An Introduction to Trustworthy Machine Learning

Daniel Gatica-Perez Idiap Research Institute
Sina Sajadmanesh Idiap Research Institute
Ali Shahin Shamsabadi The Alan Turing Institute

International Artificial Intelligence Doctoral Academy (AIDA)
November 2022

Slides are adapted from Privacy Preserving Machine Learning course by Aurélien Bellet.

Outline

1. Why Differential Privacy?
2. Differential Privacy: Definition, Properties, and Mechanisms
3. Differentially Private Machine Learning
4. Rényi Differential Privacy
5. Hands-on Tutorial

Why Differential Privacy?

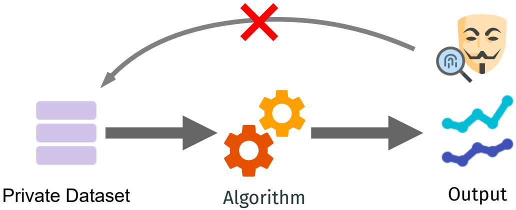
Anonymization Fiascos

- 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, 2008]
- 4 spatio-temporal points uniquely identify most people [De Montjoye et al., 2013]
- And many more ...

Removing identifiers and applying anonymization heuristics is not enough!

Private Data Analysis Setting

- An algorithm is executed on the private dataset and the output is publically released
  - E.g., an ML model is trained on a dataset of patients and the output is the parameters
- An adversary should not be able to learn much about the data by analyzing the output
  - Regardless of the adversary's side knowledge
  - In the worst case, all the records except one can be known to the adversary



What does it mean for an algorithm to be private?

*"An algorithm is private if **its output reveals no more** about an individual **than what was already known** about him/her before."*

What does it mean for an algorithm to be private?

"An algorithm is private if ***its output reveals no more*** *about an individual* ***than what was already known*** *about him/her before."*

- Not Correct!

What does it mean for an algorithm to be private?

"An algorithm is private if ***its output reveals no more*** *about an individual* ***than what was already known*** *about him/her before."*

- Not Correct!
- Impossible to reveal exactly nothing if the result is to depend at all on the data

What does it mean for an algorithm to be private?

"An algorithm is private if ***its output reveals no more*** *about an individual* ***than what was already known*** *about him/her before."*

- Not Correct!
- Impossible to reveal exactly nothing if the result is to depend at all on the data
- Before/after requirement depends on the adversary's side knowledge
  - No way to measure the information leakage when the adversary's side knowledge is unknown

What does it mean for an algorithm to be private?

“An algorithm is private if what can be learned about an individual in the dataset ***is not much more*** *than what would be learned* ***if the same algorithm is run without him/her*** *in the dataset.”*

What does it mean for an algorithm to be private?

"An algorithm is private if what can be learned about an individual in the dataset ***is not much more*** *than what would be learned* ***if the same algorithm is run without him/her*** *in the dataset."*

- Correct!
- Now the adversary **cannot infer the presence/absence of an individual** in the dataset
- Nothing **specific** can be learned about the individual
- To be robust against side knowledge, **the algorithm must be randomized**
  - Otherwise, the adversary can learn something about the individual by analyzing the difference between the output of the algorithm with and without the individual

Privacy Expectations

- Unreasonable Privacy Expectations:
  - **Privacy for free?** No, privatizing requires removing information ($\Rightarrow$accuracy loss)
  - **Absolute privacy?** No, your neighbour's habits are correlated with your habits
- Reasonable Privacy Expectations:
  - **Robust to side knowledge:** limit information leaked even in the presence of arbitrary side knowledge
  - **Quantitative:** one must be able to quantify the privacy cost
  - **Plausible deniability:** your presence in a database cannot be ascertained

Differential Privacy: Definition, Properties, and Mechanisms

Differential Privacy [Dwork et al., 2006]

