Differential Privacy

Sinan Yıldırım

Sabancı Üniversitesi

Trustworthy AI

Galatasaray Üniversitesi, İstanbul

30 Mart 2024

Motivation

Differential privacy: Definition and examples

Basic Properties

Application: Differentially private stochastic gradient descend

Data analysis vs Privacy

Sensitive data set of n individuals: $x_1, \ldots, x_n$

Two conflicting interests:

1. We want to work with sensitive data sets
   - to perform inference about a population.
   - for optimization
   - etc.

2. Individuals contributing to data sets with their sensitive information want to preserve their privacy.

A significant amount of research is devoted to developing useful methods for data analysis while protecting data privacy.

An outline

This lecture:

- Introduction to main concepts and tools of differential privacy
- A step-by-step application from data-driven optimization.

Tutorial:

- Python implementation of some differentially private algorithms.

Privacy framework

Individual i with *sensitive* information $x_i \in \mathsf{X}$.

Data collected from n individuals: $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathsf{X}^n$.

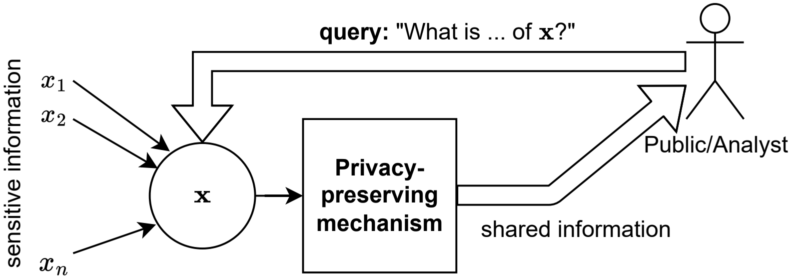
(Statistics of) the $\boldsymbol{x}$ is to be shared with the public for analysis.

Data privacy: main question

How should (statistics of) $\boldsymbol{x} = (x_1, \ldots, x_n)$ be shared so that

- privacy of each individual is protected, and
- the shared information is useful.

A graphical summary



Some extreme solutions(?)

- **Full transparency:** Share $\boldsymbol{x} = (x_1, \ldots, x_n)$.
  - Very useful, but not private.
- **Full secrecy:** Toss a coin and share the outcome.
  - Very private, but not useful.

More sensible alternatives

- **Anonymization:** Remove any identifying information from the data.

- **Statistic of private data:** Do not share $\boldsymbol{x} = (x_1, \ldots, x_n)$; share a statistic.

$$S(x_{1:n}) = \frac{1}{n} \sum_{i=1}^{n} x_i,$$

All against one

Both methods are prone to *conspiracy by all against one*.

- Imagine individuals $1, 2, \ldots, n-1$ have shared their data $x_1, \ldots, x_{n-1}$ among themselves.
- $\Rightarrow x_n$ can be found!

$$S(x_{1:n}) = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \Rightarrow x_n = nS(x_{1:n}) - \sum_{i=1}^{n-1} x_i$$

Deterministic outputs do not work!

Randomized algorithms

Set of data values (sample space): X

A data set: $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathsf{X}^n$

Set of data sets: $\mathcal{X} = \bigcup_{n=1}^{\infty} \mathsf{X}^n$.

Randomised algorithm

A randomized algorithm is essentially a *random* function
$A : \mathcal{X} \mapsto \mathcal{Y}$.
The output of the algorithm upon taking an input $\boldsymbol{x} \in \mathcal{X}$,

$$A(\boldsymbol{x}) \in \mathcal{Y},$$

is a *random variable* with support domain $\mathcal{Y}$.

The randomness is due to the inner mechanisms of the algorithm.

Neighboring data sets

$\boldsymbol{x} = (x_1, \ldots, x_n)$: sensitive data of n individuals.

