

Data privacy protection and Differential Privacy

104369: Privacitat de Dades i Seguretat

G. Navarro-Arribas

UAB – DEIC

v. 20251002

1. Differential privacy mechanisms introduction
2. Sensitivity
3. The Laplace mechanism
4. Properties of differential privacy
5. Approximate Differential Privacy and the Gaussian Mechanism
6. Randomized Response (RR)

Differential privacy mechanisms

introduction

Recall: Differential Privacy Definition

Differential Privacy

A function K_q for a query q gives ϵ -differential privacy if for all datasets D_1 and D_2 differing in at most one element, and all $S \subseteq \text{Range}(K_q)$,

$$\frac{\Pr[K_q(D_1) \in S]}{\Pr[K_q(D_2) \in S]} \leq e^\epsilon \quad (1)$$

Also expressed as:

$$\Pr[K_q(D_1) \in S] \leq e^\epsilon \Pr[K_q(D_2) \in S]$$

Differential privacy mechanisms

- Protection methods for differential privacy are commonly named **mechanisms**.
- For numerical data: mostly based in additive noise.
 - $K_q(D) = q(D) + \text{noise}$
- Amount of noise determines the ε obtained,
- which depends on the actual function or query!

Sensitivity

Sensitivity of a function

Given a database space $\mathcal{D}$, and a function $f : \mathcal{D} \rightarrow \mathbb{R}$:

Global sensitivity

The global sensitivity of f with respect to norm $\| \cdot \|$ is:

$$GS(f) = \max_{D_1, D_2 \in \mathcal{D}, d(D_1, D_2)=1} \|f(D_1) - f(D_2)\|$$

- “How much the result can change by querying neighbor datasets?”.

→ $GS(f)$ is also denoted as Δ_f or $\Delta(f)$.

A note on sensitivity and norms

The norm determines how the difference is computed, and depends on the distribution of noise used in the mechanism:

- ℓ_1 norm: $\|(x_1, \dots, x_n)\|_1 = \sum_{i=1}^n |x_i|$ ($\rightarrow$ Laplace)
- ℓ_2 norm: $\|(x_1, \dots, x_n)\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$ ($\rightarrow$ Gaussian)

Thus, we can have:

- ℓ_1 -sensitivity, GS_1 , for Laplace noise distributions.
- ℓ_2 -sensitivity, GS_2 , for Gaussian noise distributions.

(Unless otherwise specified we will use the ℓ_1 norm in the following examples).

Global sensitivity example i

<i>Name</i>	<i>Schizo.</i>
Caracalla	1
Nerva	0
Commodus	1
Trajan	0
Hadrian	0

(a) D_1 .

<i>Name</i>	<i>Schizo.</i>
Nerva	0
Commodus	1
Trajan	0
Hadrian	0

(c) D_3 .

<i>Name</i>	<i>Schizo.</i>
Tiberius	1
Caracalla	1
Nerva	0
Commodus	1
Trajan	0
Hadrian	0

(b) D_2 .

<i>Name</i>	<i>Schizo.</i>
Vespasian	0
Caracalla	1
Nerva	0
Commodus	1
Trajan	0
Hadrian	0

(d) D_4 .

- $d(D_1, D_2) = d(D_1, D_3) = d(D_1, D_4) = 1$,
- $d(D_3, D_4) = d(D_2, D_4) = d(D_3, D_2) = 2$

Global sensitivity example ii

(Considering the use of the ℓ_1 norm:)

- $f_{count} \rightarrow$ “number of records with *schizophrenia*” .
 - $f_{count}(D_1) = 2, f_{count}(D_2) = 3, f_{count}(D_3) = 1, f_{count}(D_4) = 2.$
 $\Rightarrow GS(f_{count}) = 1!$
- **Counting** queries: $GS_1(f_{count}) = 1$
- But what about summation, averages, ... ?
 - Consider bounded vs. unbounded attributes!

Global Sensitivity on bounded data

→ The data (attribute, variables) V have a natural upper ($\max V$) and lower bound ($\min V$).

↪ E.g. grade of the subject: $[0, 10]$, age: $[0, 125]$, ...