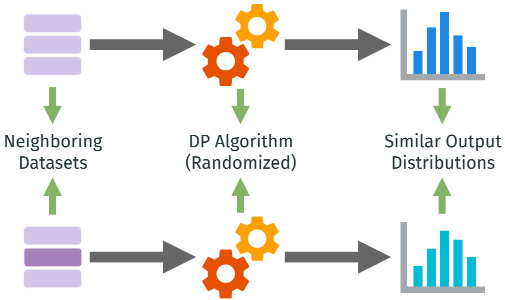
Let $\epsilon > 0$ and $\delta \in [0,1)$. A randomized algorithm $A : \mathcal{D} \rightarrow \mathcal{O}$ is (ϵ, δ)-differentially private if for all neighboring datasets $D \simeq D'$ and all sets of outputs $S \subseteq \mathcal{O}$:

$$\Pr\left[A(D) \in S\right] \leq e^{\epsilon} \Pr\left[A(D') \in S\right] + \delta$$

Differential Privacy [Dwork et al., 2006]

Let $\epsilon > 0$ and $\delta \in [0, 1)$. A randomized algorithm $A : \mathcal{D} \to \mathcal{O}$ is (ϵ, δ)-differentially private if for all neighboring datasets $D \simeq D'$ and all sets of outputs $S \subseteq \mathcal{O}$:

$$\Pr\left[A(D) \in S\right] \leq e^{\epsilon} \Pr\left[A(D') \in S\right] + \delta$$



Differential Privacy: Definition

Differential Privacy [Dwork et al., 2006]

Let $\epsilon > 0$ and $\delta \in [0,1)$. A randomized algorithm $A : \mathcal{D} \to \mathcal{O}$ is (ϵ, δ)-differentially private if for all neighboring datasets $D \simeq D'$ and all sets of outputs $S \subseteq \mathcal{O}$:

$$\Pr\left[A(D) \in S\right] \leq e^{\epsilon} \Pr\left[A(D') \in S\right] + \delta$$

- The probability bound captures how much protection we get
  - ϵ quantifies information leakage
    - Often called **privacy budget**
  - δ accounts for "bad events" that might result in high privacy losses
    - Algorithm $A(x_1, \ldots, x_n) = x_{Unif([n])}$ is $(0, 1/n)$-DP
    - Should be very small ($\delta << 1/n$)
    - if $\delta = 0$, then it is called **Pure DP**. Otherwise, it is called **Approximate DP**.

Differential Privacy: Definition

Differential Privacy [Dwork et al., 2006]

Let $\epsilon > 0$ and $\delta \in [0, 1)$. A randomized algorithm $A : \mathcal{D} \to \mathcal{O}$ is (ϵ, δ)-differentially private if for all neighboring datasets $D \simeq D'$ and all sets of outputs $S \subseteq \mathcal{O}$:

$$\Pr[A(D) \in S] \leq e^{\epsilon} \Pr\left[A(D') \in S\right] + \delta$$

- The neighboring relation captures **what is protected**
  - Definition depends on the application
  - Affects the privacy guarantee

Constructing Numeric DP Algorithms

- Suppose we want to compute a **numeric function** $f : \mathcal{D} \to \mathbb{R}^k$ of a private dataset D
- How to construct a DP algorithm (or **mechanism**) for computing $f(D)$?
  - How much randomness (error) do we add?
  - How to introduce this randomness in the output?

Global Sensitivity

Definition: Global ℓ_p sensitivity

The global ℓ_p sensitivity of a query (function) $f : \mathcal{D} \to \mathbb{R}^K$ is defined as:

$$\Delta_p(f) = \max_{D \simeq D'} \left\| f(D) - f(D') \right\|_p$$

- Indicates how much one record can affect the value of the function in the worst case
- Gives the amount of uncertainty needed to hide any single contribution
- Think about the ℓ_1 sensitivity of the following queries:
  - How many people have blond hair?
  - How many males, how many people with blond hair?
  - How many people have blond hair, how many people have dark hair, how many people have brown hair, how many people have red hair?
  - What is the average salary?

