CS 458 / 658: Computer Security and Privacy

Module 6 - Data Security and Privacy

Part 3 - Differential privacy

Meng Xu *(University of Waterloo)*

Winter 2022

Outline

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.

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:

- Inference of the salary
- Census reconstruction attack

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:

- Inference of the salary
- Census reconstruction attack

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

Formalize our setup

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, $M : S \to R_S$. We'd like a mechnism that
  - gives statistically useful responses but
  - avoids leaking sensitive information about individuals.

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*.

This mechanism is called blatantly non-private.

Attack setup

We consider the database to be a collection of n records

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

where each record corresponds to one individual.

Attack setup

We consider the database to be a collection of n records

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

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 . . .			

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 $C(a_{\{*,1\}}, a_{\{*,2\}}, \ldots, a_{\{*,k-1\}})$ 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 $C(a_{\{*,1\}}, a_{\{*,2\}}, \ldots, a_{\{*,k-1\}})$ 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 $S = [0, 1, 1, 1, \dots, 0]$

Curator mechanism

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

$$R_S = A(S)$$

Curator mechanism

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

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

And subsequently add a random noise E to the true answer.

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 $|\Sigma_{i \in S} C[i] - R_S| > E$, rule out C.
- Any database candidate not ruled out (C) differs with the actual database (D) by $4E$ at max.

The inefficient attack 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

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

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

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

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 $\Theta(\alpha^2 n)$ 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

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.

An informal 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.

An informal 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: I don't care if the adversary can reconstruct the entire database or not. All I care is that the adversary learns (almost) nothing new about me even after seeing A and T, and regardless of what other datasets are available.

An informal 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: I don't care if the adversary can reconstruct the entire database or not. All I care is that 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

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

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

For all 5 responses, the chance of getting at least one 63 is
$1 - (1 - \frac{C_{29}^2}{C_{30}^3})^5 = 40.95\%!$

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

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, $M : S \to R_S$. We'd like a mechnism that
  - gives statistically useful responses but
  - avoids leaking sensitive information about individuals.

Neighboring databases

Two databases D_1 and D_2 are neighbouring if they agree except for a single entry.

Neighboring databases

Two databases D_1 and D_2 are neighbouring if they agree except for a single entry.

- *Unbounded DP*: D_1 and D_2 are neighboring if D_2 can be obtained from D_1 by adding or removing one element
- *Bounded DP*: D_1 and D_2 are neighboring if D_2 can be obtained from D_1 by replacing one element

ϵ-differential privacy

Idea: If the mechanism M behaves nearly identically for D_1 and D_2, then an attacker can't tell whether D_1 or D_2 was used (and hence can't learn much about the individual).

ϵ-differential privacy

Idea: If the mechanism M behaves nearly identically for D_1 and D_2, then an attacker can't tell whether D_1 or D_2 was used (and hence can't learn much about the individual).

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

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

ϵ-differential privacy

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

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

ϵ-differential privacy

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

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

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

ϵ-differential privacy

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

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

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,

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

ϵ-differential privacy

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

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

ϵ-differential privacy

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

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

Suppose we have:

- $\epsilon = 0.01$
- $\Pr[M(D_1) \in T] = 0.005$
- $\Pr[M(D_2) \in T] = 0.001$

- $\epsilon = 0.01$
- $\Pr[M(D_1) \in T] = 0.96$
- $\Pr[M(D_2) \in T] = 0.94$

ϵ-differential privacy

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

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq \epsilon \times \Pr[M(D_2) \in T]$$

ϵ-differential privacy

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

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq \epsilon \times \Pr[M(D_2) \in T]$$

It does not make sense for ϵ to be < 1 or too large.

ϵ-differential privacy

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

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq (1 + \epsilon) \Pr[M(D_2) \in T]$$

ϵ-differential privacy

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

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq (1 + \epsilon) \Pr[M(D_2) \in T]$$

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

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

Safety against post-processing

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

Safety against post-processing

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

Once the data is privatized, it can't be "un-privatized"

Compositional privacy

Theorem: Given

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

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

Compositional privacy

Theorem: Given

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

We define a new mechanism $M : X \to Y_1 \times Y_2$ 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.

Group privacy

Theorem: Suppose mechanism $M : X \to Y$ 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 $M : X \to Y$ 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.

Outline

Sensitivity

Q: How much noise to add?

Sensitivity

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

Sensitivity

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

Definition: given a query processing function $f : X \to \mathbb{R}^k$, 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 $f : X \to \mathbb{R}^k$, 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 $f : X \to \mathbb{R}^k$, 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.

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

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.

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

Laplace distribution

Lap(μ, b) is defined as:

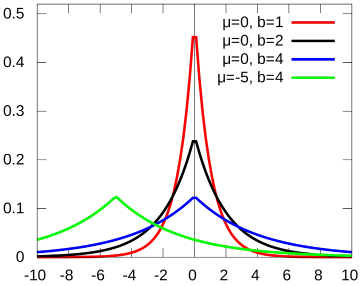
$$\Pr[x = v] = \frac{1}{2b} \exp\left(\frac{-|v - \mu|}{b}\right)$$

Laplace distribution

Lap(μ, 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 Lap(b) which is essentially Lap(0, b)
- Lap(μ, b) has variance $\sigma^2 = 2b^2$
- As b increases, the distribution becomes more flat



Laplace mechanism

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

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

where Y_i are independent and identically distributed (i.i.d) random variables sampled from Lap($\frac{\Delta_1^f}{\epsilon}$)

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:

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

where Y_i are independent and identically distributed (i.i.d) random variables sampled from $\text{Lap}(\frac{\Delta_1^f}{\epsilon})$

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 $f : X \to \mathbb{R}^k$ is the function that calculates the "true" value of a query. The Laplace mechanism is defined as:

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

where Y_i are independent and identically distributed (i.i.d) random variables sampled from $\text{Lap}(\frac{\Delta_1^f}{\epsilon})$

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 $f : X \to \mathbb{R}^k$ 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 $f : X \to \mathbb{R}^k$ be the function that calculates the "true" value
- Let $z \in \mathbb{R}^k$ being any potential response

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

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

The Laplace mechanism is ϵ-DP

Proof:

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

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

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

The Laplace mechanism is ϵ-DP

Proof:

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

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

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

The Laplace mechanism is ϵ-DP

Proof:

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

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

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

The Laplace mechanism is ϵ-DP

Proof:

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

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

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

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

Outline

Approximate differential privacy

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

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

Approximate differential privacy

Definition:
A mechanism $M : X \to Y$ 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.

Approximate differential privacy

Definition:
A mechanism $M : X \to Y$ 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

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

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.

Local differential privacy

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.

Example: Randomized response to a survey