>
<tr>
<td></td>
<td>70-80</td>
</tr>
</tbody>
</table>

- Second solution: *k*-anonymity [Sweeney, 2002]
  1. Define a set of attributes as quasi-identifiers (QIs)
  2. Suppress/generalize attributes and/or add dummy records to make every record in the dataset indistinguishable from at least *k* – 1 other records with respect to QIs

- Better now?

- No! Can still infer that W. Weld has cancer (everyone in the group has cancer)
variants of $k$-anonymity ($t$-closeness, $\ell$-diversity) try to address the previous issue but require to modify the original data even more, which often destroys utility.

In high-dimensional and sparse datasets, any combination of attributes is a potential PII that can be exploited using appropriate auxiliary knowledge.

- De-anonymization of Netflix dataset protected with $k$-anonymity using a few public ratings from IMDB [Narayanan and Shmatikov, 2008]
- De-anonymization of Twitter graph using Flickr [Narayanan and Shmatikov, 2009]
- 4 spatio-temporal points uniquely identify most people [de Montjoye et al., 2013]

**Conclusion**: data cannot be fully anonymized AND remain useful.
Queries about specific individuals cannot be safely answered with accuracy. But how about aggregate statistics about many individuals?

**Problem 1:** differencing attacks, i.e. combining aggregate queries to obtain precise information about specific individuals (note: this can be hard to detect)
- Average salary in a company before and after an employee joins

**Problem 2:** membership inference attacks, i.e. inferring presence of known individual in a dataset from (high-dimensional) aggregate statistics
- Statistics about genomic variants [Homer et al., 2008]
• Machine Learning (ML) models are elaborate kinds of aggregate statistics!

• As such, they are susceptible to membership inference attacks, i.e. inferring the presence of a known individual in the training set

• For instance, one can exploit the confidence in model predictions [Shokri et al., 2017]
• ML models are also susceptible to reconstruction attacks, i.e. inferring some of the points used to train the model

• For instance, one can run differencing attacks on ML models [Paige et al., 2020]
ORDINARY FACTS ARE NOT ALWAYS SAFE

• As hinted to before, revealing ordinary facts may also be problematic if an individual is followed over time

• Example: Alice buys bread every day for 20 years and then stops

• An analyst might conclude that Alice has been diagnosed with type 2 diabetes

• This may be wrong, but in any case Alice could be harmed (e.g., charged with higher insurance premiums)
1. **Auxiliary knowledge**: we need to be robust to whatever knowledge the adversary may have, since we cannot predict what an adversary knows or might know in the future.

2. **Multiple analyses**: we need to be able to track how much information is leaked when asking several questions about the same data, and avoid catastrophic leaks.
DIFFERENTIAL PRIVACY (DP)
• $\mathcal{X}$: abstract data domain

• Dataset $D \in \mathcal{X}^n$: multiset of $n$ elements (records, or rows) from $\mathcal{X}$

• Can also see a dataset as a histogram: $D \in \mathbb{N}^{\mathcal{X}}$

• We say that two datasets $D, D' \in \mathbb{N}^{\mathcal{X}}$ are neighboring if $\|D - D'\|_1 \leq 1$ (i.e., they differ on at most one record)

• Note: a common variant considers pairs of datasets $D, D' \in \mathcal{X}^n$ of same size which differ on one record (i.e., replacing instead adding/removing one record)
Differential Privacy

- Neighboring datasets $D = \{ x_1, x_2, \ldots, x_n \}$ and $D' = \{ x_1, x_3, \ldots, x_n \}$

- Requirement: $A(D)$ and $A(D')$ should have “close” distribution

(Figure inspired from R. Bassily)
**Definition (Differential privacy [Dwork et al., 2006])**

Let $\varepsilon > 0$ and $\delta \in (0, 1)$. A randomized algorithm $A$ is $(\varepsilon, \delta)$-differentially private (DP) if for all datasets $D, D' \in \mathbb{N}^{\mathcal{X}}$ such that $\|D - D'\|_1 \leq 1$ and for all $S \subseteq \mathcal{O}$:

$$\Pr[A(D) \in S] \leq e^\varepsilon \Pr[A(D') \in S] + \delta,$$

where the probability space is over the coin flips of $A$.