Laplace mechanism $\mathcal{A}_{\text{Lap}}(D, f : \mathcal{D} \to \mathbb{R}^K, \epsilon)$

1. Compute Δ= $\Delta_1(f)$
2. For $k = 1, \ldots, K$: draw $Y_k \sim \text{Lap}(\frac{\Delta}{\epsilon})$ independently for each k, where Lap(b) is the Laplace distribution with scale parameter b:

$$p(y; b) = \frac{1}{2b} \exp\left(-\frac{|y|}{b}\right)$$

3. Output $f(D) + Y$, where $Y = (Y_1, \ldots, Y_K) \in \mathbb{R}^K$

- **Theorem:** The Laplace mechanism $\mathcal{A}_{\text{Lap}}(D, f : \mathcal{D} \to \mathbb{R}^K, \epsilon)$ satisfies ϵ-DP

Proof.

- Consider any pair of neighboring datasets D, D' and any $\mathcal{S} \subseteq \mathbb{R}^K$
- Denoting by g and g' the p.d.f. of $\mathcal{A}_{\text{Lap}}(D, f, \varepsilon)$ and $\mathcal{A}_{\text{Lap}}(D', f, \varepsilon)$ respectively:

$$\frac{\Pr[\mathcal{A}_{\text{Lap}}(D) \in \mathcal{S}]}{\Pr[\mathcal{A}_{\text{Lap}}(D') \in \mathcal{S}]} = \frac{\int_{o \in \mathcal{S}} g(o)}{\int_{o \in \mathcal{S}} g'(o)} \leq \max_{o \in \mathcal{S}} \frac{g(o)}{g'(o)}$$

- Let p denote the p.d.f. of $\text{Lap}(\Delta/\varepsilon)$ and fix some $o = (o_1, \ldots, o_K) \in \mathcal{S}$. Then we have:

$$g(o) = \prod_{k=1}^{K} p(o_k - f_k(D)) \quad \text{and} \quad g'(o) = \prod_{k=1}^{K} p(o_k - f_k(D')),$$

where $f_k(\cdot)$ denotes the k-th entry of $f(\cdot)$

Proof.

- Plugging the definition of g and g', then using the triangle inequality, the definition of Δ, we get:

$$
\begin{aligned}
\frac{g(o)}{g'(o)} &= \prod_{k=1}^{K} \frac{p(o_k - f_k(D))}{p(o_k - f_k(D'))} = \prod_{k=1}^{K} \frac{\exp(-\frac{\varepsilon}{\Delta}|o_k - f_k(D)|)}{\exp(-\frac{\varepsilon}{\Delta}|o_k - f_k(D')|)} \\
&= \exp\left(\frac{\varepsilon}{\Delta}\sum_{k=1}^{K} |o_k - f_k(D')| - |o_k - f_k(D)|\right) \\
&\leq \exp\left(\frac{\varepsilon}{\Delta}\sum_{k=1}^{K} |f_k(D) - f_k(D')|\right) = \exp\left(\frac{\varepsilon}{\Delta}\|f(D) - f(D')\|_1\right) \leq \exp\left(\frac{\varepsilon}{\Delta}\Delta\right) = e^{\varepsilon}
\end{aligned}
$$

Gaussian mechanism $\mathcal{A}_{\text{Gauss}}(D, f : \mathcal{D} \to \mathbb{R}^K, \epsilon, \delta)$

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 $\sigma = \frac{\Delta}{\epsilon}\sqrt{2 \log(1/\delta)}$
3. Output $f(D) + Y$, where $Y = (Y_1, \ldots, Y_K) \in \mathbb{R}^K$

- **Theorem:** The Gaussian mechanism $\mathcal{A}_{\text{Gauss}}(D, f, \epsilon, \delta)$ satisfies (ϵ, δ)-DP
  - See [Dwork et al., 2014] for the proof