Neighbouring data sets (replacement)

Datasets $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ are neighbours if they differ by a single element

$$\boldsymbol{x} = (x_1, \ldots, x_k, \ldots, x_n), \quad \boldsymbol{x}' = (x_1, \ldots, x_k', \ldots, x_n)$$

We want to have a mechanism whose output on $\boldsymbol{x}$ and $\boldsymbol{x}'$ are (probabilistically) similar when $\boldsymbol{x}$ and $\boldsymbol{x}'$ are neighbors.

Differential privacy

Differential privacy (Dwork, 2006)

We say that A is (ϵ, δ)-DP if, for neighbour $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ and any subset of output values $O \subseteq \mathcal{Y}$,

$$\mathbb{P}\left[A(\boldsymbol{x}) \in O\right] \leq e^{\epsilon} \mathbb{P}\left[A(\boldsymbol{x}') \in O\right] + \delta.$$

When $\delta = 0$, we say A is ϵ-DP (pure differential privacy).

Related forms of privacy:

- Reyni differential privacy
- (zero) concentrated differential privacy
- Gaussian differential privacy (GDP)
- Bayesian differential privacy
- etc.

Alternative neighboring relations

Previously, we the neighbor relation replacement. Other relations are possible:

Neighbouring data sets (addition/removal)

Datasets $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ are neighbours if one can be obtained from the other by addition or removal of a single element. Examples:

$$\boldsymbol{x} = (x_1, \dots, x_k, \dots, x_n), \quad \boldsymbol{x}' = (x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n)$$
$$\boldsymbol{x} = (x_1, \dots, x_k, \dots, x_n), \quad \boldsymbol{x}' = (x_1, \dots, x_k, x', x_{k+1}, \dots, x_n).$$

Privacy properties can depend on the neighboring relation.

- (ϵ, δ)-DP wrt replacement $\Rightarrow$ (ϵ, δ)-DP wrt to add/rem.
- (ϵ, δ)-DP wrt add/rem $\Rightarrow$ $(2\epsilon, (1 + e^{\epsilon})\delta)$-DP wrt replacement.

Laplace mechanism

The L_1-**sensitivity** of a function $S : \mathcal{X} \mapsto \mathbb{R}^d$ is given by

$$\Delta_{S,1} = \sup_{\text{neighbour } \boldsymbol{x},\boldsymbol{x}'} \|S(\boldsymbol{x}) - S(\boldsymbol{x}')\|_1.$$

Laplace mechanism

An algorithm is ϵ-DP if it outputs

$$A(\boldsymbol{x}) = S(\boldsymbol{x}) + V, \quad V_i \overset{\text{i.i.d.}}{\sim} \text{Laplace}\left(\frac{\Delta_{S,1}}{\epsilon}\right), \quad i = 1, \ldots, d.$$

All against one - revisited

Now, instead of sharing $S(x_{1:n}) = \frac{1}{n}\sum_{i=1}^{n} x_i$, we share

$$Y = \frac{1}{n}\sum_{i=1}^{n} x_i + V.$$

- Even if individuals $1, 2, \ldots, n-1$ have shared their data $x_1, \ldots, x_{n-1}$ among themselves, x_n cannot be deduced!

  $$Y = \frac{1}{n}\sum_{i=1}^{n} x_i + V \quad \Rightarrow x_n = nY - \sum_{i=1}^{n-1} x_i - nV$$

  Randomness protects x_n.

Randomized responses

Randomization of binary responses.

Question: Do you approve the president?

ϵ-DP randomization

Answer truly with probability $\frac{e^\epsilon}{1+e^\epsilon}$; otherwise flip your answer.

Can be extended to $K \geq 2$ categories.

Question: Among K political parties, which one do you support?

ϵ-DP randomization

Answer truly with probability $\frac{e^\epsilon}{K-1+e^\epsilon}$; otherwise answer at random.

Randomized responses provide DP at the local level.

Such a DP guarantee is called Local DP.

Post-processing

One of the useful properties of DP is **post-processing**.

Post-processing

If A is (ϵ, δ), then $f \circ A$ is (ϵ, δ)-DP, too.

Note: $f \circ A(\boldsymbol{x}) = f(A(\boldsymbol{x}))$.