- DP is a *property of the analysis*, not of a particular output
- A non-trivial differentially private algorithm *must be randomized*
- For meaningful guarantees
  - $\delta$ should be $o(1/n)$
  - Generally recommend $\varepsilon \leq 1$ but concrete guarantees depend a lot on the use-case [Abowd, 2018] [Garfinkel et al., 2018] [Jayaraman and Evans, 2019]
• DP guarantees are intrinsically robust to arbitrary auxiliary knowledge
  • Knowledge of all the dataset except one record
  • All external sources of knowledge, present and future

• The algorithm $A$ can be public: only the randomness needs to remain hidden
  • A key requirement of modern security ("security by obscurity" has long been rejected)
  • Allows to openly discuss the algorithms and their guarantees
Theorem (Postprocessing)

Let $A : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{O}$ be $(\varepsilon, \delta)$-DP and let $f : \mathcal{O} \rightarrow \mathcal{O}'$ be an arbitrary (randomized) function. Then

$$f \circ A : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{O'}$$

is $(\varepsilon, \delta)$-DP.

- “Thinking about” the output of a differentially private algorithm cannot make it less differentially private
- This holds regardless of attacker strategy and computational power
PROPERTIES OF DP: SEQUENTIAL COMPOSITION

Theorem (Simple composition)
Let $A_1, \ldots, A_K$ be such that $A_k$ satisfies $(\varepsilon_k, \delta_k)$-DP. For any dataset $D$, let $A$ be such that $A(D) = (A_1(D), \ldots, A_K(D))$. Then $A$ is $(\varepsilon, \delta)$-DP with $\varepsilon = \sum_{k=1}^{K} \varepsilon_k$ and $\delta = \sum_{k=1}^{K} \delta_k$.

Theorem (Advanced composition)
Let $\varepsilon, \delta, \delta' > 0$. If $A_k$ satisfies $(\varepsilon, \delta)$-DP, then $A_{\text{adap}}$ is $(\varepsilon', K\delta + \delta')$-DP with

$$\varepsilon' = \sqrt{2K\ln(1/\delta')}\varepsilon + K\varepsilon(e^\varepsilon - 1)$$

- Sequence of algorithms can be chosen adaptively
- This allows to control the cumulative privacy loss over multiple analyses run on the same dataset, including complex multi-step algorithms
- This is worst-case: in specific cases one can do better (e.g., algorithms operating on distinct inputs as in marginal queries)
• DP has become a **gold standard metric of privacy** in fundamental science but is also being increasingly used in real-world deployments

• **Thousands of scientific papers** in the fields of privacy, security, databases, data mining, machine learning...

• DP is deployed for **computing/releasing statistics** (including by tech giants...):
  
  • Adoption by the US Census Bureau in 2020 [Abowd, 2018]
  • Telemetry in Google Chrome [Erlingsson et al., 2014]
  • Keyboard statistics in iOS and macOS [Differential Privacy Team, Apple, 2017]
  • Application usage statistics by Microsoft [Ding et al., 2017]

• Open source software for DP in machine learning: **TensorFlow Privacy, OpenMined**...
DESIGNING DP ALGORITHMS
HOW TO DESIGN DP ALGORITHMS?

Individuals (data subjects)

queries

answers
(ex: summary statistics, machine learning model)

Data users (ex: government, researchers, companies, or adversary)
• Suppose we want to compute a numeric function \( f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^K \) of a private dataset \( D \)

**Definition (Global \( \ell_p \) sensitivity)**

Let \( p \geq 1 \). The global \( \ell_p \) sensitivity of a query (function) \( f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^K \) is

\[
\Delta_p(f) = \max_{D, D' : \|D - D'\|_1 \leq 1} \|f(D) - f(D')\|_p
\]

• **Output perturbation**: use global sensitivity to calibrate noise added to the (non-private) query output
Algorithm: Laplace mechanism $\mathcal{A}_{\text{Lap}}(D, f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^K, \varepsilon)$

1. Compute $\Delta = \Delta_1(f)$
2. For $k = 1, \ldots, K$: draw $Y_k \sim \text{Lap}(\Delta / \varepsilon)$ independently for each $k$
3. Output $f(D) + Y$, where $Y = (Y_1, \ldots, Y_K) \in \mathbb{R}^K$

Theorem (DP guarantees for Laplace mechanism)

Let $\varepsilon > 0$ and $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^K$. The Laplace mechanism $\mathcal{A}_{\text{Lap}}(\cdot, f, \varepsilon)$ satisfies $\varepsilon$-DP.