- Why to use the Gaussian mechanism?
  - **Same noise type** as other sources of noise
    - Better/simpler to analyze
    - Sum of Gaussian random variables is Gaussian
  - Adds **less noise** than the Laplace mechanism in higher dimensions
    - ℓ_2 sensitivity is much less than ℓ_1 sensitivity when dimensionality increases
  - Allows **tighter composition** results

- **Robustness to post-processing:** if f is (ϵ, δ)-DP and g is an arbitrary function, then $g(f(D))$ remains (ϵ, δ)-DP

- **Robustness to post-processing:** if f is (ϵ, δ)-DP and g is an arbitrary function, then $g(f(D))$ remains (ϵ, δ)-DP
- **Basic composition:** if $f_1, \ldots, f_k$ are (ϵ_i, δ_i)-DP, then $[f_1(D), \ldots, f_k(D)]$ is $(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i)$-DP

- **Robustness to post-processing:** if f is (ϵ, δ)-DP and g is an arbitrary function, then $g(f(D))$ remains (ϵ, δ)-DP
- **Basic composition:** if $f_1, \dots, f_k$ are (ϵ_i, δ_i)-DP, then $[f_1(D), \dots, f_k(D)]$ is $(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i)$-DP
- **Advanced composition:** if $f_1, \dots, f_k$ are all (ϵ, δ)-DP, then for any $\delta' > 0$, $[f_1(D), \dots, f_k(D)]$ is $(\epsilon', k\delta + \delta')$-DP with $\epsilon' = \epsilon\sqrt{2k \ln (1/\delta')} + k\epsilon(e^\epsilon - 1)$

Fundamental Properties of Differential Privacy

- **Robustness to post-processing:** if f is (ϵ, δ)-DP and g is an arbitrary function, then $g(f(D))$ remains (ϵ, δ)-DP
- **Basic composition:** if $f_1, \dots, f_k$ are (ϵ_i, δ_i)-DP, then $[f_1(D), \dots, f_k(D)]$ is $(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i)$-DP
- **Advanced composition:** if $f_1, \dots, f_k$ are all (ϵ, δ)-DP, then for any $\delta' > 0$, $[f_1(D), \dots, f_k(D)]$ is $(\epsilon', k\delta + \delta')$-DP with $\epsilon' = \epsilon\sqrt{2k \ln(1/\delta')} + k\epsilon(e^\epsilon - 1)$
- **Parallel composition:** if $f_1, \dots, f_k$ are (ϵ_i, δ_i)-DP, then $[f_1(D_1), \dots, f_k(D_k)]$ is $(\max_i \epsilon_i, \max_i \delta_i)$-DP if $D_1, \dots, D_k$ are distinct

Fundamental Properties of Differential Privacy

- **Robustness to post-processing:** if f is (ϵ, δ)-DP and g is an arbitrary function, then $g(f(D))$ remains (ϵ, δ)-DP
- **Basic composition:** if $f_1, \dots, f_k$ are (ϵ_i, δ_i)-DP, then $[f_1(D), \dots, f_k(D)]$ is $(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i)$-DP
- **Advanced composition:** if $f_1, \dots, f_k$ are all (ϵ, δ)-DP, then for any $\delta' > 0$, $[f_1(D), \dots, f_k(D)]$ is $(\epsilon', k\delta + \delta')$-DP with $\epsilon' = \epsilon\sqrt{2k \ln(1/\delta')} + k\epsilon(e^\epsilon - 1)$
- **Parallel composition:** if $f_1, \dots, f_k$ are (ϵ_i, δ_i)-DP, then $[f_1(D_1), \dots, f_k(D_k)]$ is $(\max_i \epsilon_i, \max_i \delta_i)$-DP if $D_1, \dots, D_k$ are distinct
- **Group privacy:** if f is (ϵ, δ)-DP w.r.t $D \simeq D'$ (i.e., a single change), then f is $(t\epsilon, te^{t\epsilon}\delta)$-DP w.r.t $D \simeq^t D'$ (i.e., t changes)

