

CS 458 / 658: Computer Security and Privacy

Module 6 - Data Security and Privacy

Part 3 - Differential privacy

Meng Xu (*University of Waterloo*)

Winter 2023

Outline

- 1 The Dinur-Nissim reconstruction attack
- 2 The intuition behind differential privacy
- 3 A formal definition of differential privacy
- 4 Properties of the ϵ -DP definition
- 5 Perturbation mechanisms
- 6 More topics on differential privacy

We are being too honest...

In all the cases covered in Part 2, we always give a *faithful* aggregation result for each query sent from the data analyst.

For example:

- The SUM of the salaries
- The AVERAGE of ages in census data

We are being too honest...

In all the cases covered in Part 2, we always give a *faithful* aggregation result for each query sent from the data analyst.

For example:

- The SUM of the salaries
- The AVERAGE of ages in census data

Q: How about we add noise to the query response?

We are being too honest...

In all the cases covered in Part 2, we always give a *faithful* aggregation result for each query sent from the data analyst.

For example:

- The SUM of the salaries
- The AVERAGE of ages in census data

Q: How about we add noise to the query response?

A: It will make some of the attacks harder, but the **Dinur-Nissim reconstruction attack** illustrates why, when a mechanism adds too little noise when responding to aggregated queries, an adversary can still reconstruct the database **with high accuracy and efficiency**.

Formalize our setup

- There is a database, D , which potentially contains sensitive information about individuals.

Formalize our setup

- There is a database, D , which potentially contains sensitive information about individuals.
- The **database curator** has access to the full database.
We assume the curator is trusted.

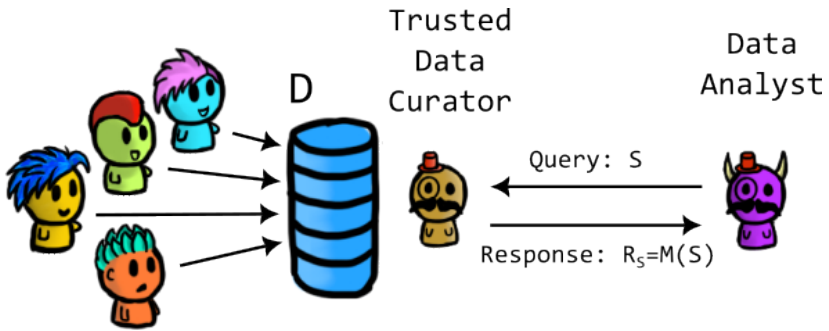
Formalize our setup

- There is a database, D , which potentially contains sensitive information about individuals.
- The **database curator** has access to the full database.
We assume the curator is trusted.
- The **data analyst** consumes the data by asking a series of **queries** to the curator. Each query is denoted as S and the curator provides a **response** to query S with R_S .
The analyst may be honest or malicious.

Formalize our setup

- There is a database, D , which potentially contains sensitive information about individuals.
- The **database curator** has access to the full database.
We assume the curator is trusted.
- The **data analyst** consumes the data by asking a series of **queries** to the curator. Each query is denoted as S and the curator provides a **response** to query S with R_S .
The analyst may be honest or malicious.
- The way in which the curator responds to queries is called the **mechanism**. Formally, $M : S \rightarrow R_S$. We'd like a mechanism that
 - gives statistically useful responses but
 - avoids leaking sensitive information about individuals.

Formalize our setup



Bad news: adding noise is tricky

Bad news: adding noise is tricky

Dinur-Nissim reconstruction attack: if the mechanism adds too little noise when responding to aggregated queries, an adversary can reconstruct the database *with high accuracy and efficiency*.

Bad news: adding noise is tricky

Dinur-Nissim reconstruction attack: if the mechanism adds too little noise when responding to aggregated queries, an adversary can reconstruct the database *with high accuracy and efficiency*.

Such a mechanism is called **blatantly non-private**.

Attack setup

We consider the database to be a collection of n records

$$D = \{d_1, d_2, \dots, d_n\}$$

where each record corresponds to one individual.

Attack setup

We consider the database to be a collection of n records

$$D = \{d_1, d_2, \dots, d_n\}$$

where each record corresponds to one individual.

Each record d_i may consist of k attributes. For simplicity, we assume that the adversary already knows $k - 1$ attribute for all records and the only attribute unknown to the adversary is a single bit.

$$D = \left[\begin{array}{cccc|c} a_{\{1,1\}} & a_{\{1,2\}} & \cdots & a_{\{1,k-1\}} & b_1 \\ a_{\{2,1\}} & a_{\{2,2\}} & \cdots & a_{\{2,k-1\}} & b_2 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{\{n,1\}} & a_{\{n,2\}} & \cdots & a_{\{n,k-1\}} & b_n \end{array} \right]$$

Attack setup example

Name	ZIP	DOB	COVID
Alice	K8V 7R6	5/2/1984	1
Bob	V5K 5J9	2/8/2001	0
Charlie	V1C 7J2	10/10/1954	1
David	R4K 5T1	4/4/1944	0
Eve	G7N 8Y3	1/1/1980	1
... 995 more entries ...			

We assume the adversary **knows all but one** attributes. The unknown attribute is a binary bit (e.g., COVID)

Threat model

The attacker is allowed to ask aggregated queries, and perhaps the most basic type of aggregate query in this case is a counting query, i.e., how many records in D that satisfies a condition $C(a_{\{*,1\}}, a_{\{*,2\}}, \dots, a_{\{*,k-1\}})$ have their secret bit set to 1?

Threat model

The attacker is allowed to ask aggregated queries, and perhaps the most basic type of aggregate query in this case is a counting query, i.e., how many records in D that satisfies a condition $C(a_{\{*,1\}}, a_{\{*,2\}}, \dots, a_{\{*,k-1\}})$ have their secret bit set to 1?

For example: How many rows satisfying condition (Name = "David" OR DOB > 1980) have COVID = 1.

Threat model

The attacker is allowed to ask aggregated queries, and perhaps the most basic type of aggregate query in this case is a counting query, i.e., how many records in D that satisfies a condition $C(a_{\{*,1\}}, a_{\{*,2\}}, \dots, a_{\{*,k-1\}})$ have their secret bit set to 1?

For example: How many rows satisfying condition (Name = "David" OR DOB > 1980) have COVID = 1.

The key point is, the adversary is allowed to pick **arbitrary rows** in the database using their **background knowledge** to formulate queries. Formally, $S \in \{0, 1\}^n$. An example is $S = [0, 1, 1, 1, \dots, 0]$

Counting query

Formally, a counting query is represented by $S \in \{0, 1\}^n$.

Counting query

Formally, a counting query is represented by $S \in \{0, 1\}^n$.

As a concrete example, if $n = 5$,

Name	ZIP	DOB	COVID
Alice	K8V 7R6	5/2/1984	1
Bob	V5K 5J9	2/8/2001	0
Charlie	V1C 7J2	10/10/1954	1
David	R4K 5T1	4/4/1944	0
Eve	G7N 8Y3	1/1/1980	1

- $S = [1, 0, 1, 0, 0]$ will be asking for $b_1 + b_3$, which is 2
- $S = [0, 1, 0, 1, 1]$ will be asking for $b_2 + b_4 + b_5$, which is 1

Curator mechanism

Upon receiving a query S , the curator will first calculate the true answer $A(S) = S \times [b_1, b_2, \dots, b_n]$.

$$R_S = A(S)$$

Curator mechanism

Upon receiving a query S , the curator will first calculate the true answer $A(S) = S \times [b_1, b_2, \dots, b_n]$.

$$R_S = A(S) + r$$

And subsequently, add a **random** noise r to the true answer.

Curator mechanism

Upon receiving a query S , the curator will first calculate the true answer $A(S) = S \times [b_1, b_2, \dots, b_n]$.

$$R_S = A(S) + r$$

And subsequently, add a **random** noise r to the true answer.

Let's consider a noise that is upper-bounded by:

$$|r| \leq E$$

Q: What are the pros/cons of using a noise with large upper bound?

The inefficient attack

Theorem: If the analyst is allowed to ask 2^n queries to a dataset of n users, and the curator adds noise with some bound E , then based on the results, the adversary can reconstruct the database in all but at most $4E$ positions.

The inefficient attack

Theorem: If the analyst is allowed to ask 2^n queries to a dataset of n users, and the curator adds noise with some bound E , then based on the results, the adversary can reconstruct the database in all but at most $4E$ positions.

e.g., $E = \frac{n}{400} \implies$ reconstruction of 99% entries in the database.

The inefficient attack

Theorem: If the analyst is allowed to ask 2^n queries to a dataset of n users, and the curator adds noise with some bound E , then based on the results, the adversary can reconstruct the database in all but at most $4E$ positions.

e.g., $E = \frac{n}{400} \implies$ reconstruction of 99% entries in the database.

Algorithm:

- For an attacker, there are only 2^n database candidates.
- For each candidate database $C \in \{0, 1\}^n$, if there exists a query S such that $|\sum_{i \in S} C[i] - R_S| > E$, rule out C .
- Any database candidate not ruled out (C) differs with the actual database (D) by $4E$ at max.

The inefficient attack: an example

In the example, we have a database with $n = 3$ users (rows):

$$D = \left[\begin{array}{cccc|c} a_{\{1,1\}} & a_{\{1,2\}} & \cdots & a_{\{1,k-1\}} & b_1 \\ a_{\{2,1\}} & a_{\{2,2\}} & \cdots & a_{\{2,k-1\}} & b_2 \\ a_{\{3,1\}} & a_{\{3,2\}} & \cdots & a_{\{3,k-1\}} & b_3 \end{array} \right]$$

The adversary queries for all 2^n combinations $\{0, 1\}^n$, i.e.,
 $S \in \{[0, 0, 0], [0, 0, 1], [0, 1, 0], \dots, [1, 1, 1]\}$

The curator uses noise r sampled randomly from $\{-0.5, +0.5\}$.

The inefficient attack: an example

In the example, we have a database with $n = 3$ users (rows):

$$D = \left[\begin{array}{cccc|c} a_{\{1,1\}} & a_{\{1,2\}} & \cdots & a_{\{1,k-1\}} & b_1 \\ a_{\{2,1\}} & a_{\{2,2\}} & \cdots & a_{\{2,k-1\}} & b_2 \\ a_{\{3,1\}} & a_{\{3,2\}} & \cdots & a_{\{3,k-1\}} & b_3 \end{array} \right]$$

The adversary queries for all 2^n combinations $\{0, 1\}^n$, i.e.,
 $S \in \{[0, 0, 0], [0, 0, 1], [0, 1, 0], \dots, [1, 1, 1]\}$

The curator uses noise r sampled randomly from $\{-0.5, +0.5\}$.

Q: At least how many bits can we reconstruct?

The inefficient attack: an example

In the example, we have a database with $n = 3$ users (rows):

$$D = \left[\begin{array}{cccc|c} a_{\{1,1\}} & a_{\{1,2\}} & \cdots & a_{\{1,k-1\}} & b_1 \\ a_{\{2,1\}} & a_{\{2,2\}} & \cdots & a_{\{2,k-1\}} & b_2 \\ a_{\{3,1\}} & a_{\{3,2\}} & \cdots & a_{\{3,k-1\}} & b_3 \end{array} \right]$$

The adversary queries for all 2^n combinations $\{0, 1\}^n$, i.e.,
 $S \in \{[0, 0, 0], [0, 0, 1], [0, 1, 0], \dots, [1, 1, 1]\}$

The curator uses noise r sampled randomly from $\{-0.5, +0.5\}$.

Q: At least how many bits can we reconstruct?