Meaning: Differential privacy is preserved under post-processing.

Composition

Repeated application of DP algorithms on the same dataset degrade privacy.

K-fold composition

Assume A_k is (ϵ_k, δ_k)-DP for $k = 1, \ldots, K$. Application of A_k, $k = 1, \ldots, K$ on the same input data set results in

$$\left(\sum_{k=1}^{K} \epsilon_k, \sum_{k=1}^{K} \delta_k \right) \text{-DP.}$$

This result still holds when an algorithm depends on the outputs of the previous algorithms.

- particularly useful for adaptive/iterative algorithms.

When δ_k's are 0, the result is tight. With non-zero δ_k's, other definitions of DP compose better.

Reyni DP and zero-concentrated DP (zCDP)

Renyi divergence

For probability distributions P and Q the Renyi divergence of order $\alpha > 1$

$$D_\alpha(P||Q) := \frac{1}{\alpha - 1} \ln \mathbb{E}\left[P(x)/Q(x)\right]^\alpha$$

If $X \sim P$ and $Y \sim Q$, we $D_\alpha(X||Y)$ is equivalent to $D_\alpha(P||Q)$.

Reyni DP (Mironov, 2017) and zCDP (Bun and Steinke, 2016)

An algorithm A is (α, ε)-Reyni DP if for all neighbour $x, x' \in \mathrm{X}$,

$$D_\alpha(A(x)||A(x')) \leq \varepsilon.$$

An algorithm A is is ρ-zCDP if for all neighbor x, x' and $\alpha > 1$,

$$D_\alpha(A(x)||A(x')) \leq \alpha\rho$$

Composition properties for Reyni DP and zCDP

Composition theorem for Reyni DP

The composition of (α, ε_i)-Reyni-DP algorithms for $i = 1, \ldots, T$ is

$$\left(\alpha, \sum_{i=1}^{T} \varepsilon_i(\alpha)\right)\text{-Reyni DP.}$$

Composition theorem for zCDP

The composition of ρ_i-zCDP algorithms for $i = 1, \ldots, T$ is

$$\left(\sum_{i=1}^{T} \rho_i\right)\text{-zCDP.}$$

Gaussian mechanism

The L_2-**sensitivity** of a function $S : \mathcal{X} \mapsto \mathbb{R}^d$ is given by

$$\Delta_{S,2} = \sup_{\text{neighbour } \boldsymbol{x},\boldsymbol{x}'} \|S(\boldsymbol{x}) - S(\boldsymbol{x}')\|_2.$$

Gaussian mechanism

An algorithm is ρ-zCDP if it outputs

$$Y = S(\boldsymbol{x}) + V, \quad V_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \frac{\Delta_{S,2}^2}{\rho}\right), \quad i = 1, \ldots, d.$$

Conversions

To be able to convert one DP definition to another offers huge flexibility in designing algorithms.

zCDP to Reyni DP

If an algorithm ρ-zCDP, it is $(\alpha, \alpha\rho)$-Reyni DP for any α.

Reyni DP to (ϵ, δ)-DP

If an algorithm (α, ε)-Reyni DP, it is $(\epsilon, e^{-(\alpha-1)(\epsilon-\varepsilon)})$-DP for any $\epsilon > \varepsilon$.

zCDP to (ϵ, δ)-DP

If an algorithm A is ρ-zCDP, then it is (ϵ, δ) for all (ϵ, δ) satisfying

$$\delta > 0, \quad \epsilon = \rho + 4\rho \ln(1/\delta).$$

More conversions exist.

Privacy amplification by subsampling

Let A be private algorithm that operates on datasets

$$\boldsymbol{x} = (x_1, \ldots, x_n)$$

Consider another algorithm A', who

- Takes a random subsample from $\boldsymbol{x}$
- Operates on the subset just like A.

Question: What is the privacy of A'?

The answer depends on

- Type of privacy of A,
- Type of subsampling
- Neighborhood relation

See Balle et al. (2018); Steinke (2022) for more relations.