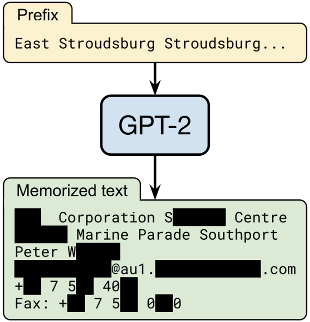
Fundamental Properties of Differential Privacy

- **Robustness to post-processing:** if f is (ϵ, δ)-DP and g is an arbitrary function, then $g(f(D))$ remains (ϵ, δ)-DP
- **Basic composition:** if $f_1, \ldots, f_k$ are (ϵ_i, δ_i)-DP, then $[f_1(D), \ldots, f_k(D)]$ is $(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i)$-DP
- **Advanced composition:** if $f_1, \ldots, f_k$ are all (ϵ, δ)-DP, then for any $\delta' > 0$, $[f_1(D), \ldots, f_k(D)]$ is $(\epsilon', k\delta + \delta')$-DP with $\epsilon' = \epsilon\sqrt{2k \ln(1/\delta')} + k\epsilon(e^\epsilon - 1)$
- **Parallel composition:** if $f_1, \ldots, f_k$ are (ϵ_i, δ_i)-DP, then $[f_1(D_1), \ldots, f_k(D_k)]$ is $(\max_i \epsilon_i, \max_i \delta_i)$-DP if $D_1, \ldots, D_k$ are distinct
- **Group privacy:** if f is (ϵ, δ)-DP w.r.t $D \simeq D'$ (i.e., a single change), then f is $(t\epsilon, te^{t\epsilon}\delta)$-DP w.r.t $D \simeq^t D'$ (i.e., t changes)
- **Robustness to side knowledge:** if for a data record $x \in D$ the attacker has prior P^x_{prior} and computes $P^x_{posterior}$ after observing $f(D)$ where f is (ϵ, δ)-DP, then $dist(P^x_{prior}, P^x_{posterior}) = O(\epsilon)$

Differentially Private Machine Learning

ML Models Are Not Safe

- ML models are elaborate kinds of aggregate statistics!
- They are susceptible to privacy attacks, e.g.,
  - **Membership inference attack:** infer whether a particular data record is in the training dataset [Shokri et al., 2017]
  - **Reconstruction attack:** reconstruct all or part of the training data [Carlini et al., 2021]



Setup: A curator has a dataset $D = [(x_1, y_1), \ldots, (x_n, y_n)]$ of n individuals and wants to train a model on D that minimizes the empirical risk over model parameters θ:

$$L(D, \theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, y_i, \theta) + \lambda R(\theta)$$

- **Examples:** logistic regression, SVM, linear regression, neural networks, etc.

Private ERM Algorithms

- **Output Perturbation** [Chaudhuri et al., 2011]**:** add noise Z to $\hat{\theta} = \arg\min_\theta L(D, \theta)$
  - Difficult to find the output sensitivity
  - Requires restrictive assumptions on the model (e.g., linear model, convexity)

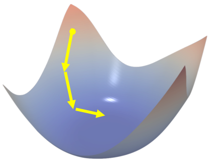
Private ERM Algorithms

- **Output Perturbation [Chaudhuri et al., 2011]:** add noise Z to $\hat{\theta} = \arg\min_\theta L(D, \theta)$
  - Difficult to find the output sensitivity
  - Requires restrictive assumptions on the model (e.g., linear model, convexity)
- **Objective Perturbation [Chaudhuri et al., 2011]:** add noise Z to $L(D, \theta)$ and then solve the perturbed optimization problem
  - Difficult to find the objective sensitivity
  - Requires restrictive assumptions on the model (e.g., linear model, convexity)