A: At least 1 bit

The inefficient attack: an example

True database has:

$B=[1,0,1]$

Query



$S=[0,0,0]$

$S=[0,0,1]$

$S=[0,1,0]$

$S=[0,1,1]$

$S=[1,0,0]$

$S=[1,0,1]$

$S=[1,1,0]$

$S=[1,1,1]$

Candidate Binary Vectors

$C=[0,0,0]$

$C=[0,0,1]$

$C=[0,1,0]$

$C=[0,1,1]$

$C=[1,0,0]$

$C=[1,0,1]$

$C=[1,1,0]$

$C=[1,1,1]$

0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	1
0	0	1	1	0	0	1	1	1
0	1	1	2	0	1	1	1	2
0	0	0	0	1	1	1	1	1
0	1	0	1	1	2	1	1	2
0	0	1	1	1	1	2	2	2
0	1	1	2	1	2	2	2	3

Expected answers
(without noise)
for each candidate



Noise sampled
at random from
 $\{-0.5, +0.5\}$

The inefficient attack: an example

True database has:

$B=[1,0,1]$

Query True answer Noise (secret) Reported answer

$S=[0,0,0]$

$R_s=0+(+0.5)=0.5$

$S=[0,0,1]$

$R_s=1+(-0.5)=0.5$

$S=[0,1,0]$

$S=[0,1,1]$

$S=[1,0,0]$

$S=[1,0,1]$

$S=[1,1,0]$

$S=[1,1,1]$

Candidate Binary Vectors

$C=[0,0,0]$
 $C=[0,0,1]$
 $C=[0,1,0]$
 $C=[0,1,1]$
 $C=[1,0,0]$
 $C=[1,0,1]$
 $C=[1,1,0]$
 $C=[1,1,1]$

0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	1	1	2	0	1	1	2
0	0	0	0	1	1	1	1
0	1	0	1	1	2	1	2
0	0	1	1	1	1	2	2
0	1	1	2	1	2	2	3

Expected answers
 (without noise)
 for each candidate

Noise sampled
 at random from
 $\{-0.5, +0.5\}$

The inefficient attack: an example

True database has:

$B=[1,0,1]$

Query True answer Noise (secret) Reported answer

$S=[0,0,0]$	$R_S=0+(+0.5)=0.5$
$S=[0,0,1]$	$R_S=1+(-0.5)=0.5$
$S=[0,1,0]$	$R_S=0+(+0.5)=0.5$
$S=[0,1,1]$	$R_S=1+(+0.5)=1.5$
$S=[1,0,0]$	$R_S=1+(-0.5)=0.5$
$S=[1,0,1]$	$R_S=2+(-0.5)=1.5$
$S=[1,1,0]$	$R_S=1+(-0.5)=0.5$
$S=[1,1,1]$	$R_S=2+(-0.5)=1.5$

Candidate Binary Vectors

$C=[0,0,0]$
 $C=[0,0,1]$
 $C=[0,1,0]$
 $C=[0,1,1]$
 $C=[1,0,0]$
 $C=[1,0,1]$
 $C=[1,1,0]$
 $C=[1,1,1]$

0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	1	1	2	0	1	1	2
0	0	0	0	1	1	1	1
0	1	0	1	1	2	1	2
0	0	1	1	1	1	2	2
0	1	1	2	1	2	2	3

Expected answers
(without noise)
for each candidate

Noise sampled
at random from
 $\{-0.5, +0.5\}$

The inefficient attack: an example

True database has:

$B = [1, 0, 1]$

Query	True answer	Noise (secret)	Reported answer
$S = [0, 0, 0]$		$R_S = 0 + (+0.5) = 0.5$	
$S = [0, 0, 1]$		$R_S = 1 + (-0.5) = 0.5$	
$S = [0, 1, 0]$		$R_S = 0 + (+0.5) = 0.5$	
$S = [0, 1, 1]$		$R_S = 1 + (+0.5) = 1.5$	
$S = [1, 0, 0]$		$R_S = 1 + (-0.5) = 0.5$	
$S = [1, 0, 1]$		$R_S = 2 + (-0.5) = 1.5$	
$S = [1, 1, 0]$		$R_S = 1 + (-0.5) = 0.5$	
$S = [1, 1, 1]$		$R_S = 2 + (-0.5) = 1.5$	

Candidate Binary Vectors

	$C = [0, 0, 0]$	$C = [0, 0, 1]$	$C = [0, 1, 0]$	$C = [0, 1, 1]$	$C = [1, 0, 0]$	$C = [1, 0, 1]$	$C = [1, 1, 0]$	$C = [1, 1, 1]$
0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	1
0	0	1	1	0	0	1	1	1
1	1	2		1	1	1	2	2
0	0	0	0	1	1	1	1	1
1		1	1	2	1	2		2
0	0	1	1	1	1			
1	1	2	1	2		2		2

Expected answers (without noise) for each candidate

Noise sampled at random from $\{-0.5, +0.5\}$

The inefficient attack: more practice

True database has:
 $B=[?, ?, ?]$

Query

- $S=[0, 0, 0]$
- $S=[0, 0, 1]$
- $S=[0, 1, 0]$
- $S=[0, 1, 1]$
- $S=[1, 0, 0]$
- $S=[1, 0, 1]$
- $S=[1, 1, 0]$
- $S=[1, 1, 1]$

Candidate Binary Vectors

- $C=[0, 0, 0]$
- $C=[0, 0, 1]$
- $C=[0, 1, 0]$
- $C=[0, 1, 1]$
- $C=[1, 0, 0]$
- $C=[1, 0, 1]$
- $C=[1, 1, 0]$
- $C=[1, 1, 1]$

Reported answer

- $R_S=0.5$
- $R_S=0.5$
- $R_S=1.0$
- $R_S=1.0$
- $R_S=0.5$
- $R_S=1.5$
- $R_S=1.5$
- $R_S=2.5$

0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	1	1	2	0	1	1	2
0	0	0	0	1	1	1	1
0	1	0	1	1	2	1	2
0	0	1	1	1	1	2	2
0	1	1	2	1	2	2	3

Expected answers
(without noise)
for each candidate



Noise sampled
at random from
{-0.5, 0, +0.5}

Q: Can you guess the privacy-sensitive column B (or a list of candidates)

The inefficient attack: more practice

True database has:
 $B = [?, ?, ?]$

Query

- $S = [0, 0, 0]$
- $S = [0, 0, 1]$
- $S = [0, 1, 0]$
- $S = [0, 1, 1]$
- $S = [1, 0, 0]$
- $S = [1, 0, 1]$
- $S = [1, 1, 0]$
- $S = [1, 1, 1]$

Candidate Binary Vectors

- $C = [0, 0, 0]$
- $C = [0, 0, 1]$
- $C = [0, 1, 0]$
- $C = [0, 1, 1]$
- $C = [1, 0, 0]$
- $C = [1, 0, 1]$
- $C = [1, 1, 0]$
- $C = [1, 1, 1]$

Reported answer

- $R_S = 0.5$
- $R_S = 0.5$
- $R_S = 1.0$
- $R_S = 1.0$
- $R_S = 0.5$
- $R_S = 1.5$
- $R_S = 1.5$
- $R_S = 2.5$

0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	1	1	2	0	1	1	2
0	0	0	0	1	1	1	1
0	1	0	1	1	2	1	2
0	0	1	1	1	1	2	2
0	1	1	2	1	2	2	3

Expected answers
(without noise)
for each candidate



Noise sampled
at random from
 $\{-0.5, 0, +0.5\}$

Q: Can you guess the privacy-sensitive column B (or a list of candidates)

A: There is only one candidate: $B = [1, 1, 0]$

The inefficient attack proof

Proof: Any database candidate not ruled out (C) differs with the actual database (D) by $4E$ at max

Consider query $I_0 \leftarrow \{i | D[i] = 0\}$, we know that

$$|\sum_{i \in I_0} C[i] - R_{I_0}| \leq E, |\sum_{i \in I_0} D[i] - R_{I_0}| \leq E, \implies \sum_{i \in I_0} |C[i] - D[i]| \leq 2E$$

Consider query $I_1 \leftarrow \{i | D[i] = 1\}$, we know that

$$|\sum_{i \in I_1} C[i] - R_{I_1}| \leq E, |\sum_{i \in I_1} D[i] - R_{I_1}| \leq E, \implies \sum_{i \in I_1} |C[i] - D[i]| \leq 2E$$

The efficient attack

Theorem: If the analyst is allowed to ask $O(n)$ queries to a dataset of n users, and the curator adds noise with some bound $E = O(\alpha\sqrt{n})$, then based on the results, a computationally efficient adversary can reconstruct the database in all but at most $O(\alpha^2 n)$ positions.

Blatantly non-private

Definition: A mechanism is **blatantly non-private** if an adversary can reconstruct a database that matches with the true database in all but $o(n)$ entries.

Blatantly non-private

Definition: A mechanism is **blatantly non-private** if an adversary can reconstruct a database that matches with the true database in all but $o(n)$ entries.

NOTE 1: According to the efficient attack scenario, adding a noise of $O(\sqrt{n})$ is blatantly non-private.

Blatantly non-private

Definition: A mechanism is **blatantly non-private** if an adversary can reconstruct a database that matches with the true database in all but $o(n)$ entries.

NOTE 1: According to the efficient attack scenario, adding a noise of $O(\sqrt{n})$ is blatantly non-private.

NOTE 2: This definition does not specify whether a mechanism is private. Instead, it defines a criteria to show that a mechanism is clearly not private.

Blatantly non-private

Definition: A mechanism is **blatantly non-private** if an adversary can reconstruct a database that matches with the true database in all but $o(n)$ entries.

NOTE 1: According to the efficient attack scenario, adding a noise of $O(\sqrt{n})$ is blatantly non-private.

NOTE 2: This definition does not specify whether a mechanism is private. Instead, it defines a criteria to show that a mechanism is clearly not private.

Differential privacy, on the other hand, is a definition on whether a mechanism is private.

Outline

- ① The Dinur-Nissim reconstruction attack
- ② The intuition behind differential privacy**
- ③ A formal definition of differential privacy
- ④ Properties of the ϵ -DP definition
- ⑤ Perturbation mechanisms
- ⑥ More topics on differential privacy

So..., more noise maybe?

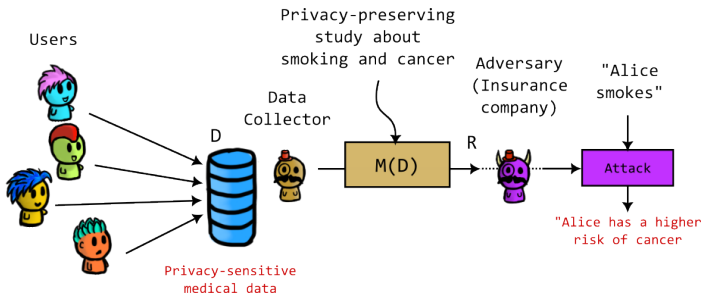
We add more noise such that the adversary cannot reconstruct the database. But how much more is more?

So..., more noise maybe?

We add more noise such that the adversary cannot reconstruct the database. But how much more is more?

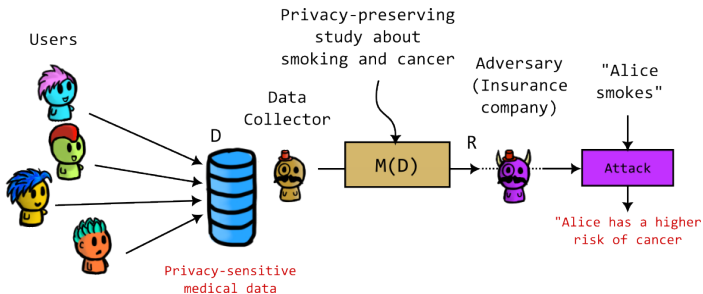
Well, that depends on what your privacy goal is.