- Summation: $GS_1(f_{sum}) = \max V$
- Average: $f_{avg} = \frac{f_{sum}}{f_{count}}$

→ **Important** We cannot look at the actual data and get its current maximum and lower value to bound it. That's cheating! (it can reveal information from the actual data)

- Try to transform the query to a bounded one.
- **Clipping**
 - Determine a max and min value and clip (truncate) the data to these values.
 - Need to fix them without looking at the data.
 - Otherwise: trial and error. Test, the result of the query with different clipping values until the result is stable. The final epsilon needs to add the composition of the number of tests done.

The Laplace mechanism

The **Laplace mechanism** for a function f is:

$$M_L(X, f, \varepsilon) = f(X) + (N_1, \dots, N_M)$$

where N_i are independent random variables from a Laplace distribution, $N_i \sim \text{Lap}(0, GS_1(f)/\varepsilon)$, and M is the number of attributes.

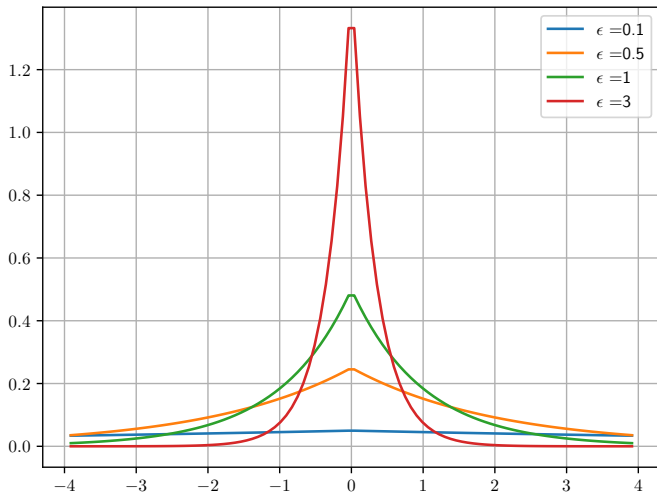
- The Laplace mechanism, using $N_i \sim \text{Lap}(0, GS_1(f)/\varepsilon)$, **satisfies ε -differential privacy** for f .
- Here we use norm ℓ_1 for GS !

Laplace distribution

$$\text{Lap}(\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right)$$

In our case, $\mu = 0$ and $b = GS_1(f)/\varepsilon$.

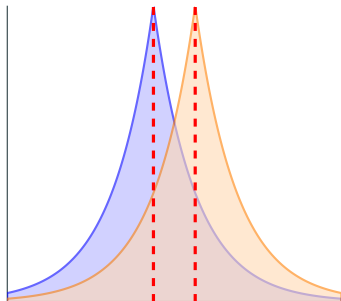
Recalling Laplace ii



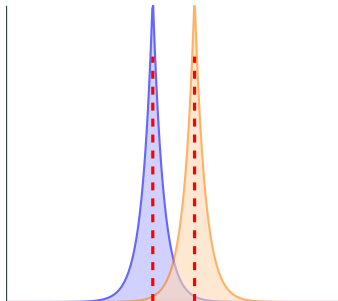
$Lap(0, 1/\epsilon)$ for $\epsilon = 0.1, 0.5, 1, 3$.

Laplace Mechanism: Effect of ε

Small ε (more noise)



Large ε (less noise)



Two queries, performed on neighboring databases (red dashed lines correspond to the original answer).

Properties of differential privacy

Composition theorems i

Sequential composition

Given $K_{q_1}, \dots, K_{q_n}$ mechanisms satisfying differential privacy with $\varepsilon_1, \dots, \varepsilon_n$ respectively, if applied to **the same database** D , then:

$$\varepsilon = \sum_{i=1}^n \varepsilon_i$$

Parallel composition

Given $K_{q_1}, \dots, K_{q_n}$ mechanisms satisfying differential privacy with $\varepsilon_1, \dots, \varepsilon_n$ respectively, if applied to **disjoint database** $D_1, \dots, D_n$, then:

$$\varepsilon = \max_{i=1}^n \varepsilon_i$$

Post-processing

Given K_q mechanism satisfying differential privacy with ε , then for any (deterministic or randomized) function g , $g(K_q(D))$ satisfies differential privacy for ε .

Approximate Differential Privacy and the Gaussian Mechanism

Approximate Differential Privacy $((\epsilon, \delta)$ -DP)

Definition

A randomized mechanism M with domain $\mathcal{D}$ and range $\mathcal{R}$ is (ϵ, δ) -differentially private if for all neighboring datasets D_1, D_2 and all measurable $S \subseteq \mathcal{R}$,

$$\Pr[M(D_1) \in S] \leq e^\epsilon \Pr[M(D_2) \in S] + \delta.$$

- **Parameters:** $\epsilon \geq 0$ (privacy loss), $0 \leq \delta < 1$ (failure probability).
- If $\delta = 0$ we get standard (pure) ϵ -DP.

Properties of (ε, δ) -differential privacy i

Sequential composition

K_{q_1}, K_{q_2} , with $(\varepsilon_1, \delta_1)$, and $(\varepsilon_2, \delta_2)$, respectively, if applied to **the same database** D , then:

$$(\varepsilon, \delta) = (\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$$

Sequential composition

K_{q_1}, K_{q_2} , with $(\varepsilon_1, \delta_1)$, and $(\varepsilon_2, \delta_2)$, respectively, if applied to **disjoint databases**, then:

$$(\varepsilon, \delta) = (\max \{\varepsilon_1, \varepsilon_2\}, \max \{\delta_1, \delta_2\})$$

Post-processing

If M is (ε, δ) -DP, then $f \circ M$ is also (ε, δ) -DP for any (randomized) f independent of D .

The Gaussian Mechanism

Let $f : \mathcal{D} \rightarrow \mathbb{R}^k$ be a query with ℓ_2 -sensitivity

$$GS_f = \max_{D \sim D'} \|f(D) - f(D')\|_2.$$

The **Gaussian mechanism** is defined as:

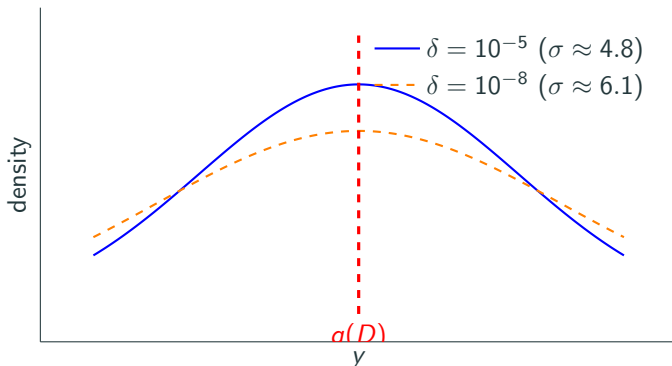
$$M(D) = f(D) + \mathcal{N}(0, \sigma^2).$$

where

$$\sigma^2 = \frac{2\Delta_2 \ln(1.25/\delta)}{\epsilon^2}$$

Impact of δ on the Gaussian Mechanism

Noise distributions for different δ (fix $\varepsilon = 1$, $\Delta_2 = 1$)



- Smaller $\delta \Rightarrow$ stronger privacy but more noise.
- Larger $\delta \Rightarrow$ less privacy, less noise.

Understanding ϵ vs δ

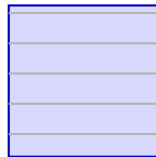
Key idea

- ϵ : controls the **everyday privacy leakage**. Smaller $\epsilon \Rightarrow$ outputs look almost the same with or without your data.
 $\rightsquigarrow \epsilon = \text{how much you leak most of the time.}$
- δ : a small **failure probability**. With probability δ , the guarantee of ϵ may not hold.
 $\rightsquigarrow \delta = \text{tiny chance of a bigger leak.}$

Metaphor

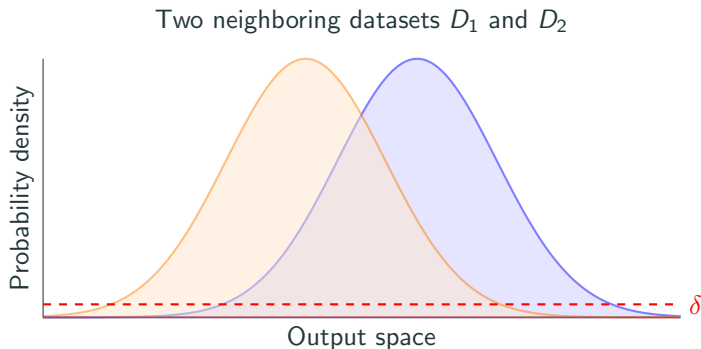
- $\epsilon = \text{tint of a window}$: how much you can see through it each time.
- $\delta = \text{chance the blinds open}$: very rare, but then you might see everything.

ϵ : tint



δ : rare chance blinds open

Visualizing the role of δ



- ε : controls how close the two distributions must be (multiplicative bound), or how big is the shared area.
- δ : allows a small additive slack. The bound can fail, but only with probability $\leq \delta$.

Randomized Response (RR)

Randomized Response

- We want to ask a sensitive question with YES/NO response:
 - “Have you consumed drugs during the last month?”
- Answer:

1. Toss a **coin**

- 1.1 if *heads* → tell the TRUTH
- 1.2 if *tails* → say YES

- The coin toss is private (only known to the person answering the question).
- Given a set of answers, we can estimate the true YES/NO answers.
- Provides **strong deniability** for YES answers.

RR example

- If distribution of head and tails is uniform.
- Then:

Coin	True answer	RR answer	Percentages
heads	YES	YES	50%
heads	NO	NO	
tails	YES	YES	50%
tails	NO	YES	

- NO answer does not provide strong deniability!

1. Toss **coin_1**:
 - 1.1 if *heads* → tell the TRUTH
 - 1.2 if *tails* → toss **coin_2**:
 - 1.2.1 if *heads* → say YES
 - 1.2.2 if *tails* → say NO

- Can you estimate now the number of true YES and NO?
- Usually the coin is replaced by probabilities p_1 and p_2 (not necessarily 0.5)

1. Toss **coin_1**:

1.1 $[p_1]$ if *heads* $\rightarrow$ tell the TRUTH

1.2 $[1 - p_1]$ if *tails* $\rightarrow$ toss **coin_2**:

1.2.1 $[p_2]$ if *heads* $\rightarrow$ say YES

1.2.2 $[1 - p_2]$ if *tails* $\rightarrow$ say NO

- Can you estimate now the number of true YES and NO?
- Usually the coin is replaced by probabilities p_1 and p_2 (not necessarily 0.5)

RR for non-binary categorical data

- Given a set of classes $C = \{c_1, \dots, c_c\}$
- Consider a probability distribution for each category.
- $P(c_i, c_j) = P(X' = c_j | X = c_i) = P(X' = c_j | c_i)$: probability that given a TRUE c_i we output c_j .
- We normally use a transition matrix of probabilities.

Example PRAM

- Suppose we have the category blood type in medical data $C = A, B, AB, O$. The probabilities could be:

$$\begin{pmatrix} 0.4 & 0.3 & 0.2 & 0.1 \\ 0.3 & 0.4 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.4 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.4 \end{pmatrix}$$

- $P(A, B) = 0.3$ $P(A, O) = 0.1$, $P(A, A) = 0.4$
- This method is also called **PRAM (Post-Randomization Method)** and has been widely used in data protection (before DP became popular)

- For the matrix to provide ε -differential privacy we need, for all i, j :

$$\max_{k=1}^c \frac{P(X' = c_k | c_i)}{P(X' = c_k | c_j)} \leq e^\varepsilon$$

More precisely, if we want ε -differential privacy, then:

$$P(X' = c_i | c_i) = e^\varepsilon / (c - 1 + e^\varepsilon)$$

$$P(X' = c_j | c_i) = 1 / (c - 1 + e^\varepsilon)$$

- [1] Cynthia Dwork and Aaron Roth.
The Algorithmic Foundations of Differential Privacy, volume 9 of *Foundations and Trends in Theoretical Computer Science*.
2014.
- [2] Staal A. Vinterbo.
An Introduction to Data Privacy.
Master edition edition, 2024.
- [3] Vicenç Torra.
Guide to Data Privacy: Models, Technologies, Solutions.
Undergraduate Topics in Computer Science. Springer International Publishing, Cham, 2022.