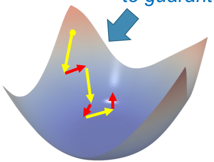
Private ERM Algorithms

- **Output Perturbation** [Chaudhuri et al., 2011]: add noise Z to $\hat{\theta} = \arg\min_\theta L(D, \theta)$
  - Difficult to find the output sensitivity
  - Requires restrictive assumptions on the model (e.g., linear model, convexity)
- **Objective Perturbation** [Chaudhuri et al., 2011]: add noise Z to $L(D, \theta)$ and then solve the perturbed optimization problem
  - Difficult to find the objective sensitivity
  - Requires restrictive assumptions on the model (e.g., linear model, convexity)
- **Gradient Perturbation** [Bassily et al., 2014, Abadi et al., 2016]: optimize $L(D, \theta)$ using mini-batch SGD with noisy gradients
  - Easy to bound the gradient sensitivity
  - Requires no assumptions on the model

Gradient Perturbation [Bassily et al., 2014, Abadi et al., 2016]: optimize $L(D, \theta)$ using mini-batch SGD with noisy gradients



Stochastic Gradient Descent



DP Stochastic Gradient Descent

SGD Algorithm

input : Data $\{\vec{x}_1 \ldots, \vec{x}_N\}$, learning rate η, batch size B, epochs E,

Initialize $\vec{\theta}_0$ randomly

for $t \in [E \cdot \frac{N}{B}]$ **do**

  Sample a batch $\vec{B}_t$ by selecting each $\vec{x}_i$ independently with probability $\frac{B}{N}$

  For each $\vec{x}_i \in \vec{B}_t$: $\vec{g}_t(\vec{x}_i) \leftarrow \nabla_{\vec{\theta}_t} L(\vec{x}_i, \vec{\theta}_t)$ // `compute per-sample gradients`

  $\tilde{\vec{g}}_t \leftarrow \frac{1}{B} \left(\sum_{\vec{x}_i \in \vec{B}_t} \tilde{\vec{g}}_t(\vec{x}_i) \right)$

  $\vec{\theta}_{t+1} \leftarrow \vec{\theta}_t - \eta \tilde{\vec{g}}_t$ // `SGD step`

end

output: $\vec{\theta}_{\frac{TN}{B}}$

Differentially Private SGD

DP-SGD Algorithm [Abadi et al., 2016]

input : Data $\{\vec{x}_1 \dots, \vec{x}_N\}$, learning rate η, batch size B, epochs E, clipping threshold C, noise variance σ^2,

Initialize $\vec{\theta}_0$ randomly

for $t \in [E \cdot \frac{N}{B}]$ **do**

  Sample a batch $\vec{B}_t$ by selecting each $\vec{x}_i$ independently with probability $\frac{B}{N}$

  For each $\vec{x}_i \in \vec{B}_t$: $\vec{g}_t(\vec{x}_i) \leftarrow \nabla_{\vec{\theta}_t} L(\vec{x}_i, \vec{\theta}_t)$ `// compute per-sample gradients`

  $\tilde{\vec{g}}_t(\vec{x}_i) \leftarrow \mathrm{clip}(\vec{g}_t(\vec{x}_i), C)$ `// clip gradients to max norm` C

  $\tilde{\vec{g}}_t \leftarrow \frac{1}{B}\left(\sum_{\vec{x}_i \in \vec{B}_t} \tilde{\vec{g}}_t(\vec{x}_i) + \mathcal{N}(0, \sigma^2 \vec{I})\right)$ `// add Gaussian noise with variance` σ^2

  $\vec{\theta}_{t+1} \leftarrow \vec{\theta}_t - \eta \tilde{\vec{g}}_t$ `// SGD step`

end

output: $\vec{\theta}_{\frac{TN}{B}}$

Privacy Analysis of DP-SGD

- At each step the gradient is (ϵ, δ)-DP w.r.t. the group (batch)
- What is the DP guarantee of the whole dataset?