Example: strong auxiliary information



A study proved that smoking and cancer are correlated. Thanks to the study, the adversary learns that Alice has higher risk of cancer.

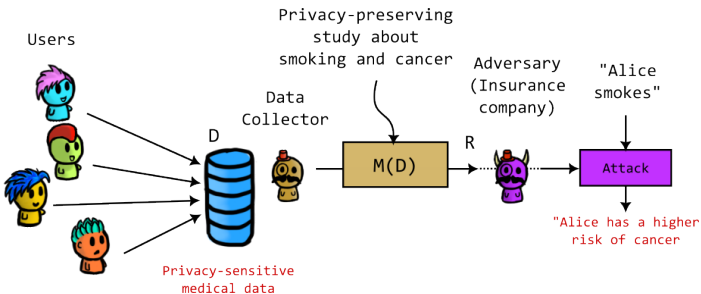
Example: strong auxiliary information



A study proved that smoking and cancer are correlated. Thanks to the study, the adversary learns that Alice has higher risk of cancer.

Q: Is this a violation of Alice's privacy? Is this the study's fault? Should we design an M to prevent this?

Example: strong auxiliary information

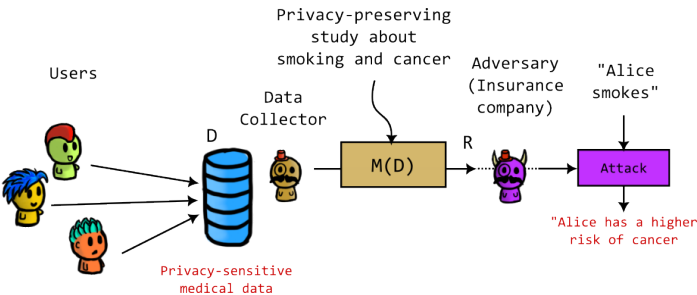
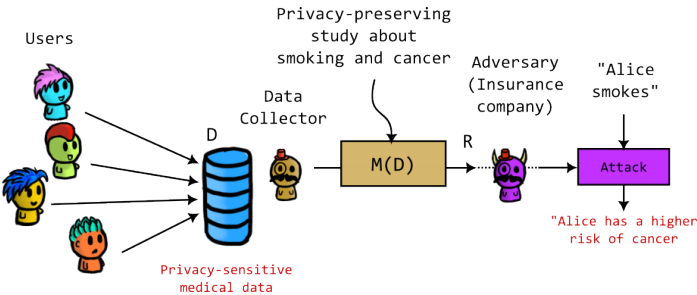


A study proved that smoking and cancer are correlated. Thanks to the study, the adversary learns that Alice has higher risk of cancer.

Q: Is this a violation of Alice's privacy? Is this the study's fault? Should we design an M to prevent this?

A: The adversary would've reached the same conclusion even if Alice hadn't participated in the study! We cannot prevent this (without completely destroying utility, i.e., not doing the study).

Example: strong auxiliary information



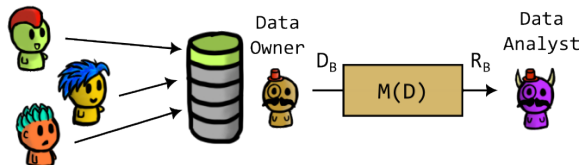
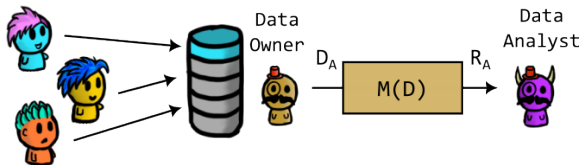
Possible privacy goal

We cannot guarantee **absolute privacy** — if the adversary has sufficiently strong background information, there is nothing M can do about it!

We should instead ensure that the adversary cannot gain (*significantly*) *new* information from R (i.e., we want a “**differential**” and not an “**absolute**” privacy)

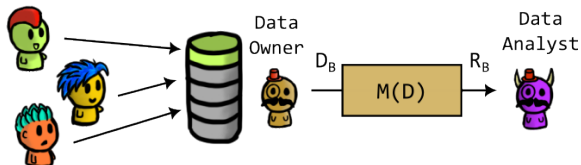
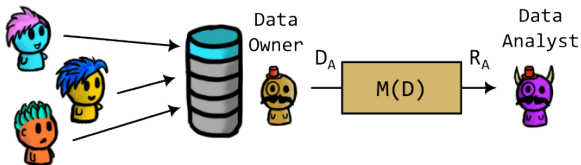
Possible privacy goal

What if we try to make these cases similar?



Possible privacy goal

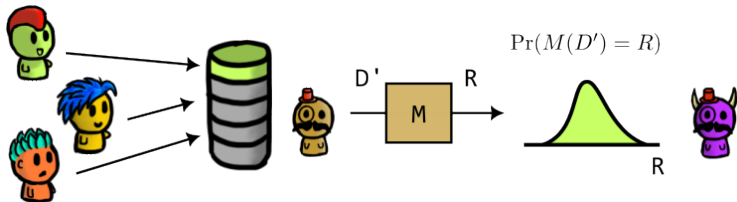
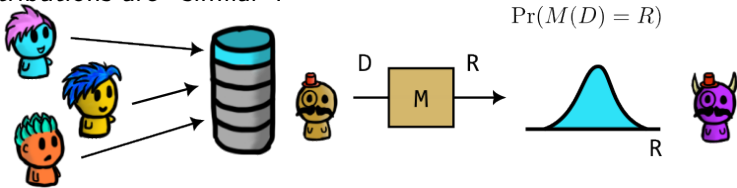
What if we try to make these cases similar?



- We want the R_A and R_B to be “similar” ($R_A \approx R_B$).
- This would ensure M does not depend “too much” on any single user.

Possible privacy goal

In addition, note that M is randomized (e.g., adds noise). Thus, instead of ensuring $R_A \approx R_B$, we ensure their probability distributions are “similar”.



i.e., for all R , the chance of producing R by D and D' are close:
 $\Pr(M(D) = R) \approx \Pr(M(D') = R)$

A bit more formal on the possible privacy goal

Consider a setting where

- I hand in my data to a database D (which is trusted),
- an algorithm A runs over D and releases a set of data T ,
- the adversary knows the details of A and has access to T .

A bit more formal on the possible privacy goal

Consider a setting where

- I hand in my data to a database D (which is trusted),
- an algorithm A runs over D and releases a set of data T ,
- the adversary knows the details of A and has access to T .

A privacy notion: The adversary learns (almost) **nothing new** about me even after seeing A and T , and regardless of what **other datasets** are available.

A bit more formal on the possible privacy goal

Consider a setting where

- I hand in my data to a database D (which is trusted),
- an algorithm A runs over D and releases a set of data T ,
- the adversary knows the details of A and has access to T .

A privacy notion: The adversary learns (almost) **nothing new** about me even after seeing A and T , and regardless of what **other datasets** are available.

This privacy notion makes no assumption about what background knowledge the adversary might possess:

- If the adversary does not know whether I am in the database, it won't know that either after seeing the result.
- If the adversary already knows whether I am in the database, it won't know more about the secret values I supplied.

An example from the attacker's perspective

An example from the attacker's perspective

Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of $[45, 55]$.

An example from the attacker's perspective

Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of $[45, 55]$.

Algorithm: You are given an algorithm that

- allows you to make 5 queries,
- each query returns the average score of 3 randomly selected students (out of 30 scores in total).

An example from the attacker's perspective

Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of $[45, 55]$.

Algorithm: You are given an algorithm that

- allows you to make 5 queries,
- each query returns the average score of 3 randomly selected students (out of 30 scores in total).

Q: How can you infer whether Alice is enrolled in CS458 or not?

The attack

Just send 5 queries and observe what is returned by the database.

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
 - Everyone else (29 of them): 50
-

D2 with Alice not enrolled:

- Everyone (30 of them): 50

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Q: What will happen if Alice IS NOT enrolled (i.e., D2)?

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Q: What will happen if Alice IS NOT enrolled (i.e., D2)?

A: Expect [50, 50, 50, 50, 50] in response.

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Q: What will happen if Alice IS NOT enrolled (i.e., D2)?

A: Expect [50, 50, 50, 50, 50] in response.

Q: What will happen if Alice IS enrolled (i.e., D1)?

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50
-

Q: What will happen if Alice IS NOT enrolled (i.e., D2)?

A: Expect [50, 50, 50, 50, 50] in response.

Q: What will happen if Alice IS enrolled (i.e., D1)?

A: For a single response, we either get

- $63 \leftrightarrow \frac{C_{29}^2}{C_{30}^3} = 10\%$
- $50 \leftrightarrow$ otherwise

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Q: What will happen if Alice IS NOT enrolled (i.e., D2)?

A: Expect [50, 50, 50, 50, 50] in response.

Q: What will happen if Alice IS enrolled (i.e., D1)?

A: For a single response, we either get

- $63 \leftrightarrow \frac{C_{29}^2}{C_{30}^3} = 10\%$
- $50 \leftrightarrow$ otherwise

For all 5 responses, the chance of getting at least one 63 is

$$1 - \left(1 - \frac{C_{29}^2}{C_{30}^3}\right)^5 = 40.95\%$$

What went wrong?

Alice's score has too much impact on the output! As a result, seeing the output of the algorithm allows the attacker to differentiate which database is the underlying database representing the class score.

What went wrong?

Alice's score has too much impact on the output! As a result, seeing the output of the algorithm allows the attacker to differentiate which database is the underlying database representing the class score.

This is exactly what *Differential Privacy (DP)* tries to capture!

What went wrong?

Alice's score has too much impact on the output! As a result, seeing the output of the algorithm allows the attacker to differentiate which database is the underlying database representing the class score.

This is exactly what *Differential Privacy (DP)* tries to capture!

Informally, the DP notion requires any single element in a dataset to have only a limited impact on the output.

The defense

The defense

Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of $[45, 55]$.

Algorithm: You are given an algorithm that

- allows you to make 5 queries,
- each query returns the average score of 3 randomly selected students (out of 30 scores in total)

The defense

Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of $[45, 55]$.

Algorithm: You are given an algorithm that

- allows you to make 5 queries,
- each query returns the average score of 3 randomly selected students (out of 30 scores in total) **plus a random value**

Demo time (dp-demo.py)

The data collectors' argument

... on trying to persuade you to join a differentially private survey:

You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.

The data collectors' argument

... on trying to persuade you to join a differentially private survey:

You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.

But this is only true if they tell you what algorithm they use to release your data and you have verified that their algorithm is indeed differentially private.