Amplification of (ϵ, δ)-DP

Assume A is (ϵ, δ)-DP.

Suppose that

- the subsample size is fixed to m and
- the subsample is drawn by sampling without replacement.
- the neighborhood relation is replacement.

Then, A' is (ϵ', δ')-DP, where

$$\epsilon' = \ln\left(1 + \frac{m}{n}(e^{\epsilon} - 1)\right), \quad \delta' = \frac{m}{n}\delta.$$

Amplification of Reyni DP

Assume A is $(\alpha, \varepsilon(\alpha))$-Reyni DP.
Meaning: A satisfies $(\alpha, \varepsilon(\alpha))$-Reyni DP for all $\alpha > 1$

Suppose that

- each element in $\boldsymbol{x}$ is included in the subsample with γ probability, independently of the other elements (Poisson subsampling).
- the neighborhood relation is addition/removal.

Then, A' is

$$(\alpha, \varepsilon_\gamma(\lceil \alpha \rceil))\text{-Reyni DP},$$

where

$$\varepsilon_\gamma(k) = \frac{1}{k-1} \ln \left((1-\gamma)^{k-1}(1+(k-1)\gamma) + \sum_{i=2}^{k} \binom{k}{i} (1-\gamma)^{k-i} \gamma^i e^{(i-1)\varepsilon(i)} \right)$$

Application: Differentially private stochastic gradient descend

Differentially private optimization with stochastic gradients

A data-driven optimization problem:

$$\min_{\theta \in \Theta} F(\theta; x_{1:n})$$

where

$$F(\theta; x_{1:n}) := \frac{1}{n} \sum_{i=1}^{n} f(\theta; x_i) + \frac{\lambda}{2} \|\theta\|^2$$

In a data-related framework,

- y_i: data from individual i,
- θ: model parameter,
- n: the data size.
- λ: regularizer (prior?)

Stochastic gradient and Nesterov's accelerated gradient

The gradient vector:

$$\nabla F(\theta; x_{1:n}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\theta; x_i).$$

Gradient descend:

$$\theta_{t+1} = \theta_t - \alpha \nabla F(\theta_t; x_{1:n}), \quad t \geq 0$$

Stochastic Gradient descend:

$$\theta_{t+1} = \theta_t - \alpha \frac{1}{m_t} \sum_{i \in U_t} \nabla f_i(\theta; x_i), \quad t \geq 0,$$

where $U_t \subset \{1, \ldots, n\}$ is a random subsample of size $m_t \leq n$.

Differentially private SGD

To achieve (ϵ, δ)-DP after T iterations

DP-SGD

$$\theta_{t+1} = \theta_t - \eta \left(\frac{1}{m_t} \sum_{i \in U_t} \nabla f_i(\theta_t; x_i) + v_t \right)$$

The distribution of the DP noise v_t depends on

- DP parameters: ϵ, δ.
- Sensitivity of $\nabla f_i(\theta_t; \cdot)$
- m_t (privacy amplification by subsampling)
- T (composition property)

Example: Logistic regression

Let $x = (z, y)$, where

- $z \in \mathbb{R}^d$ is the feature vector
- $y \in \{0, 1\}$: binary response.

The probability of observing a label "1" given the feature vector z and regression parameter $\theta \in \mathbb{R}^d$ is

$$p(y|z,\theta) = \frac{e^{yz\theta}}{1 + e^{z\theta}},$$

Let

$$f(\theta; x) = -\ln p(y|z,\theta)$$

Estimate θ by minimizing

$$F(\theta; x_{1:n}) := \frac{1}{n}\sum_{i=1}^{n} f(\theta; x_i) + \lambda\|\theta\|$$

Logistic regression - sensitivity

L_p sensitivity of $\nabla f(\theta, \cdot)$:

$$\Delta_p(\theta) = \sup_{x,x'} \|\nabla f(\theta; x) - \nabla f(\theta; x')\|_p = 2 \sup_x \|x\|_p$$

With unbounded data, the sensitivity is ∞.

Solutions:

- If the data is bounded $\|x\|_p \leq B_p/2$ for some $B_p < \infty$, then

$$\Delta_p(\theta) = B_p$$

- Clipping: Use a clipped version of $\nabla f(\theta; x')$

$$\nabla \widehat{f(\theta; x)} = \min\{B_p, \|\nabla f(\theta; x)\|_p\} \frac{\nabla f(\theta; x)}{\|\nabla f(\theta, x)\|_p}.$$

  The sensitivity of the clipped gradient is B_p.

Scenario 1

We want ϵ-DP after T iterations, using subsampling without replacement with fixed subsample size $m < n$.

- By the composition theorem for DP, we need to achieve ϵ/T-DP per iteration.
- Laplace noise is needed to achieve pure DP.

$$v_t \sim \text{Laplace}(\sigma)$$

By amplification due to subsampling, the privacy loss per iteration is

$$\left[(e^{B_1/\sigma m} - 1)\frac{m}{n} + 1\right]$$

Equate this to $e^{\epsilon/T}$, and solve for σ:

$$\sigma = \frac{B_1}{m \ln\left[1 + (e^{\epsilon/T} - 1)n/m\right]}$$

Scenario 2

We want (ϵ, δ)-DP after T iterations, without subsampling ($m = n$).

- Find ρ-zCDP that implies (ϵ, δ)-DP.

$$\epsilon = \rho + 2\sqrt{\rho \ln(1/\delta)} \Rightarrow \rho = \sqrt{\ln(1/\delta) + \epsilon} - \sqrt{\ln(1/\delta)}$$

- By the basic composition theorem for zCDP, we need to achieve ρ/T-zCDP per iteration.
- Gaussian noise is needed for zCDP.

$$v_t \sim \mathcal{N}(0, \sigma^2) \text{ provides } \frac{B_2^2}{n^2\sigma^2}$$

  Since the zCDP privacy loss per iteration is ρ/T, we solve

$$\frac{\rho}{T} = \frac{B_2^2}{n^2\sigma^2}$$

  for σ^2 to find

$$\sigma^2 = \frac{TB_2^2}{n^2\rho^2}$$

Scenario 3

We want (ϵ, δ)-DP after T iterations, *with* subsampling $(m < n)$.

DP-SGD

For $t = 1, \dots, T$,

$$\theta_{t+1} = \theta_t - \eta \left(\frac{1}{m} \sum_{i \in U_t} \nabla f_i(\theta_t; x_i) + v_t \right), \quad v_t \sim \mathcal{N}(0, \sigma^2 I)$$

Caution: This time, differently than the other two scenarios, we will assume that the neighboring relation is addition/removal.

Scenario 3: Algorithmic outline

An analytical formula for σ that gives (ϵ, δ)-DP after T iterations using σ is difficult to obtain.

This time, we will take the following approach:

- For a fixed noise level is σ and T iterations,
  1. Calculate the zCDP of the algorithm for one iteration if full data is used.
  2. Convert zCDP to Reyni-DP (because the latter behaves well under subsampling)
  3. Find the privacy amplification of a Reyni-DP algorithm in terms of Reyni-DP.
  4. Apply composition and find the overall Reyni-DP after T iterations.
  5. Convert Reyni-DP to (ϵ, δ)-DP
- The resulting DP parameters depend on σ, so lets denote them by $(\epsilon(\sigma), \delta(\sigma))$. We will arrange σ such that

  $$\epsilon(\sigma) \leq \epsilon, \delta(\sigma) \leq \delta$$

  and the differences are as small as possible.

Step 1: Find zCDP of a single iteration w.o subsampling

DP-SGD

For $t = 1, \ldots, T$,

$$\theta_{t+1} = \theta_t - \eta \left(\frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\theta_t; x_i) + v_t \right), \quad v_t \sim \mathcal{N}(0, \sigma^2 I)$$

If iterations were performed on the full data set, we would have

$$\frac{B_2^2}{n^2 \sigma^2}\text{-zCDP}$$

per iteration.

Step 2: Convert to zCDP to Reyni DP

DP-SGD

For $t = 1, \ldots, T$,

$$\theta_{t+1} = \theta_t - \eta \left(\frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\theta_t; x_i) + v_t \right), \quad v_t \sim \mathcal{N}(0, \sigma^2 I)$$

zCDP to Reyni DP

If an algorithm ρ-zCDP, it is $(\alpha, \alpha\rho)$-Reyni DP for any α.

Using the theorem

$$\frac{B_2^2}{n^2\sigma^2}\text{-zCDP} \quad \Rightarrow \quad \left(\alpha, \varepsilon(\alpha) := \alpha \frac{B_2^2}{n^2\sigma^2} \right) \text{- Reyni DP.}$$

Step 3: Privacy amplification with subsampling

DP-SGD

For $t = 1, \dots, T$,

$$\theta_{t+1} = \theta_t - \eta \left(\frac{1}{m_t} \sum_{i \in U_t} \nabla f_i(\theta_t; x_i) + v_t \right), \quad v_t \sim \mathcal{N}(0, \sigma^2 I)$$

Under Poisson subsampling, the privacy per iteration is amplified:

$(\alpha, \varepsilon(\alpha))$-Reyni DP + Poiss subs. with $\gamma \Rightarrow (\alpha, \varepsilon_\gamma(\lceil \alpha \rceil))$-Reyni DP

where, for a subsampling rate of $\gamma \in [0, 1]$, we have

$$\varepsilon_\gamma(k) = \frac{1}{k-1} \ln \left((1-\gamma)^{k-1}(1 + (k-1)\gamma) + \sum_{i=2}^{k} \binom{k}{i} (1-\gamma)^{k-i} \gamma^i e^{(i-1)\varepsilon(i)} \right)$$

Step 4: Privacy after T steps

Composition theorem for Reyni DP

The composition of $(\alpha, \varepsilon_i(\alpha))$-Reyni-DP algorithms for $i = 1, \ldots, T$ is

$$\left(\alpha, \sum_{i=1}^{T} \varepsilon_i(\alpha)\right) \text{-Reyni DP.}$$

After T steps, the algorithm becomes

$$\left(\alpha,\, T\epsilon_\gamma\left(\left\lceil \alpha \frac{B_2^2}{n^2\sigma^2} \right\rceil\right)\right) \text{-Reyni DP}$$

Step 5: Convert to (ϵ, δ)-DP

Reyni DP to (ϵ, δ)-DP

If an algorithm (α, ε)-Reyni DP, it is $(\epsilon, e^{-(\alpha-1)(\epsilon-\varepsilon)})$-DP for any $\epsilon > \varepsilon$.

Therefore, the algorithm after T iterations is

$$\left(\epsilon, \exp\left\{-(\alpha-1)\left[\epsilon - T\epsilon_\gamma\left(\left\lceil \alpha\frac{B_2^2}{n^2\sigma^2}\right\rceil\right)\right]\right\}\right)$$

for any

$$\epsilon > T\epsilon_\gamma\left(\left\lceil \alpha\frac{B_2^2}{n^2\sigma^2}\right\rceil\right)$$

Play with σ and α to achieve a targeted (ϵ, δ) privacy.

Balle, B., Barthe, G., and Gaboardi, M. (2018). Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 6280–6290, Red Hook, NY, USA. Curran Associates Inc.

Bun, M. and Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Proceedings, Part I, of the 14th International Conference on Theory of Cryptography - Volume 9985*, pages 635–658, New York, NY, USA. Springer-Verlag New York, Inc.

Dwork, C. (2006). Differential privacy. In Bugliesi, M., Preneel, B., Sassone, V., and Wegener, I., editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg. Springer Berlin Heidelberg.

Mironov, I. (2017). Rényi differential privacy. *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275.

Steinke, T. (2022). Composition of differential privacy & privacy amplification by subsampling.