one step, within the group

ϵ, δ

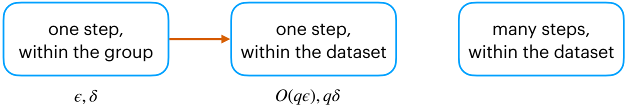
one step, within the dataset

many steps, within the dataset

Privacy amplification by subsampling [Balle et al., 2018]

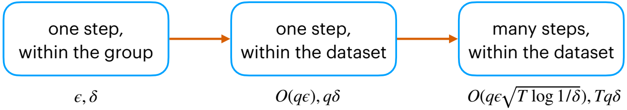
Let A be an (ϵ, δ)-DP algorithm and $S : \mathcal{X}^n \to \mathcal{X}^m$ be a subsampling procedure returning m out of n samples uniformly at random without replacement. Let $q = m/n$ be the sampling probability. Then $A \circ S$ is $(\epsilon', q\delta)$-DP with $\epsilon' = \ln\left(1 + q(e^{\epsilon} - 1)\right)$.

- At each iteration of DP-SGD, each data point $x \in D$ is sampled with probability $q = B/N$
- Based on the privacy amplification theorem, the privacy guarantee of DP-SGD at each iteration is $(O(q\epsilon), q\delta)$-DP
- What is the privacy guarantee of DP-SGD over all the iterations?



Privacy Analysis of DP-SGD

- With $T = E \cdot N/B$ iterations, the DP-SGD algorithm is a composition of T smaller $(O(q\epsilon), q\delta)$-DP algorithms
- Based on the advanced composition, the total privacy guarantee of DP-SGD is $(O(q\epsilon\sqrt{T \log 1/\delta}), qT\delta)$-DP
- However, the advanced composition is not tight
  - Can we do better?



Rényi Differential Privacy

Why Another Privacy Definition?

- The results of advanced composition are not quite tight: they give somewhat loose upper bounds on the privacy cost
- Rényi DP is a generalization of standard (ϵ, δ)-DP that provides a tighter privacy bound
    - In particular, it provides **tighter composition results** for the **Gaussian mechanism**
- One can perform the privacy analysis using Rényi DP (composition, subsampling, etc) and then convert back to (ϵ, δ)-DP at the end
    - This shaves off a logarithmic factor in δ and gives better constants
- Rényi DP has all the good properties of (ϵ, δ)-DP (e.g., robustness to post-processing, composability, robustness to side knowledge, etc), plus some more

Rényi Differential Privacy [Mironov, 2017]

Let $\alpha > 1, \epsilon > 0$. A randomized algorithm $\mathcal{A}$ is (α, ϵ)-RDP if for every neighboring datasets $X \simeq X'$, we have:

$$D_\alpha \left(\mathcal{A}(X) \| \mathcal{A}(X') \right) \leq \epsilon$$

where $D_\alpha(P\|Q)$ is the Rényi divergence of order α between probability distributions P and Q defined as:

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q} \left[\frac{P(x)}{Q(x)} \right]^\alpha .$$

Gaussian Mechanism under Rényi DP

Gaussian mechanism $\mathcal{A}_{\text{Gauss}}(D, f : \mathcal{D} \to \mathbb{R}^K)$

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
3. Output $f(D) + Y$, where $Y = (Y_1, \ldots, Y_K) \in \mathbb{R}^K$

- **Theorem:** For any $\alpha > 1$, the Gaussian mechanism $\mathcal{A}_{\text{Gauss}}(D, f)$ satisfies (α, ϵ)-RDP with $\epsilon = \alpha \frac{\Delta_2(f)}{2\sigma^2}$.
    - See [Mironov, 2017] for the proof

Properties of Rényi DP

- **Sequential composition [Mironov, 2017]:** if $f_1, \ldots, f_k$ are (α, ϵ_i)-RDP, then $[f_1(D), \ldots, f_k(D)]$ is $(\alpha, \sum_{i=1}^{k} \epsilon_i)$-RDP