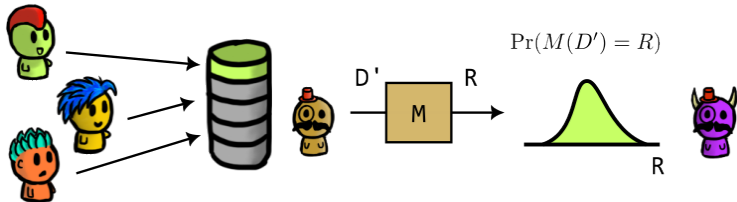
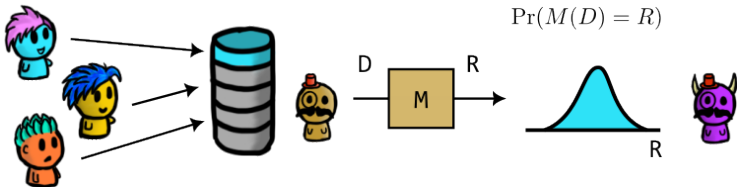
Outline

- ① The Dinur-Nissim reconstruction attack
- ② The intuition behind differential privacy
- ③ A formal definition of differential privacy
- ④ Properties of the ϵ -DP definition
- ⑤ Perturbation mechanisms
- ⑥ More topics on differential privacy

Formalize our setup

- There is a database, D , which potentially contains sensitive information about individuals.
- The **database curator** has access to the full database.
We assume the curator is trusted.
- The **data analyst** consumes the data by asking a series of **queries** to the curator. Each query is denoted as S and the curator provides a **response** to query S with R_S .
The analyst may be honest or malicious.
- The way in which the curator responds to queries is called the **mechanism**. Formally, $M : S \rightarrow R_S$. We'd like a mechanism that
 - gives statistically useful responses but
 - avoids leaking sensitive information about individuals.

Our informal privacy goal



For all R , the chance of producing R by D and D' are close:
 $\Pr[M(D) = R] \approx \Pr[M(D') = R]$

Q: How do we define close?

Neighboring databases

Two databases D and D' are **neighbouring** if they agree except for a single entry.

Neighboring databases

Two databases D and D' are **neighbouring** if they agree except for a single entry.

- *Unbounded DP*: D and D' are neighboring if D' can be obtained from D by removing one element
- *Bounded DP*: D and D' are neighboring if D' can be obtained from D by replacing one element

Neighboring databases

Two databases D and D' are **neighbouring** if they agree except for a single entry.

- *Unbounded DP*: D and D' are neighboring if D' can be obtained from D by removing one element
- *Bounded DP*: D and D' are neighboring if D' can be obtained from D by replacing one element

These are just slightly different guarantees of privacy. It is important to know which one your DP algorithm is providing. In practice, there is not a big difference.

How do we define “close” distributions?

Tentative privacy definition (this is not an actual definition)

A mechanism M is private (with some privacy parameter p) if the following holds for all possible outputs R and all pairs of neighboring datasets (D, D') :

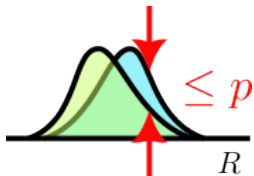
$$\Pr[M(D') = R] - p \leq \Pr[M(D) = R] \leq \Pr[M(D') = R] + p$$

How do we define “close” distributions?

Tentative privacy definition (this is not an actual definition)

A mechanism M is private (with some privacy parameter p) if the following holds for all possible outputs R and all pairs of neighboring datasets (D, D') :

$$\Pr[M(D') = R] - p \leq \Pr[M(D) = R] \leq \Pr[M(D') = R] + p$$

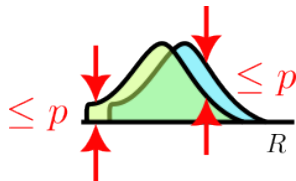
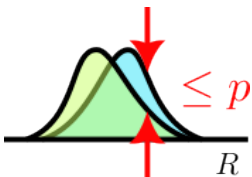


How do we define “close” distributions?

Tentative privacy definition (this is not an actual definition)

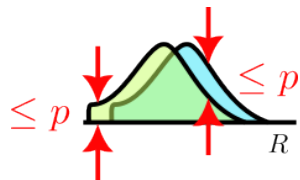
A mechanism M is private (with some privacy parameter p) if the following holds for all possible outputs R and all pairs of neighboring datasets (D, D') :

$$\Pr[M(D') = R] - p \leq \Pr[M(D) = R] \leq \Pr[M(D') = R] + p$$



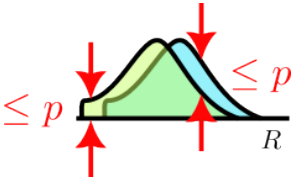
Potential issues with this closeness measurement

Q: What can go wrong with this case?



Potential issues with this closeness measurement

Q: What can go wrong with this case?



A: Suppose we have:

- $\epsilon = 0.01$
- $\Pr[M(D) = R] = 0.005$
- $\Pr[M(D') = R] = 0.001$
- $\epsilon = 0.01$
- $\Pr[M(D) = R] = 0.96$
- $\Pr[M(D') = R] = 0.94$

What if we make the distance multiplicative?

Tentative privacy definition II (this is not an actual definition)

A mechanism M is private (with some privacy parameter p) if the following holds for all possible outputs R and all pairs of neighboring datasets (D, D') :

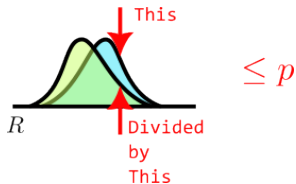
$$\Pr[M(D') = R] \cdot \frac{1}{p} \leq \Pr[M(D) = R] \leq \Pr[M(D') = R] \cdot p$$

What if we make the distance multiplicative?

Tentative privacy definition II (this is not an actual definition)

A mechanism M is private (with some privacy parameter p) if the following holds for all possible outputs R and all pairs of neighboring datasets (D, D') :

$$\Pr[M(D') = R] \cdot \frac{1}{p} \leq \Pr[M(D) = R] \leq \Pr[M(D') = R] \cdot p$$

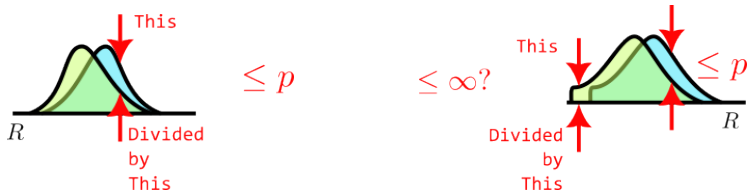


What if we make the distance multiplicative?

Tentative privacy definition II (this is not an actual definition)

A mechanism M is private (with some privacy parameter p) if the following holds for all possible outputs R and all pairs of neighboring datasets (D, D') :

$$\Pr[M(D') = R] \cdot \frac{1}{p} \leq \Pr[M(D) = R] \leq \Pr[M(D') = R] \cdot p$$



Almost there...

Instead of using p , we can use e^p as the privacy parameter:

Tentative privacy definition III (this is not an actual definition)

A mechanism M is private (with some privacy parameter p) if the following holds for all possible outputs R and all pairs of neighboring datasets (D, D') :

$$\Pr[M(D') = R] \cdot \frac{1}{e^p} \leq \Pr[M(D) = R] \leq \Pr[M(D') = R] \cdot e^p$$

Almost there...

Instead of using p , we can use e^p as the privacy parameter:

Tentative privacy definition III (this is not an actual definition)

A mechanism M is private (with some privacy parameter p) if the following holds for all possible outputs R and all pairs of neighboring datasets (D, D') :

$$\Pr[M(D') = R] \cdot \frac{1}{e^p} \leq \Pr[M(D) = R] \leq \Pr[M(D') = R] \cdot e^p$$

Further with the symmetry between D and D' , we can simplify the definition as:

$$\Pr[M(D) = R] \leq \Pr[M(D') = R] \cdot e^p$$

ϵ -differential privacy

Idea: If the mechanism M behaves nearly identically for D and D' , then an attacker can't tell whether D or D' was used (and hence can't learn much about the individual).

ϵ -differential privacy

Idea: If the mechanism M behaves nearly identically for D and D' , then an attacker can't tell whether D or D' was used (and hence can't learn much about the individual).

What we have so far

A mechanism M is private (with some privacy parameter p) if the following holds for all possible outputs R and all pairs of neighboring datasets (D, D'):

$$\Pr[M(D) = R] \leq \Pr[M(D') = R] \cdot e^p$$

ϵ -differential privacy

Idea: If the mechanism M behaves nearly identically for D and D' , then an attacker can't tell whether D or D' was used (and hence can't learn much about the individual).

What we have so far

A mechanism M is private (with some privacy parameter p) if the following holds for all possible outputs R and all pairs of neighboring datasets (D, D') :

$$\Pr[M(D) = R] \leq \Pr[M(D') = R] \cdot e^p$$

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

ϵ -DP elaboration: perspective

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

ϵ -DP elaboration: perspective

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

The $\forall T \subseteq Y$ means that the attacker cannot find a **perspective** through which the two databases behaves differently.

ϵ -DP elaboration: perspective

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

The $\forall T \subseteq Y$ means that the attacker cannot find a **perspective** through which the two databases behaves differently.

In the CS458 grades example, for a single query,

- $M : \{\text{Name} \times [0 - 100]\} \rightarrow [0 - 100]$
- $T : [60 - 100]$
- $\Pr[M(D_1) \in T] = 10\%$
- $\Pr[M(D_2) \in T] = 0\%$

ϵ -DP elaboration: interpreting ϵ

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

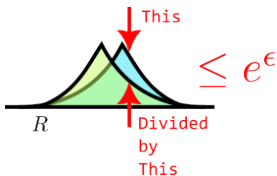
ϵ -DP elaboration: interpreting ϵ

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

$\epsilon \in [0, \infty) \implies$ this ensures that $e^\epsilon \in [1, \infty)$



ϵ -DP elaboration: interpreting ϵ

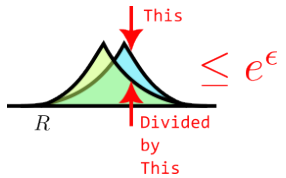
ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

$\epsilon \in [0, \infty) \implies$ this ensures that $e^\epsilon \in [1, \infty)$

Q: Which is “more private”: $\epsilon = 1$ or $\epsilon = 2$?



ϵ -DP elaboration: interpreting ϵ

ϵ -DP

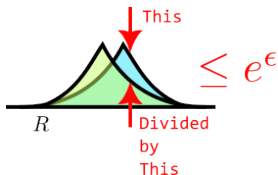
A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

$\epsilon \in [0, \infty) \implies$ this ensures that $e^\epsilon \in [1, \infty)$

Q: Which is “more private”: $\epsilon = 1$ or $\epsilon = 2$?

A: Smaller ϵ means more privacy



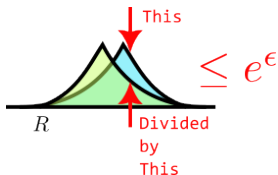
ϵ -DP elaboration: interpreting ϵ

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

$\epsilon \in [0, \infty) \implies$ this ensures that $e^\epsilon \in [1, \infty)$



Q: Which is “more private”: $\epsilon = 1$ or $\epsilon = 2$?

A: Smaller ϵ means more privacy

Q: What does $\epsilon = 0$ mean?

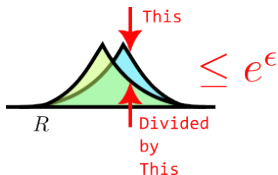
ϵ -DP elaboration: interpreting ϵ

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

$\epsilon \in [0, \infty) \implies$ this ensures that $e^\epsilon \in [1, \infty)$



Q: Which is “more private”: $\epsilon = 1$ or $\epsilon = 2$?

A: Smaller ϵ means more privacy

Q: What does $\epsilon = 0$ mean?

A: Perfect privacy!

ϵ -DP elaboration: value of ϵ

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

ϵ -DP elaboration: value of ϵ

ϵ -DP

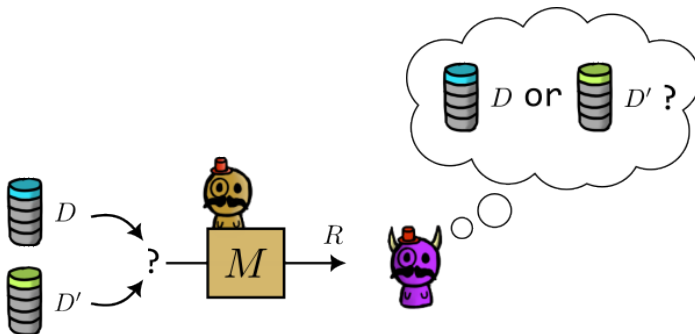
A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

There is no consensus on how small ϵ should be. “Roughly”:

- $\epsilon < 0.1$ is high privacy ($e^{0.1} \approx 1.1$)
- $0.1 < \epsilon < 1$ is good privacy ($e^1 \approx 2.7$)
- $\epsilon > 5$ starts getting too big ($e^5 \approx 148$)
- $\epsilon > 100\,000$ is crazy... yet some works use this

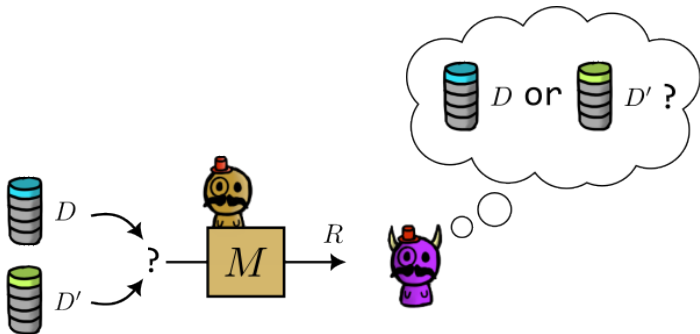
DP interpretation as a game



Assumption: The adversary has narrowed down to two databases (D and D') which only differ in one entry. The adversary knows M .

- These assumptions are many times unrealistic, but we want privacy even in this **worst-case scenario**

DP interpretation as a game

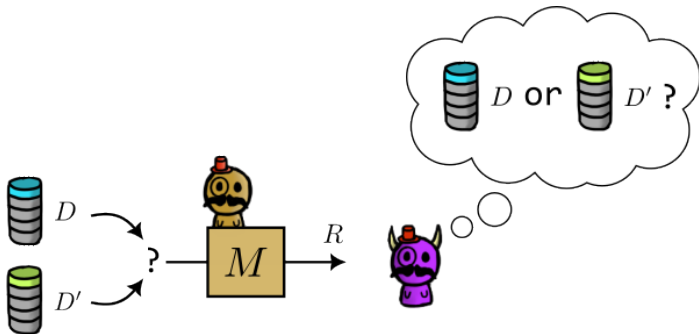


Guessing procedure: The adversary computes

- $p_D = \Pr[M(D) = R]$
- $p_{D'} = \Pr[M(D') = R]$

and guess *optimally*, i.e., guess D if $p_D > p_{D'}$ and D' otherwise.

DP interpretation as a game



Guessing procedure: The adversary computes

- $p_D = \Pr[M(D) = R]$
- $p_{D'} = \Pr[M(D') = R]$

and guess *optimally*, i.e., guess D if $p_D > p_{D'}$ and D' otherwise.

Claim: If M is ϵ -DP, the adversary's probability of error (i.e., wrong guess) is

$$\frac{1}{e^\epsilon + 1} \leq p_{\text{error}} \leq 0.5$$

DP interpretation as a game


Claim: If M is ϵ -DP, the adversary's probability of error (i.e., wrong guess) is

$$\frac{1}{e^\epsilon + 1} \leq p_{\text{error}} \leq 0.5$$

DP interpretation as a game

Claim: If M is ϵ -DP, the adversary's probability of error (i.e., wrong guess) is

$$\frac{1}{e^\epsilon + 1} \leq p_{\text{error}} \leq 0.5$$

ϵ	p_{err} range	Privacy
0	$0.5 \leq p_{\text{err}} \leq 0.5$	Perfect!
0.1	$0.47 \leq p_{\text{err}} \leq 0.5$	Very high
1	$0.26 \leq p_{\text{err}} \leq 0.5$	OK?
5	$0.006 \leq p_{\text{err}} \leq 0.5$	Bad
10	$0.00004 \leq p_{\text{err}} \leq 0.5$	Meaningless?
100 000	$10^{-43430} \leq p_{\text{err}} \leq 0.5$	

ϵ -DP elaboration: small ϵ

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

Another definition (not ϵ -DP)

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot (1 + \epsilon)$$

ϵ -DP elaboration: small ϵ

ϵ -DP

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon$$

Another definition (not ϵ -DP)

A mechanism $M : X \rightarrow Y$ is ϵ -differentially private (ϵ -DP) if for any two neighboring databases $D : X$ and $D' : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot (1 + \epsilon)$$

NOTE: for small ϵ , $e^\epsilon \approx 1 + \epsilon$ by Tolor series

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

Outline

- ① The Dinur-Nissim reconstruction attack
- ② The intuition behind differential privacy
- ③ A formal definition of differential privacy
- ④ Properties of the ϵ -DP definition**
- ⑤ Perturbation mechanisms
- ⑥ More topics on differential privacy

Safety against post-processing

Theorem: Suppose mechanism $M : X \rightarrow Y$ is ϵ -differentially private. Then, for any mechanism $A : Y \rightarrow Z$, we have that $A \circ M : X \rightarrow Z$ is also ϵ -differentially private.

Safety against post-processing

Theorem: Suppose mechanism $M : X \rightarrow Y$ is ϵ -differentially private. Then, for any mechanism $A : Y \rightarrow Z$, we have that $A \circ M : X \rightarrow Z$ is also ϵ -differentially private.

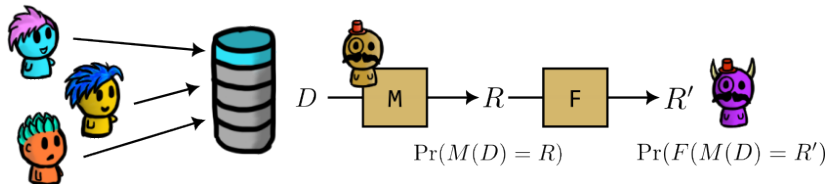
$$\begin{aligned} \forall T \subseteq Y, \quad \Pr[M(D) \in T] &\leq \Pr[M(D') \in T] \cdot e^\epsilon \Rightarrow \\ \forall T' \subseteq Z, \quad \Pr[F(M(D)) \in T'] &\leq \Pr[F(M(D')) \in T'] \cdot e^\epsilon \end{aligned}$$

Safety against post-processing

Theorem: Suppose mechanism $M : X \rightarrow Y$ is ϵ -differentially private. Then, for any mechanism $A : Y \rightarrow Z$, we have that $A \circ M : X \rightarrow Z$ is also ϵ -differentially private.

$$\forall T \subseteq Y, \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon \Rightarrow$$

$$\forall T' \subseteq Z, \Pr[F(M(D)) \in T'] \leq \Pr[F(M(D')) \in T'] \cdot e^\epsilon$$

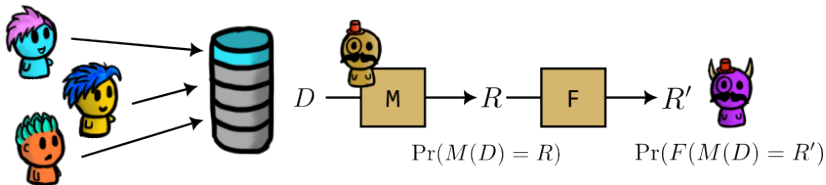


Safety against post-processing

Theorem: Suppose mechanism $M : X \rightarrow Y$ is ϵ -differentially private. Then, for any mechanism $A : Y \rightarrow Z$, we have that $A \circ M : X \rightarrow Z$ is also ϵ -differentially private.

$$\forall T \subseteq Y, \Pr[M(D) \in T] \leq \Pr[M(D') \in T] \cdot e^\epsilon \Rightarrow$$

$$\forall T' \subseteq Z, \Pr[F(M(D)) \in T'] \leq \Pr[F(M(D')) \in T'] \cdot e^\epsilon$$



\Rightarrow Once the data is privatized, it can't be “un-privatized”

Compositional privacy

Theorem: Given

- $M_1 : X \rightarrow Y_1$ being ϵ_1 -DP, and
- $M_2 : X \rightarrow Y_2$ being ϵ_2 -DP.

We define a new mechanism $M : X \rightarrow (Y_1, Y_2)$ as $M(X) = (M_1(X), M_2(X))$. Then M is $(\epsilon_1 + \epsilon_2)$ -DP.

Compositional privacy

Theorem: Given

- $M_1 : X \rightarrow Y_1$ being ϵ_1 -DP, and
- $M_2 : X \rightarrow Y_2$ being ϵ_2 -DP.

We define a new mechanism $M : X \rightarrow (Y_1, Y_2)$ as $M(X) = (M_1(X), M_2(X))$. Then M is $(\epsilon_1 + \epsilon_2)$ -DP.

This has a gossip analogy:

- If A tells you something (potentially with noise),
- and then B tells you some other things (again, with noise).

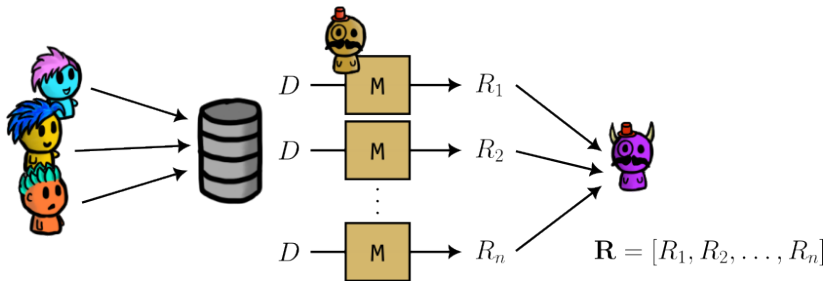
At the end of the day you might have learned more information by combining them together.

Privacy on sequential composition

If we run mechanisms with $\epsilon_1, \epsilon_2, \dots, \epsilon_n$, publishing all results provides $(\sum_{i=1}^n \epsilon_i)$ -DP

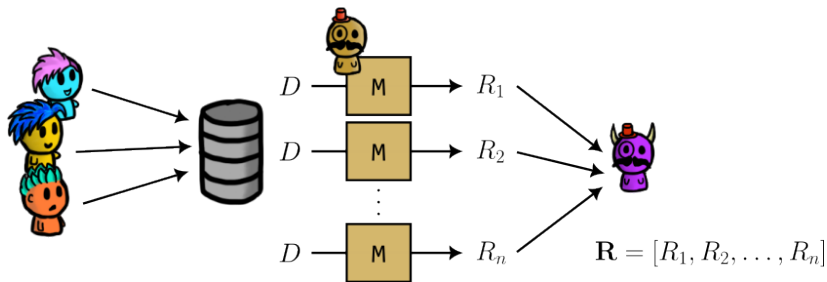
Privacy on sequential composition

If we run mechanisms with $\epsilon_1, \epsilon_2, \dots, \epsilon_n$, publishing all results provides $(\sum_{i=1}^n \epsilon_i)$ -DP



Privacy on sequential composition

If we run mechanisms with $\epsilon_1, \epsilon_2, \dots, \epsilon_n$, publishing all results provides $(\sum_{i=1}^n \epsilon_i)$ -DP



Here, we have

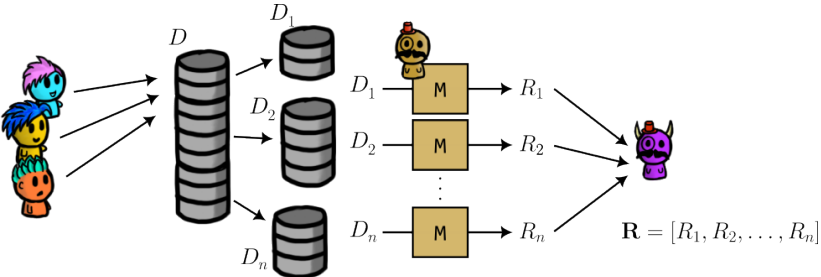
$$\Pr([M_1(D), M_2(D), \dots, M_n(D)] = \mathbf{R}) \leq \Pr([M_1(D'), M_2(D'), \dots, M_n(D')] = \mathbf{R}) \cdot e^{\sum_{i=1}^n \epsilon_i}$$

Privacy on parallel composition

If we run mechanisms with $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ **over disjoint subsets**, publishing all results provides $(\max_i \epsilon_i)$ -DP

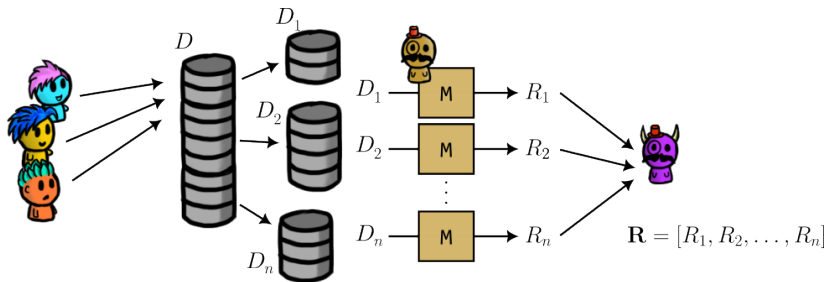
Privacy on parallel composition

If we run mechanisms with $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ **over disjoint subsets**, publishing all results provides $(\max_i \epsilon_i)$ -DP



Privacy on parallel composition

If we run mechanisms with $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ **over disjoint subsets**, publishing all results provides $(\max_i \epsilon_i)$ -DP



Here, we have

$$\Pr([M_1(D_1), M_2(D_2), \dots, M_n(D_n)] = \mathbf{R}) \leq \Pr([M_1(D'_1), M_2(D_2), \dots, M_n(D_n)] = \mathbf{R}) \cdot e^{\max_i \epsilon_i}$$

Group privacy

Theorem: Suppose mechanism $M : X \rightarrow Y$ is ϵ -differentially private. Suppose D_1 and D_2 are two datasets which differ in exactly k positions. Then:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{k\epsilon} \Pr[M(D_2) \in T]$$

Group privacy

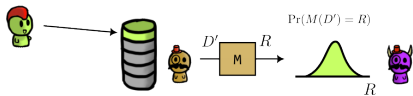
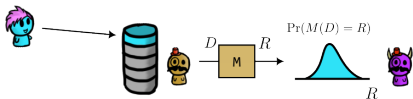
Theorem: Suppose mechanism $M : X \rightarrow Y$ is ϵ -differentially private. Suppose D_1 and D_2 are two datasets which differ in exactly k positions. Then:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{k\epsilon} \Pr[M(D_2) \in T]$$

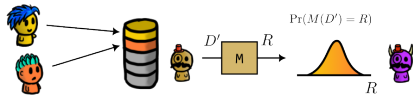
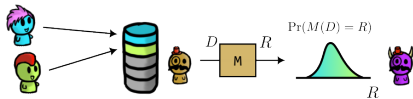
If you need to hide the “effect” if a whole group, you need to prepare a larger privacy budget.

Group privacy

If this is ϵ -DP:



Then this is 2ϵ -DP:



Outline

- ① The Dinur-Nissim reconstruction attack
- ② The intuition behind differential privacy
- ③ A formal definition of differential privacy
- ④ Properties of the ϵ -DP definition
- ⑤ Perturbation mechanisms**
- ⑥ More topics on differential privacy

Sensitivity

Q: How much noise to add?

Sensitivity

Q: How much noise to add? ← Sensitivity is a measurement

Sensitivity

Q: How much noise to add? \longleftarrow Sensitivity is a measurement

Definition: given a query processing function $f : X \rightarrow \mathbb{R}^k$, the ℓ_1 -sensitivity of f is defined as:

$$\Delta_1^f = \max_{D_1 \sim D_2} \|f(D_1) - f(D_2)\|_1 \quad \text{where } D_1, D_2 \in X$$

Sensitivity

Q: How much noise to add? \longleftarrow Sensitivity is a measurement

Definition: given a query processing function $f : X \rightarrow \mathbb{R}^k$, the ℓ_1 -sensitivity of f is defined as:

$$\Delta_1^f = \max_{D_1 \sim D_2} \|f(D_1) - f(D_2)\|_1 \quad \text{where } D_1, D_2 \in X$$

NOTE 1: The range of f is k -dimensional

Sensitivity

Q: How much noise to add? \longleftarrow Sensitivity is a measurement

Definition: given a query processing function $f : X \rightarrow \mathbb{R}^k$, the ℓ_1 -sensitivity of f is defined as:

$$\Delta_1^f = \max_{D_1 \sim D_2} \|f(D_1) - f(D_2)\|_1 \quad \text{where } D_1, D_2 \in X$$

NOTE 1: The range of f is k -dimensional

NOTE 2: ℓ_1 -sensitivity is $\|\vec{x}_1 - \vec{x}_2\|_1 = \sum_i |\vec{x}_1[i] - \vec{x}_2[i]|$

Sensitivity w/ one pair of neighboring databases

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50
-

Algorithm: You are allowed to make a query that returns the average score of this course.

Q: What is the ℓ_1 -sensitivity here?

Sensitivity w/ one pair of neighboring databases

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50
-

Algorithm: You are allowed to make a query that returns the average score of this course.

Q: What is the ℓ_1 -sensitivity here?

A: $|\text{Avg}(D_1) - \text{Avg}(D_2)| = 1.33$

Sensitivity w/ more database candidates

Q: What if we don't know the scores?

Suppose we only know that each student's score $\in [0 - 100]$, and

- (in bounded DP): there are 30 students enrolled
- (in unbounded DP): there are 29 or 30 students enrolled

Algorithm: You are allowed to make a query that returns the average score of this course.

Q: What is the ℓ_1 -sensitivity here?

Sensitivity w/ more database candidates - bounded

Suppose we only know that each student's score $\in [0 - 100]$, and there are 30 students enrolled in the course.

Algorithm: You are allowed to make a query that returns the average score of this course.

$$\begin{aligned}
 \ell_1 &= \max\left(\left|\frac{\sum_{29 \text{ students}} + k_1}{30} - \frac{\sum_{29 \text{ students}} + k_2}{30}\right|\right) \\
 &= \frac{1}{30} \max(|k_1 - k_2|) \\
 &= \frac{1}{30} \times 100 \quad \leftrightarrow (k_1 = 0 \wedge k_2 = 100) \vee (k_1 = 100 \wedge k_2 = 0) \\
 &= \frac{10}{3}
 \end{aligned}$$

Sensitivity w/ more database candidates - unbounded

Suppose we only know that each student's score $\in [0 - 100]$, and there are either 29 or 30 students enrolled in the course.

Algorithm: You are allowed to make a query that returns the average score of this course.

$$\begin{aligned}
 \ell_1 &= \max\left(\left|\frac{\sum_{29 \text{ students}}}{29} - \frac{\sum_{29 \text{ students} + k}}{30}\right|\right) \\
 &= \max\left(\left|\frac{\sum_{29 \text{ students}}}{29 \times 30} - \frac{k}{30}\right|\right) \\
 &\xrightarrow{\text{case1}} \max\left(\frac{\sum_{29 \text{ students}}}{29 \times 30}\right) - \min\left(\frac{k}{30}\right) \\
 &\xrightarrow{\text{case2}} \max\left(\frac{k}{30}\right) - \min\left(\frac{\sum_{29 \text{ students}}}{29 \times 30}\right) \\
 &= \frac{10}{3} \text{ for both cases}
 \end{aligned}$$

Laplace distribution

Lap(μ, b) is defined as:

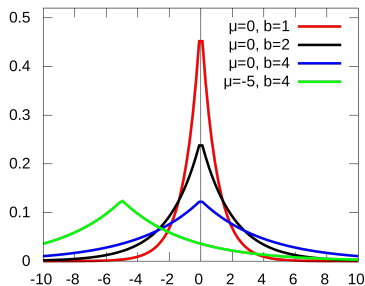
$$\Pr[x = v] = \frac{1}{2b} \exp\left(\frac{-|v - \mu|}{b}\right)$$

Laplace distribution

$\text{Lap}(\mu, b)$ is defined as:

$$\Pr[x = v] = \frac{1}{2b} \exp\left(\frac{-|v - \mu|}{b}\right)$$

- Usually, for DP, we set $\mu = 0$, so you may see $\text{Lap}(b)$ which is essentially $\text{Lap}(0, b)$
- $\text{Lap}(\mu, b)$ has variance $\sigma^2 = 2b^2$
- As b increases, the distribution becomes more flat



Laplace mechanism

Definition: Let $f : X \rightarrow \mathbb{R}^k$ is the function that calculates the “true” value of a query. The Laplace mechanism is defined as:

$$M(D) = f(D) + (Y_1, Y_2, \dots, Y_k)$$

where Y_i are independent and identically distributed (i.i.d) random variables sampled from $\text{Lap}(\frac{\Delta_f}{\epsilon})$

Laplace mechanism

Definition: Let $f : X \rightarrow \mathbb{R}^k$ is the function that calculates the “true” value of a query. The Laplace mechanism is defined as:

$$M(D) = f(D) + (Y_1, Y_2, \dots, Y_k)$$

where Y_i are independent and identically distributed (i.i.d) random variables sampled from $\text{Lap}(\frac{\Delta_f}{\epsilon})$

In our CS458 example:

let's take $\epsilon = 0.1$, and together with $\Delta = 1.33$, we have

$$M(D) = f(D) + \text{Lap}(13.3)$$

Laplace mechanism

Definition: Let $f : X \rightarrow \mathbb{R}^k$ is the function that calculates the “true” value of a query. The Laplace mechanism is defined as:

$$M(D) = f(D) + (Y_1, Y_2, \dots, Y_k)$$

where Y_i are independent and identically distributed (i.i.d) random variables sampled from $\text{Lap}(\frac{\Delta_f}{\epsilon})$

In our CS458 example:

let's take $\epsilon = 0.1$, and together with $\Delta = 1.33$, we have

$$M(D) = f(D) + \text{Lap}(13.3)$$

Demo time (average-demo.py)

Does the Laplace mechanism work in our example?

Let's first update the PDF by replacing $b = \frac{\Delta}{\epsilon}$:

$$\Pr[x = v] = \frac{\epsilon}{2\Delta} \exp\left(\frac{-\epsilon|v - \mu|}{\Delta}\right)$$

For D_1 , $\mu = 50$,

$$\Pr_1[x = 51.33] = \frac{\epsilon}{2\Delta} \exp\left(\frac{-\epsilon|51.33 - 50|}{\Delta}\right) = C \times e^{-0.1}$$

For D_2 , $\mu = 51.33$,

$$\Pr_2[x = 51.33] = \frac{\epsilon}{2\Delta} \exp\left(\frac{-\epsilon|51.33 - 51.33|}{\Delta}\right) = C \times e^{-0.075}$$

$$\frac{\Pr_2[x = 51.33]}{\Pr_1[x = 51.33]} = \frac{C \times e^{-0.075}}{C \times e^{-0.1}} = e^{0.025} \approx 1.025$$

The Laplace mechanism is ϵ -DP

Proof:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \rightarrow \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

The Laplace mechanism is ϵ -DP

Proof:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \rightarrow \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

$$\frac{\Pr[M(D_1) = z]}{\Pr[M(D_2) = z]} = \frac{\prod_{i=1}^k \frac{\epsilon}{2\Delta} \exp\left(\frac{-\epsilon}{\Delta} |f(D_1)[i] - z[i]|\right)}{\prod_{i=1}^k \frac{\epsilon}{2\Delta} \exp\left(\frac{-\epsilon}{\Delta} |f(D_2)[i] - z[i]|\right)}$$

The Laplace mechanism is ϵ -DP

Proof:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \rightarrow \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

$$\begin{aligned} \frac{\Pr[M(D_1) = z]}{\Pr[M(D_2) = z]} &= \frac{\prod_{i=1}^k \frac{\epsilon}{2\Delta} \exp\left(\frac{-\epsilon}{\Delta} |f(D_1)[i] - z[i]|\right)}{\prod_{i=1}^k \frac{\epsilon}{2\Delta} \exp\left(\frac{-\epsilon}{\Delta} |f(D_2)[i] - z[i]|\right)} \\ &= \frac{\prod_{i=1}^k \exp\left(\frac{-\epsilon}{\Delta} |f(D_1)[i] - z[i]|\right)}{\prod_{i=1}^k \exp\left(\frac{-\epsilon}{\Delta} |f(D_2)[i] - z[i]|\right)} \end{aligned}$$

The Laplace mechanism is ϵ -DP

Proof:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \rightarrow \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

$$\begin{aligned} \frac{\Pr[M(D_1) = z]}{\Pr[M(D_2) = z]} &= \frac{\prod_{i=1}^k \exp\left(\frac{-\epsilon}{\Delta} |f(D_1)[i] - z[i]| \right)}{\prod_{i=1}^k \exp\left(\frac{-\epsilon}{\Delta} |f(D_2)[i] - z[i]| \right)} \\ &= \prod_{i=1}^k \frac{\exp\left(\frac{-\epsilon}{\Delta} |f(D_1)[i] - z[i]| \right)}{\exp\left(\frac{-\epsilon}{\Delta} |f(D_2)[i] - z[i]| \right)} \end{aligned}$$

The Laplace mechanism is ϵ -DP

Proof:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \rightarrow \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

$$\begin{aligned} \frac{\Pr[M(D_1) = z]}{\Pr[M(D_2) = z]} &= \prod_{i=1}^k \frac{\exp\left(\frac{-\epsilon}{\Delta} |f(D_1)[i] - z[i]| \right)}{\exp\left(\frac{-\epsilon}{\Delta} |f(D_2)[i] - z[i]| \right)} \\ &= \prod_{i=1}^k \exp\left(\frac{\epsilon}{\Delta} (|f(D_1)[i] - z[i]| - |f(D_2)[i] - z[i]|)\right) \end{aligned}$$

The Laplace mechanism is ϵ -DP

Proof:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \rightarrow \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

$$\frac{\Pr[M(D_1) = z]}{\Pr[M(D_2) = z]} = \prod_{i=1}^k \exp\left(\frac{\epsilon}{\Delta} (|f(D_1)[i] - z[i]| - |f(D_2)[i] - z[i]|)\right)$$

$$\leq \prod_{i=1}^k \exp\left(\frac{\epsilon}{\Delta} |f(D_1)[i] - f(D_2)[i]|\right)$$

The Laplace mechanism is ϵ -DP

Proof:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \rightarrow \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

$$\begin{aligned} \frac{\Pr[M(D_1) = z]}{\Pr[M(D_2) = z]} &\leq \prod_{i=1}^k \exp\left(\frac{\epsilon}{\Delta} |f(D_1)[i] - f(D_2)[i]|\right) \\ &= \exp\left(\frac{\epsilon}{\Delta} \sum_{i=1}^k |f(D_1)[i] - f(D_2)[i]|\right) \end{aligned}$$

The Laplace mechanism is ϵ -DP

Proof:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \rightarrow \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

$$\frac{\Pr[M(D_1) = z]}{\Pr[M(D_2) = z]} \leq \exp\left(\frac{\epsilon}{\Delta} \sum_{i=1}^k |f(D_1)[i] - f(D_2)[i]|\right)$$

$$= \exp\left(\frac{\epsilon}{\Delta} \|f(D_1) - f(D_2)\|_1\right)$$

The Laplace mechanism is ϵ -DP

Proof:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \rightarrow \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

$$\begin{aligned} \frac{\Pr[M(D_1) = z]}{\Pr[M(D_2) = z]} &\leq \exp\left(\frac{\epsilon}{\Delta} \|f(D_1) - f(D_2)\|_1\right) \\ &\leq \exp\left(\frac{\epsilon}{\Delta} \Delta\right) \end{aligned}$$

The Laplace mechanism is ϵ -DP

Proof:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \rightarrow \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

$$\frac{\Pr[M(D_1) = z]}{\Pr[M(D_2) = z]} \leq \exp(\epsilon)$$

Outline

- ① The Dinur-Nissim reconstruction attack
- ② The intuition behind differential privacy
- ③ A formal definition of differential privacy
- ④ Properties of the ϵ -DP definition
- ⑤ Perturbation mechanisms
- ⑥ More topics on differential privacy

Approximate differential privacy

Definition:

A mechanism $M : X \rightarrow Y$ is (ϵ, δ) -differentially private $((\epsilon, \delta)$ -DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^\epsilon \Pr[M(D_2) \in T] + \delta$$

Approximate differential privacy

Definition:

A mechanism $M : X \rightarrow Y$ is (ϵ, δ) -differentially private $((\epsilon, \delta)$ -DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^\epsilon \Pr[M(D_2) \in T] + \delta$$

Interpretation: The new privacy parameter, δ , represents a “failure probability” for the definition.

- With probability $1 - \delta$ we will get the same guarantee as pure differential privacy;
- With probability δ , we get no privacy guarantee at all.

Approximate differential privacy

Definition:

A mechanism $M : X \rightarrow Y$ is (ϵ, δ) -differentially private ((ϵ, δ) -DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^\epsilon \Pr[M(D_2) \in T] + \delta$$

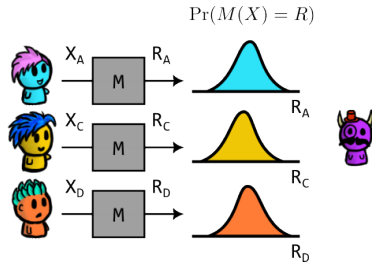
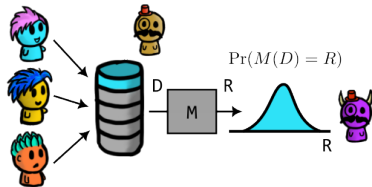
Interpretation: The new privacy parameter, δ , represents a “failure probability” for the definition.

- With probability $1 - \delta$ we will get the same guarantee as pure differential privacy;
- With probability δ , we get no privacy guarantee at all.

This definition allows us to add a much smaller noise.

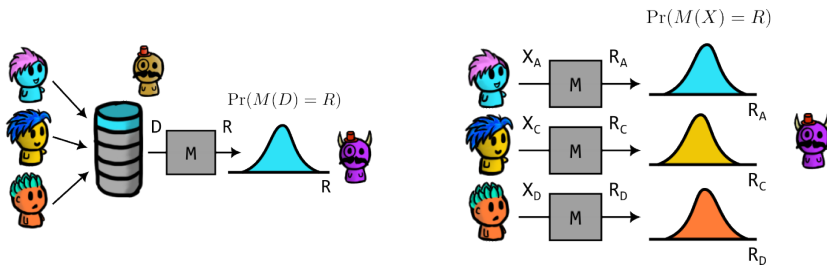
Local differential privacy (LDP)

Local differential privacy (LDP) is a model of differential privacy with the added restriction that **even if an adversary has access to the personal responses of an individual in the database**, that adversary will still be unable to learn too much about the user's personal data.



Local differential privacy (LDP)

Local differential privacy (LDP) is a model of differential privacy with the added restriction that **even if an adversary has access to the personal responses of an individual in the database**, that adversary will still be unable to learn too much about the user's personal data.



This eliminates the trust on the database [curator](#).

“Neighboring datasets” in LDP

In the local setting, usually the user has a value X , and providing ϵ -LDP means hiding whether the value was X or another value X' .

“Neighboring datasets” in LDP

In the local setting, usually the user has a value X , and providing ϵ -LDP means hiding whether the value was X or another value X' .

Definition M provides ϵ -LDP, if the following holds for all X, X' and outputs R :

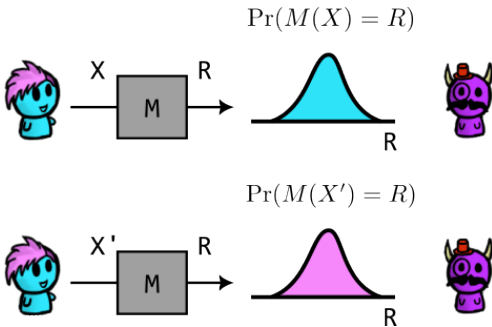
$$\Pr[M(X) = R] \leq \Pr[M(X') = R] \cdot e^\epsilon$$

“Neighboring datasets” in LDP

In the local setting, usually the user has a value X , and providing ϵ -LDP means hiding whether the value was X or another value X' .

Definition M provides ϵ -LDP, if the following holds for all X, X' and outputs R :

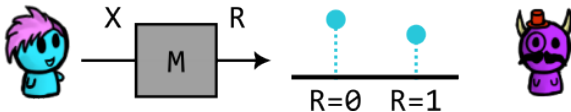
$$\Pr[M(X) = R] \leq \Pr[M(X') = R] \cdot e^\epsilon$$



Concrete use cases of LDP

Recap We are in the local model. Alice has a secret bit X , and reports another bit R .

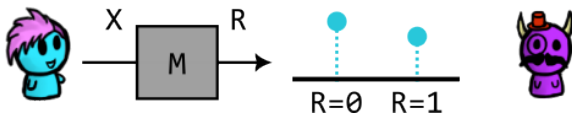
$$\Pr(M(X) = R)$$



Concrete use cases of LDP

Recap We are in the local model. Alice has a secret bit X , and reports another bit R .

$$\Pr(M(X) = R)$$

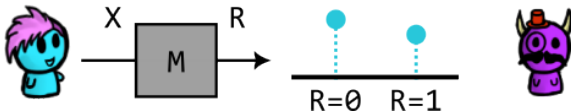


Q: Can you name some use cases?

Concrete use cases of LDP

Recap We are in the local model. Alice has a secret bit X , and reports another bit R .

$$\Pr(M(X) = R)$$



Q: Can you name some use cases?

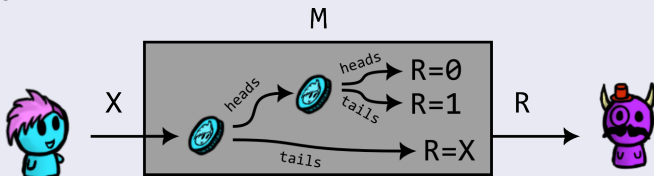
A:

- “Have you voted for party Y ?”
- “Have you tested positive for virus Z ?”
- “Have you cheated in exam W ?”

Randomized Response

Randomized Response

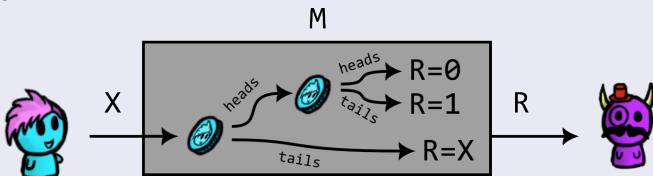
Given your true answer X , you report an answer R following this process:



Randomized Response

Randomized Response

Given your true answer X , you report an answer R following this process:



Q: What are these probabilities?

$$\Pr[R = 0 | X = 0]$$

$$\Pr[R = 1 | X = 0]$$

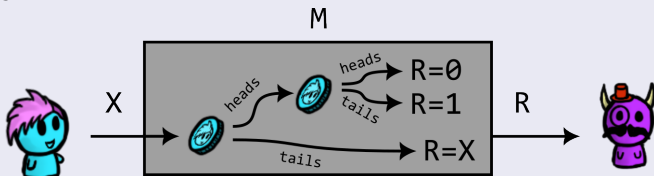
$$\Pr[R = 0 | X = 1]$$

$$\Pr[R = 1 | X = 1]$$

Randomized Response

Randomized Response

Given your true answer X , you report an answer R following this process:



Q: What are these probabilities?

$$\Pr[R = 0 | X = 0]$$

$$\Pr[R = 1 | X = 0]$$

$$\Pr[R = 0 | X = 1]$$

$$\Pr[R = 1 | X = 1]$$

A: The probabilities are:

$$\Pr[R = 0 | X = 0] = 0.75$$

$$\Pr[R = 1 | X = 0] = 0.25$$

$$\Pr[R = 0 | X = 1] = 0.25$$

$$\Pr[R = 1 | X = 1] = 0.75$$

Randomized Response: computing ϵ

Recall: LDP definition

A mechanism M is ϵ -LDP if, for all possible input pairs (X, X') and all possible outputs R ,

$$\Pr[R|X] \leq \Pr[R|X'] \cdot e^\epsilon \quad \Rightarrow \quad \frac{\Pr[R|X]}{\Pr[R|X']} \leq e^\epsilon$$

Randomized Response: computing ϵ

Recall: LDP definition

A mechanism M is ϵ -LDP if, for all possible input pairs (X, X') and all possible outputs R ,

$$\Pr[R|X] \leq \Pr[R|X'] \cdot e^\epsilon \quad \Rightarrow \quad \frac{\Pr[R|X]}{\Pr[R|X']} \leq e^\epsilon$$

A: We have this:

$$\Pr[R = 0|X = 0] = 0.75$$

$$\Pr[R = 1|X = 0] = 0.25$$

$$\Pr[R = 0|X = 1] = 0.25$$

$$\Pr[R = 1|X = 1] = 0.75$$

Randomized Response: computing ϵ

Recall: LDP definition

A mechanism M is ϵ -LDP if, for all possible input pairs (X, X') and all possible outputs R ,

$$\Pr[R|X] \leq \Pr[R|X'] \cdot e^\epsilon \quad \Rightarrow \quad \frac{\Pr[R|X]}{\Pr[R|X']} \leq e^\epsilon$$

A: We have this:

$$\Pr[R = 0|X = 0] = 0.75$$

$$\Pr[R = 1|X = 0] = 0.25$$

$$\Pr[R = 0|X = 1] = 0.25$$

$$\Pr[R = 1|X = 1] = 0.75$$

Q: How to calculate ϵ ?

Randomized Response: computing ϵ

Recall: LDP definition

A mechanism M is ϵ -LDP if, for all possible input pairs (X, X') and all possible outputs R ,

$$\Pr[R|X] \leq \Pr[R|X'] \cdot e^\epsilon \quad \Rightarrow \quad \frac{\Pr[R|X]}{\Pr[R|X']} \leq e^\epsilon$$

A: We have this:

$$\Pr[R = 0|X = 0] = 0.75$$

$$\Pr[R = 1|X = 0] = 0.25$$

$$\Pr[R = 0|X = 1] = 0.25$$

$$\Pr[R = 1|X = 1] = 0.75$$

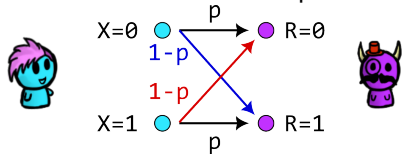
Q: How to calculate ϵ ?

A: $\epsilon = \ln 3 \approx 1.10$

This is because all $\frac{\Pr[R|X]}{\Pr[R|X']} \leq 3$

Statistical utility from randomized responses?

We draw randomized response as:



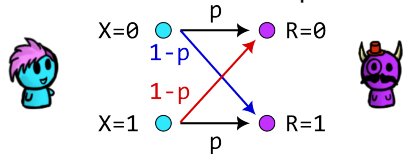
Additional facts:

Assume there are n users reporting values, and a fraction p_0 have $X = 0$, while a fraction $p_1 = 1 - p_0$ have $X = 1$.

Q: Given p_0 and p_1 , what is the probability that a response is $R = 1$? (or $\mathbb{E}[R]$)

Statistical utility from randomized responses?

We draw randomized response as:



Additional facts:

Assume there are n users reporting values, and a fraction p_0 have $X = 0$, while a fraction $p_1 = 1 - p_0$ have $X = 1$.

Q: Given p_0 and p_1 , what is the probability that a response is $R = 1$? (or $\mathbb{E}[R]$)

A: From the users that had $X = 0$, a fraction $1 - p$ of them will report $R = 1$. From the users that had $X = 1$, a fraction p will report $R = 1$. Therefore,

$$\mathbb{E}[R] = p_0 \cdot (1 - p) + p_1 \cdot p = 1 - p + (2p - 1) \cdot p_1$$

Statistical utility from randomized responses?

Q: Given p_0 and p_1 , what is the probability that a response is $R = 1$? (or $\mathbb{E}[R]$)

A: From the users that had $X = 0$, a fraction $1 - p$ of them will report $R = 1$. From the users that had $X = 1$, a fraction p will report $R = 1$. Therefore,

$$\mathbb{E}[R] = p_0 \cdot (1 - p) + p_1 \cdot p = 1 - p + (2p - 1) \cdot p_1$$

Q: Out of n responses, we received k 1s and $n - k$ 0s. How would you estimate the percentage of users that had $X = 1$?

Statistical utility from randomized responses?

Q: Given p_0 and p_1 , what is the probability that a response is $R = 1$? (or $\mathbb{E}[R]$)

A: From the users that had $X = 0$, a fraction $1 - p$ of them will report $R = 1$. From the users that had $X = 1$, a fraction p will report $R = 1$. Therefore,

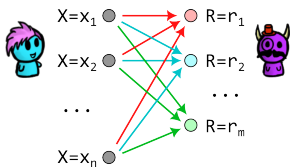
$$\mathbb{E}[R] = p_0 \cdot (1 - p) + p_1 \cdot p = 1 - p + (2p - 1) \cdot p_1$$

Q: Out of n responses, we received k 1s and $n - k$ 0s. How would you estimate the percentage of users that had $X = 1$?

A: $\mathbb{E}[R] \approx \bar{R} = \frac{k}{n} \Rightarrow \hat{p}_1 = \frac{\bar{R} - (1 - p)}{2p - 1}$

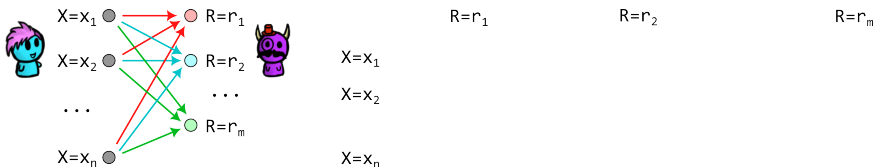
Computing ϵ for discrete mechanisms

Given a mechanism M with a discrete input space $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ and discrete output space $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$, you can compute ϵ this way:



Computing ϵ for discrete mechanisms

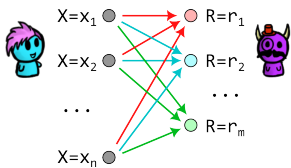
Given a mechanism M with a discrete input space $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ and discrete output space $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$, you can compute ϵ this way:



- 1 List inputs vs outputs

Computing ϵ for discrete mechanisms

Given a mechanism M with a discrete input space $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ and discrete output space $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$, you can compute ϵ this way:

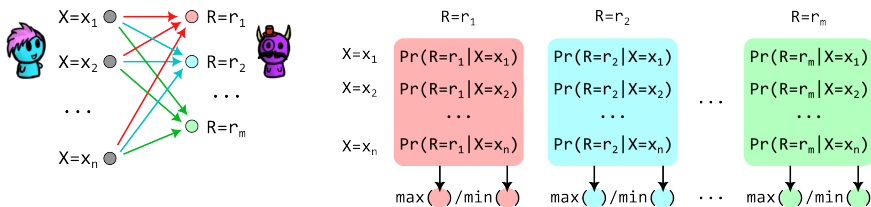


	$R=r_1$	$R=r_2$	$R=r_m$
$X=x_1$	$\Pr(R=r_1 X=x_1)$	$\Pr(R=r_2 X=x_1)$	$\Pr(R=r_m X=x_1)$
$X=x_2$	$\Pr(R=r_1 X=x_2)$	$\Pr(R=r_2 X=x_2)$	\dots $\Pr(R=r_m X=x_2)$
\dots	\dots	\dots	\dots
$X=x_n$	$\Pr(R=r_1 X=x_n)$	$\Pr(R=r_2 X=x_n)$	$\Pr(R=r_m X=x_n)$

- 1 List inputs vs outputs
- 2 Compute the probabilities

Computing ϵ for discrete mechanisms

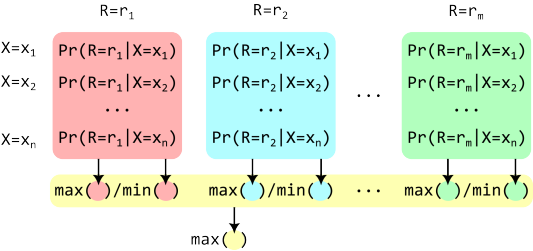
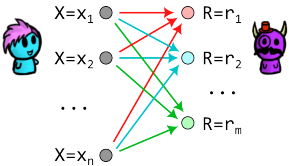
Given a mechanism M with a discrete input space $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ and discrete output space $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$, you can compute ϵ this way:



- 1 List inputs vs outputs
- 2 Compute the probabilities
- 3 Compute the max. ratio per possible value of output

Computing ϵ for discrete mechanisms

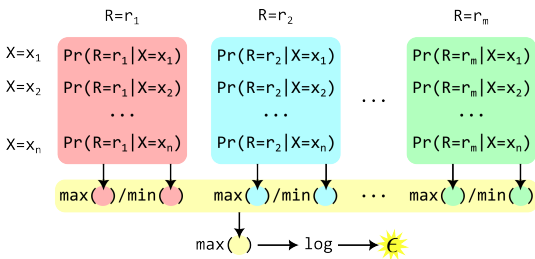
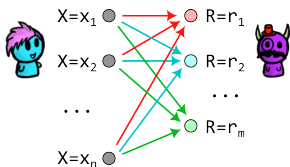
Given a mechanism M with a discrete input space $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ and discrete output space $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$, you can compute ϵ this way:



- 1 List inputs vs outputs
- 2 Compute the probabilities
- 3 Compute the max. ratio per possible value of output
- 4 Take the maximum of the ratios calculated above

Computing ϵ for discrete mechanisms

Given a mechanism M with a discrete input space $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ and discrete output space $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$, you can compute ϵ this way:



- 1 List inputs vs outputs
- 2 Compute the probabilities
- 3 Compute the max. ratio per possible value of output
- 4 Take the maximum of the ratios calculated above
- 5 ϵ is the natural log of that maximum