- Utility guarantees follow from properties of Laplace distribution, for instance:

$$\mathbb{E}[\|\mathcal{A}_{\text{Lap}}(D, f, \varepsilon) - f(D)\|_1] \leq K\frac{\Delta_1(f)}{\varepsilon}$$
### Algorithm: Gaussian mechanism $\mathcal{A}_{\text{Gauss}}(D, f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^K, \varepsilon, \delta)$

1. Compute $\Delta = \Delta_2(f)$
2. For $k = 1, \ldots, K$: draw $Y_k \sim \mathcal{N}(0, \sigma^2)$ independently for each $k$, where $\sigma = \sqrt{\frac{2 \ln(1.25/\delta)}{\varepsilon}} \Delta$
3. Output $f(D) + Y$, where $Y = (Y_1, \ldots, Y_K) \in \mathbb{R}^K$

### Theorem (DP guarantees for Gaussian mechanism)

Let $\varepsilon, \delta > 0$ and $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^K$. The Gaussian mechanism $\mathcal{A}_{\text{Gauss}}(\cdot, f, \varepsilon, \delta)$ is $(\varepsilon, \delta)$-DP.

- Slightly weaker guarantee but Gaussian has useful properties which make it easier to analyze when used as building block in a more complex algorithm
• We have seen approaches based on output perturbation: $A(D) = f(D) + Y$

• This only works for numeric queries

• It does not work if the utility function is irregular (e.g., think about auctions)
NON-NUMERIC QUERIES

• We can instead consider queries \( f : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{O} \) with an abstract output space \( \mathcal{O} \)

• We have a score function (or utility function) representing the quality of each output

\[
s : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{O} \to \mathbb{R}
\]

**Definition (Sensitivity of score function)**

The sensitivity of a \( s : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{O} \to \mathbb{R} \) is

\[
\Delta(s) = \max_{o \in \mathcal{O}} \max_{D, D' : \|D - D'\|_1 \leq 1} |s(D, o) - s(D', o)|
\]
Algorithm: Exponential mechanism $\mathcal{A}_{\text{Exp}}(D, f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{O}, s : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{O} \rightarrow \mathbb{R}, \varepsilon)$

1. Compute $\Delta = \Delta(s)$
2. Output $o \in \mathcal{O}$ with probability:

$$\Pr[o] = \frac{\exp\left(\frac{s(D,o) \cdot \varepsilon}{2\Delta}\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\frac{s(D,o') \cdot \varepsilon}{2\Delta}\right)}$$

• Make high quality outputs \textit{exponentially} more likely, at a rate that depends on the sensitivity of the score and the privacy parameter

Theorem (DP guarantees for exponential mechanism)

Let $\varepsilon > 0, f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^K$ and $s : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{O} \rightarrow \mathbb{R}$. $\mathcal{A}_{\text{Exp}}(\cdot, f, s, \varepsilon)$ satisfies $\varepsilon$-DP.
APPLICATION: EMPIRICAL RISK MINIMIZATION

- $D = \{(x_i, y_i)\}_{i=1}^n$: training points drawn i.i.d. from distribution $\mu$ over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$

- Models $h_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ parameterized by $\theta \in \Theta \subseteq \mathbb{R}^p$

- $L(\theta; x, y)$: loss of model $h_\theta$ on data point $(x, y)$

- $\hat{R}(\theta; D) = \frac{1}{n} \sum_{i=1}^n L(\theta; x_i, y_i)$: empirical risk of model $h_\theta$

- **Empirical Risk Minimization (ERM)** consists in choosing the parameters

  $$\hat{\theta} \in \arg \min_{\theta \in \Theta} [F(\theta; D) := \hat{R}(\theta; D) + \lambda \psi(\theta)]$$
• Basic DP building blocks can be used to design differentially private ERM solvers

• Such a solver (optimization algorithm) must interact with the data only through DP mechanisms
NON-PRIVATE STOCHASTIC GRADIENT DESCENT (SGD)

- For simplicity, let us assume that \( \psi(\theta) = 0 \) (no regularization)
- Denote by \( \Pi_\Theta(\theta) = \arg\min_{\theta' \in \Theta} ||\theta - \theta'||_2 \) the projection operator onto \( \Theta \)

Algorithm: Non-private (projected) SGD

- Initialize parameters to \( \theta^{(0)} \in \Theta \)
- For \( t = 0, \ldots, T - 1 \):
  - Pick \( i_t \in \{1, \ldots, n\} \) uniformly at random
  - \( \theta^{(t+1)} \leftarrow \Pi_\Theta(\theta^{(t)} - \gamma_t \nabla L(\theta^{(t)}; x_{i_t}, y_{i_t})) \)
- Return \( \theta^{(T)} \)

- SGD is a natural candidate solver: simple, flexible, scalable, heavily used in ML
- How to design a DP version of SGD?
### Algorithm: Differentially Private SGD \( \mathcal{A}_{\text{DP-SGD}}(D, L, \varepsilon, \delta) \)

- **Initialize parameters to** \( \theta^{(0)} \in \Theta \) (must be independent of \( D \))
- **For** \( t = 0, \ldots, T - 1: \)
  - Pick \( i_t \in \{1, \ldots, n\} \) uniformly at random
  - \( \eta^{(t)} \leftarrow (\eta^{(t)}_1, \ldots, \eta^{(t)}_p) \in \mathbb{R}^p \) where each \( \eta^{(t)}_j \sim \mathcal{N}(0, \sigma^2) \) with \( \sigma = \frac{16l\sqrt{\ln(2/\delta)\ln(1.25T/\delta n)}}{n\varepsilon} \)
  - \( \theta^{(t+1)} \leftarrow \Pi_\Theta \left( \theta^{(t)} - \gamma_t \left( \nabla L(\theta^{(t)}; x_{i_t}, y_{i_t}) + \eta^{(t)} \right) \right) \)
- **Return** \( \theta^{(T)} \)

- **More data** (larger \( n \)) \( \rightarrow \) less noise added to each gradient
- **More iterations** (larger \( T \)) \( \rightarrow \) more noise added to each gradient

### Theorem (DP guarantees for DP-SGD)

Let \( \varepsilon \leq 1, \delta > 0 \). Let the loss function \( L(\cdot; x, y) \) be \( l \)-Lipschitz w.r.t. the \( \ell_2 \) norm for all \( x, y \in \mathcal{X} \times \mathcal{Y} \). Then \( \mathcal{A}_{\text{DP-SGD}}(\cdot, L, \varepsilon, \delta) \) is \((\varepsilon, \delta)\)-DP.
Sketch of proof.

- Recall that for a query with $\ell_2$ sensitivity $\Delta$, achieving $(\varepsilon', \delta')$ with the Gaussian mechanism requires to add noise with standard deviation $\sigma' = \frac{\sqrt{2 \ln(1.25/\delta')}}{\varepsilon'} \Delta$

- The loss function $L$ is $l$-Lipschitz, which implies that $\ell_2$-norm of gradients is bounded by $l$ and therefore $\Delta = 2l$

- Hence, with $\sigma = \frac{16l\sqrt{T \ln(2/\delta) \ln(1.25T/\delta n)}}{n \varepsilon}$, each noisy gradient is $(\frac{n \varepsilon}{4 \sqrt{2T \ln(2/\delta)}}, \frac{\delta n}{2T})$-DP

- Using privacy amplification by subsampling [Balle et al., 2018] allows to leverage the randomness in the choice of $i_t$: each noisy gradient is in fact $(\frac{\varepsilon}{2 \sqrt{2T \ln(2/\delta)}}, \frac{\delta}{2T})$-DP

- DP-SGD is an adaptive composition of $T$ DP mechanisms, so by advanced composition we obtain that it is $(\varepsilon, \delta)$-DP
Theorem (Utility guarantees for DP-SGD [Bassily et al., 2014])

Let $\Theta$ be a convex domain of diameter bounded by $R$, and let the loss function $L$ be convex and $l$-Lipschitz over $\Theta$. For $T = n^2$ and $\gamma_t = O(R/\sqrt{t})$, DP-SGD guarantees:

$$\mathbb{E}[F(\theta^{(T)})] - \min_{\theta \in \Theta} F(\theta) \leq O\left(\frac{lR\sqrt{p \ln(1/\delta)} \ln^{3/2}(n/\delta)}{n\varepsilon}\right).$$

- Proof: plug variance of stochastic gradients in analysis of SGD [Shamir and Zhang, 2013]
- The utility gap with respect to the non-private model reduces with $n$ at rate $\tilde{O}(1/n)$
- Privacy induces a larger cost for high-dimensional models
DP WITHOUT A TRUSTED CURATOR
Trusted curator model (also called global model or centralized model): $A$ is differentially private wrt dataset $D$

Untrusted curator model (also called local model or distributed model): Each $R_i$ is differentially private wrt record (or local dataset) $x_i$
As always, let $\mathcal{X}$ denote an abstract data domain.

A local randomizer $\mathcal{R} : \mathcal{X} \rightarrow \mathcal{Z}$ is a randomized function which maps an input $x \in \mathcal{X}$ to an output $z \in \mathcal{Z}$.

**Definition (Local Differential Privacy [Kasiviswanathan et al., 2008, Duchi et al., 2013])**

Let $\varepsilon > 0$ and $\delta \in (0, 1)$. A local randomizer algorithm $\mathcal{R}$ is $(\varepsilon, \delta)$-locally differentially private (LDP) if for all $x, x' \in \mathcal{X}$ and any possible output $z \in \mathcal{Z}$:

$$\Pr[\mathcal{R}(x) = z] \leq e^\varepsilon \Pr[\mathcal{R}(x') = z] + \delta.$$

- Equivalent to $(\varepsilon, \delta)$-DP for datasets of size 1
- LDP is a much stronger model than central DP: data analyst does not see raw data
- LDP allows participants to have plausible deniability even if the curator is compromised: they can deny having value $x$ on the basis of lack of evidence
Let $f$ be a public function from $\mathcal{X}$ to a bounded numeric range (say $f : \mathcal{X} \to [0, 1]$)

We want to compute an averaging query $\bar{f} = \frac{1}{n} \sum_{i=1}^{n} f(x_i)$

This is the key primitive needed in distributed/federated learning [Kairouz et al., 2019]

We can readily use the Laplace and Gaussian mechanisms: seeing each input as a dataset of size 1, we have

$$\Delta_1(f) = \max_{x,x'} |f(x) - f(x')| = 1,$$

and similarly $\Delta_2(f) = 1$

For instance, with the Laplace mechanism, we get an estimate of $\bar{f}$ with variance $2/n\varepsilon^2$
THE COST OF THE LOCAL MODEL

• There is a large utility gap between the central and the local model of DP: it is typically a factor of $O(1/\sqrt{n})$ in $\ell_1$ error (or $O(1/n)$ in $\ell_2$ error)

• In particular, for averaging queries
  • In the local model, we have seen that we get a variance of $O(1/n)$
  • In the central model, we can compute the exact $\bar{f}$ and add Laplace noise calibrated to its $\ell_1$ sensitivity $\Delta_1(\bar{f}) = 1/n$, hence we get a variance of $O(1/n^2)$

• This gap is known to be unavoidable [Chan et al., 2012]

• This restricts the usefulness of LDP to applications where $n$ is very large
Algorithm 1 GOPA protocol

Parameters: graph $G$, variances $\sigma^2_{\Delta}, \sigma^2_{\eta} \in \mathbb{R}^+$

for all neighboring parties $\{i,j\}$ in $G$ do
  $i$ and $j$ draw $y \sim \mathcal{N}(0, \sigma^2_{\Delta})$
  set $\Delta_{i,j} \leftarrow y$, $\Delta_{j,i} \leftarrow -y$

for each user $i$ do
  $i$ draws $\eta_i \sim \mathcal{N}(0, \sigma^2_{\eta})$
  $i$ reveals $f(\hat{x}_i) \leftarrow f(x_i) + \sum_{j \sim i} \Delta_{i,j} + \eta_i$

1. Neighbors $\{i,j\}$ in $G$ securely exchange pairwise-canceling Gaussian noise
2. Each user $i$ generates personal Gaussian noise
3. User $i$ reveals the sum of private value, pairwise and personal noise terms

- **Accurate**: the result $\hat{f} = \frac{1}{n} \sum_i f(\hat{x}_i)$ can match the accuracy of the centralized setting
- **Scalable**: it is sufficient for each user to communicate with $O(\log n)$ others
- **Robust**: it can handle some collusions, dropouts and malicious behavior
Most work on local DP focuses on statistics that are separable across individual users (sums, histograms...) [Bassily and Smith, 2015, Kulkarni et al., 2019, Bassily et al., 2017]

This is not the case when considering $U$-statistics (of degree 2):

$$U_{f,n} := \frac{2}{n(n-1)} \sum_{i<j} f(x_i, x_j)$$

where the pairwise function $f$ is called the kernel.

Examples of such statistics: sample variance, Gini mean difference, Kendall’s $\tau$, Wilcoxon Mann-Whitney hypothesis test, Area under the ROC Curve (AUC)...

Also used as risk measures in pairwise learning problems such as metric learning and bipartite ranking [Kar et al., 2013, Clémençon et al., 2016]

Computing $U$-statistics in LDP cannot generally be addressed by resorting to standard local randomizers due to the pairwise nature of the terms.
GENERIC LDP PROTOCOL FOR $u$-STATISTICS [Bell et al., 2020]

1. Discretize domain into $k$ bins
2. Local randomization: each user answers a random bin with prob. $\beta$
3. Estimation: Compute $U$-statistic on randomized answers and debias the result
1. Discretize domain into $k$ bins
1. Discretize domain into $k$ bins

2. Local randomization: each user answers a random bin with prob. $\beta$

3. Estimation: Compute $U$-statistic on randomized answers and debias the result
Theorem

For simplicity, assume bounded domain $\mathcal{X} = [0, 1]$ and kernel values $f(x, y) \in [0, 1]$ for all $x, y \in \mathcal{X}$. Let $\pi$ correspond to simple rounding, $\varepsilon > 0$, $k \geq 1$ and $\beta = k/(k + e^\varepsilon - 1)$. Then the algorithm satisfies $\varepsilon$-LDP. Furthermore:

- If $f$ is $L_f$-Lipschitz, then $\text{MSE}(\hat{U}_{f,n}) \leq \frac{1}{n(1-\beta)^2} + \frac{(1+\beta)^2}{2n(n-1)(1-\beta)^4} + \frac{L_f^2}{2k^2}$.
- If $d\mu/d\lambda$ is $L_\mu$-Lipschitz, then $\text{MSE}(\hat{U}_{f,n}) \leq \frac{1}{n(1-\beta)^2} + \frac{(1+\beta)^2}{2n(n-1)(1-\beta)^4} + \frac{4L_\mu^2}{k^2} + \frac{4L_\mu^4}{k^4}$.

Corollary

For $\varepsilon \leq 1$ and large enough $n$, taking $k = n^{1/4}\sqrt{L\varepsilon}$ leads to $\text{MSE}(\hat{U}_{f,n}) = O(L/\sqrt{n\varepsilon})$, where $L$ corresponds to $L_f$ or $L_\mu$ depending on the assumption.

- Sum of errors from randomized response and quantization
- See paper for other algorithms, e.g. for AUC on large discrete domains
Wrapping up
• Any personal information can be sensitive, and anonymization is hard

• Differential privacy provides a robust mathematical definition of privacy and a strong algorithmic framework allowing to design complex private algorithms

• When there is no trusted curator, DP can be deployed locally at the participants’ level so as to analyze data while keeping it decentralized and confidential

• There are lots of cool open problems at the intersection of privacy, algorithms, statistics and machine learning
    The U.S. Census Bureau Adopts Differential Privacy.
    In KDD.

    Privacy amplification by subsampling: tight analyses via couplings and divergences.
    In NeurIPS.

    Practical locally private heavy hitters.
    In NIPS.

    Local, private, efficient protocols for succinct histograms.
    In STOC.

    In FOCS.

    Private Protocols for U-Statistics in the Local Model and Beyond.
    In AISTATS.


Local Privacy and Statistical Minimax Rates.
In FOCS.

Calibrating noise to sensitivity in private data analysis.
In Theory of Cryptography (TCC).

Rappor: Randomized aggregatable privacy-preserving ordinal response.
In CCS.

Issues encountered deploying differential privacy.
In WPES@CCS.

Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays.
Evaluating Differentially Private Machine Learning in Practice.
In USENIX Security.

Advances and Open Problems in Federated Learning.

On the Generalization Ability of Online Learning Algorithms for Pairwise Loss Functions.
In ICML.

What Can We Learn Privately?
In FOCS.
Answering range queries under local differential privacy.
In SIGMOD.

Robust de-anonymization of large sparse datasets.
In IEEE Symposium on Security and Privacy (S&P).

De-anonymizing social networks.
In IEEE Symposium on Security and Privacy (S&P).

Reconstructing Genotypes in Private Genomic Databases from Genetic Risk Scores.
In International Conference on Research in Computational Molecular Biology RECOMB.

Non-interactive and information-theoretic secure verifiable secret sharing.
In CRYPTO.

Distributed Differentially Private Averaging with Improved Utility and Robustness to Malicious Parties.


• **Adversary**: proportion $1 - \rho$ of colluding malicious parties who observe all communications they take part in

• Denote by $H$ the set of honest-but-curious parties, and by $G^H$ the honest subgraph

• GOPA can achieve $(\varepsilon, \delta)$-DP for any $\varepsilon, \delta > 0$ for connected $G^H$ and large enough $\sigma^2_\eta, \sigma^2_\Delta$

• We show that $\sigma^2_\eta$ can be as small as in the centralized setting (matching its utility)

• We show that the required $\sigma^2_\Delta$ depends on the topology of $G^H$
Theorem (Case of random $k$-out graph)

Let $\varepsilon, \delta' \in (0, 1)$ and let:

- $G$ be obtained by letting all parties randomly choose $k = O(\log(\rho n)/\rho)$ neighbors
- $\sigma^2_{\eta}$ so as to satisfy $(\varepsilon, \delta)$-DP in the centralized (trusted curator) setting
- $\sigma^2_{\Delta} = O(\sigma^2_{\eta}|H|/k)$

Then GOPA is $(\varepsilon, \delta)$-differentially private for $\delta = O(\delta')$.

- Trusted curator utility with logarithmic number of messages per user
- Our theoretical results give practical values for $k$ and $\sigma^2_{\Delta}$
• Utility can be compromised by malicious parties tampering with the protocol (e.g., sending incorrect values to bias the outcome)

• It is impossible to force a user to give the “right” input (this also holds in the trusted curator setting)

• We enable each user $u$ to prove the following properties:

\[
f(x_i) \in [0, 1], \quad \forall i \in \{1, \ldots, n\}
\]

\[
\Delta_{i,j} = -\Delta_{j,i}, \quad \forall \{i,j\} \text{ neighbors in } G
\]

\[
\eta_i \sim \mathcal{N}(0, \sigma^2), \quad \forall i \in \{1, \ldots, n\}
\]

\[
f(\hat{x}_k) = f(x_k) + \sum_{j \sim i} \Delta_{i,j} + \eta_i, \quad \forall i \in \{1, \ldots, n\}
\]
• Parties publish an encrypted log of the computation using Pedersen commitments [Blum, 1983, Pedersen, 1991], which are additively homomorphic.

• Based on these commitments, parties prove that the computation was done correctly using zero knowledge proofs.

**Theorem (Informal)**

A user $i$ that passes the verification proves that $f(\hat{x}_i)$ was computed correctly. Additionally, by doing so, $i$ does not reveal any additional information about $x_i$.

• Costs per user remain linear in the number of neighbors.

• Can prove consistency across multiple runs on same/similar data.

• Can handle drop out.
Algorithm 2 LDP algorithm based on quantization and private histograms

Public parameters: Privacy budget \( \varepsilon \), quantization scheme \( \pi \), number of bins \( k \)

for each user \( i \in [n] \) do

Form quantized input \( \pi(x_i) \in [k] \)

For \( \beta = k / (k + e^\varepsilon - 1) \), generate \( \tilde{x}_i \in [k] \) s.t.

\[
P(\tilde{x}_i = i) = \begin{cases} 
1 - \beta & \text{for } i = \pi(x_i), \\
\beta/k & \text{for } i \neq \pi(x_i).
\end{cases}
\]

Send \( \tilde{x}_i \) to untrusted curator

return \( \hat{U}_{f,n} = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \hat{f}_A(\tilde{x}_i, \tilde{x}_j) \), where \( \hat{f}_A(\mathcal{R}(x_1), \mathcal{R}(x_2)) = (1 - \beta)^{-2}(e_{\mathcal{R}(x_1)} - b)^T A(e_{\mathcal{R}(x_2)} - b) \), \( A \in \mathbb{R}^{k \times k} \) with \( A_{ij} = f(i, j) \), and \( b = \frac{\beta}{k} \mathbf{1} \)