Properties of Rényi DP

- **Sequential composition [Mironov, 2017]:** if $f_1, \ldots, f_k$ are (α, ϵ_i)-RDP, then $[f_1(D), \ldots, f_k(D)]$ is $(\alpha, \sum_{i=1}^{k} \epsilon_i)$-RDP
- **Privacy amplification by subsampling [Wang et al., 2019]:** if A is (α, ϵ)-RDP and S is a subsampling procedure with sampling probability q, then $A \circ S$ is (α, ϵ')-RDP with:

$$\epsilon' \leq \frac{1}{\alpha - 1} \log \left(1 + q^2 \binom{\alpha}{2} \min \left\{ 4(e^{\epsilon(2)} - 1), e^{\epsilon(2)} \min\{2, (e^{\epsilon(\infty)} - 1)^2\} \right\} \right.$$
$$\left. + \sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} e^{(j-1)\epsilon(j)} \min\{2, (e^{\epsilon(\infty)} - 1)^j\} \right)$$

- **Sequential composition [Mironov, 2017]:** if $f_1, \dots, f_k$ are (α, ϵ_i)-RDP, then $[f_1(D), \dots, f_k(D)]$ is $(\alpha, \sum_{i=1}^k \epsilon_i)$-RDP
- **Privacy amplification by subsampling [Wang et al., 2019]:** if A is (α, ϵ)-RDP and S is a subsampling procedure with sampling probability q, then $A \circ S$ is (α, ϵ')-RDP with:

$$\epsilon' \leq \frac{1}{\alpha - 1} \log \Bigg(1 + q^2 \binom{\alpha}{2} \min\left\{4(e^{\epsilon(2)} - 1), e^{\epsilon(2)} \min\{2, (e^{\epsilon(\infty)} - 1)^2\}\right\} + \sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} e^{(j-1)\epsilon(j)} \min\{2, (e^{\epsilon(\infty)} - 1)^j\}\Bigg)$$

- **Conversion to (ϵ, δ)-DP [Mironov, 2017]:** If $\mathcal{A}$ is an (α, ϵ)-RDP algorithm, then for any $\delta \in (0, 1)$ it satisfies (ϵ', δ)-DP with $\epsilon' = \epsilon + {}^{\log(1/\delta)}/_{\alpha - 1}$

Hands-on Tutorial

Thank You!

Questions?

sajadmanesh@idiap.ch

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016).
Deep learning with differential privacy.
In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Balle, B., Barthe, G., and Gaboardi, M. (2018).
Privacy amplification by subsampling: Tight analyses via couplings and divergences.
Advances in Neural Information Processing Systems, 31.

References II

Bassily, R., Smith, A., and Thakurta, A. (2014).
Private empirical risk minimization: Efficient algorithms and tight error bounds.
In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. (2021).
Extracting training data from large language models.
In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. (2011).
Differentially private empirical risk minimization.
Journal of Machine Learning Research, 12(3).

- De Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., and Blondel, V. D. (2013).
  Unique in the crowd: The privacy bounds of human mobility.
  Scientific reports, 3(1):1–5.
- Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006).
  Calibrating noise to sensitivity in private data analysis.
  In *Theory of cryptography conference*, pages 265–284. Springer.
- Dwork, C., Roth, A., et al. (2014).
  The algorithmic foundations of differential privacy.
  Foundations and Trends® in Theoretical Computer Science, 9(3–4):211–407.
- Mironov, I. (2017).
  Rényi differential privacy.
  In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE.

References IV

Narayanan, A. and Shmatikov, V. (2008).
Robust de-anonymization of large sparse datasets.
In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017).
Membership inference attacks against machine learning models.
In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

Wang, Y.-X., Balle, B., and Kasiviswanathan, S. P. (2019).
Subsampled rényi differential privacy and analytical moments accountant.
In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR.