数据隐私方法伦理和实践
Methodology, Ethics and Practice of Data Privacy

7.差分隐私理论和方法
The Algorithmic Foundations of Differential Privacy

张兰
中国科学技术大学 计算机学院
2020春季

“

Differential privacy addresses the paradox of learning nothing about an individual while learning useful information about a population.

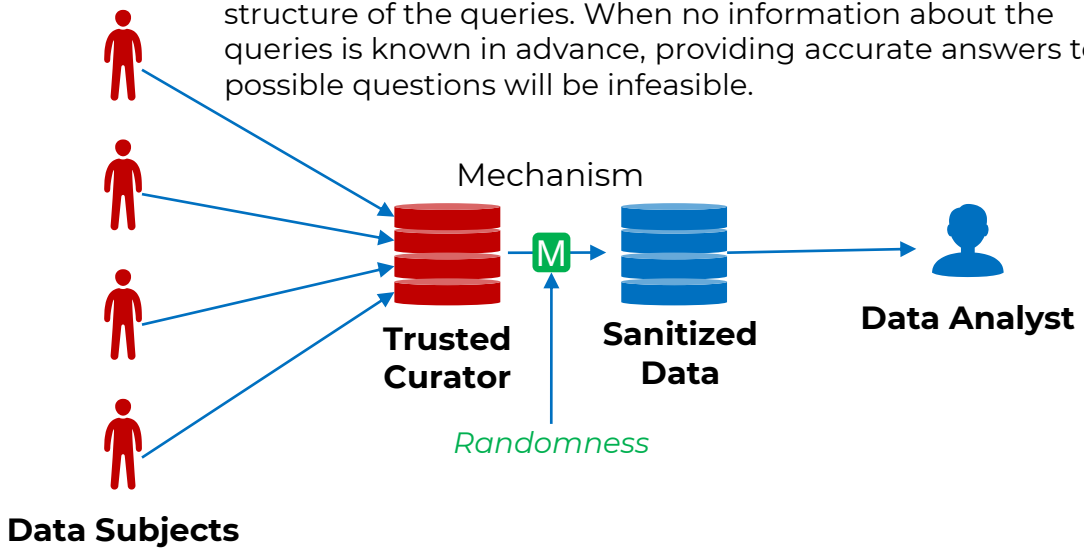
“Differential privacy” describes a promise, made by a data holder to a data subject: “You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.”

1. Basic Terms

Model of Computation

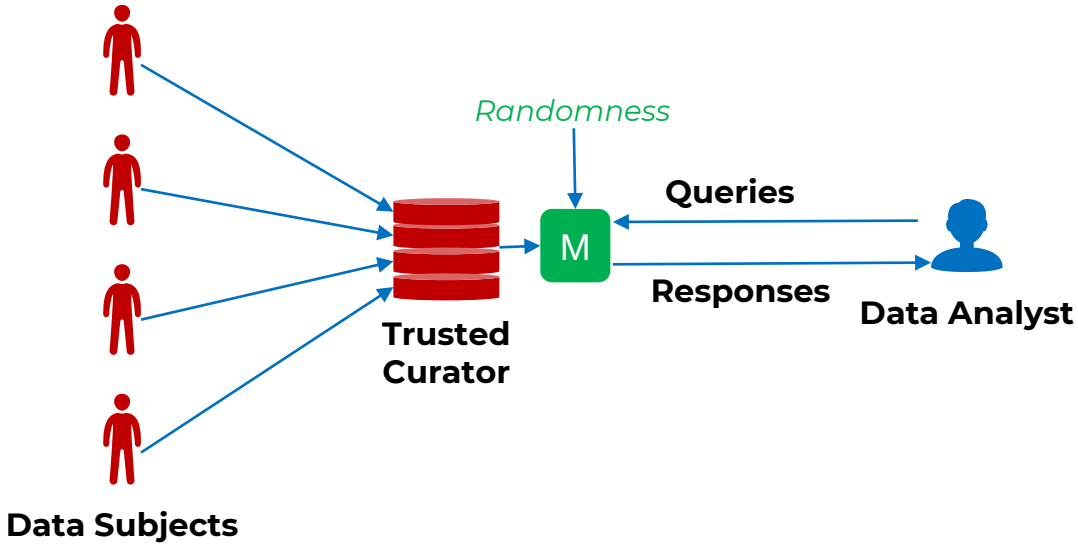
» Offline/ non-interactive

- When all the queries are known in advance, it should give the best accuracy, as it is able to correlate noise knowing the structure of the queries. When no information about the queries is known in advance, providing accurate answers to all possible questions will be infeasible.



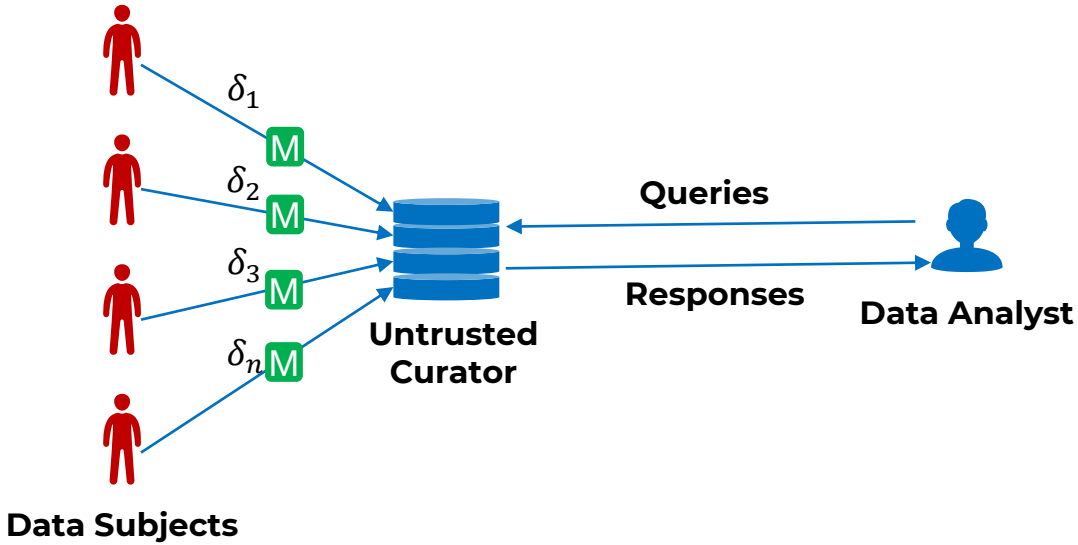
Model of Computation

- Online/interactive



Model of Computation

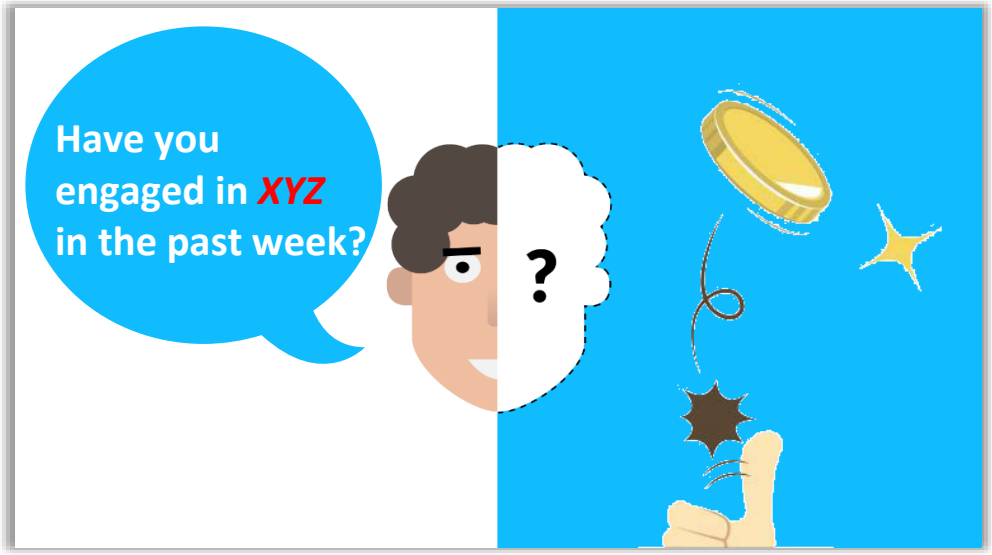
- Untrusted Curator/local model



Randomized Response

Let ***XYZ*** be such an activity.
Faced with the query .

- **Embarrassing**
- **illegal**



Investigator **Survey Respondent**

Randomized Response

- Each user u_i's private data is represented by a tuple t_i
- Each t_i contains 1 attributes A_1.
- Domain $\{0, 1\}$.
- Perturbed data $f(t_i) = t_i^*$

$$t_i^* = \begin{cases} t_i & w.p.\ p = \frac{e^\epsilon}{1+e^\epsilon} \\ \overline{t_i} & w.p.\ 1-p = \frac{1}{1+e^\epsilon} \end{cases}$$

- The probability of staying the same is p
- The probability of reversal is $1 - p$

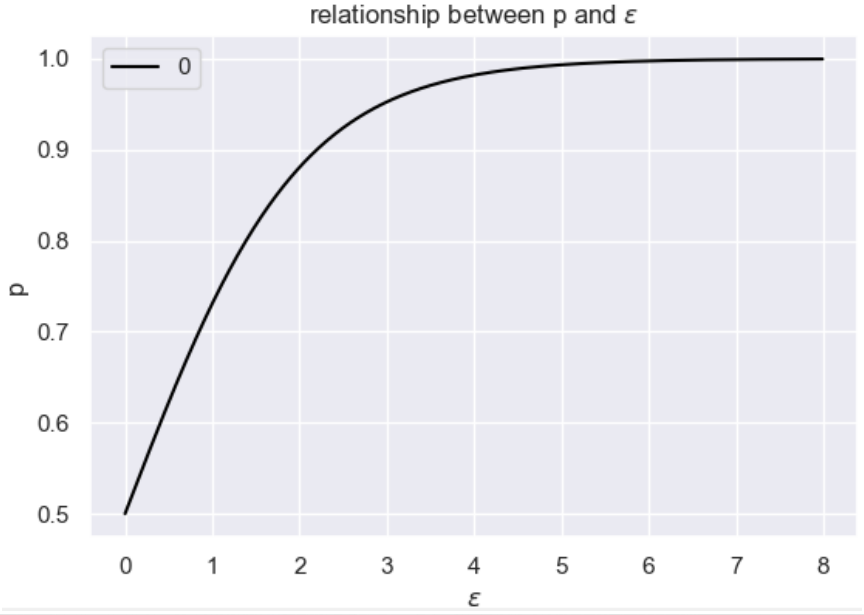
- The actual frequency of "yes" is f^T
- The frequency of "yes" the investigators saw is f^C

$$f^T \cdot p + (1 - f^T) \cdot (1 - p) = f^C$$

$$f^T = \frac{f^C - 1 + p}{2p - 1}$$

Randomized Response

$$t_i^* = \begin{cases} t_i & w.p.\ p = \frac{e^\epsilon}{1+e^\epsilon} \\ \overline{t_i} & w.p.\ 1-p = \frac{1}{1+e^\epsilon} \end{cases}$$



Randomized Algorithm

- A randomized algorithm M with domain A and discrete range B is associated with a mapping $M : A \rightarrow \Delta(B)$.

  Probability Simplex:

$$\Delta(B) = \left\{ x \in \mathbb{R}^{|B|} : x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{|B|} x_i = 1 \right\}$$

- On input $a \in A$, the algorithm M outputs $M(a) = b$ with probability $\left(M(a)\right)_b$ for each $b \in B$. The probability space is over the coin flips of the algorithm M.

“

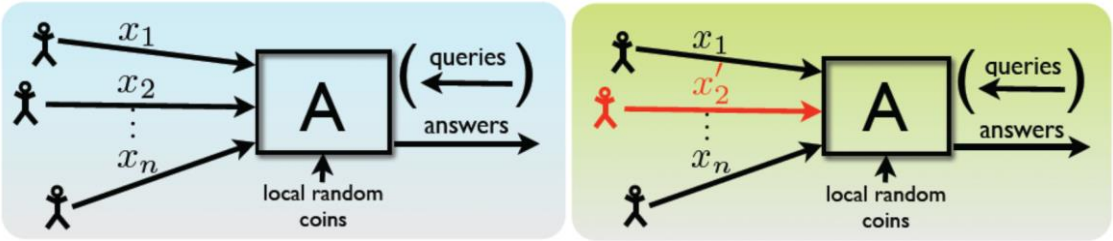
Whether your data is in the database or not,
I will get similar results.

Differential Privacy

Definition (Differential Privacy): A randomized algorithm M with domain $\mathbb{N}^{|X|}$ is (ϵ, δ)-differentially private if for all $S \subset Range(M)$ and for all $x, y \in \mathbb{N}^{|X|}$ such that $||x - y||_1 \leq 1$:

$$Pr[M(x) \in S] \leq exp(\epsilon) Pr[M(y) \in S] + \delta$$

where the probability space is over the coin flips of the mechanism M. If $\delta = 0$, we say that M is δ-differentially private.



x' is a neighbor of x if they differ in one row

From the released statistics, it is hard to tell which case it is.

Differential Privacy

Definition (Differential Privacy): A randomized algorithm M with domain $\mathbb{N}^{|X|}$ is (ϵ, δ)-differentially private if for all $S \subset Range(M)$ and for all $x, y \in \mathbb{N}^{|X|}$ such that $||x - y||_1 \leq 1$:

$$Pr[M(x) \in S] \leq exp(\epsilon) Pr[M(y) \in S] + \delta$$

where the probability space is over the coin flips of the mechanism M. If $\delta = 0$, we say that M is δ-differentially private.

Typically, we are interested in values of δ that are less than the inverse of any polynomial in the size of the database.

Values of δ on the order of $1/||x||_1$ are very dangerous:

"Just a Few" philosophy: provide privacy protection for "typical" members of a data set and compromise the privacy of "just a few" participants. It can be achieved by randomly selecting a subset of rows and releasing them in their entirety.

Differential Privacy

- Even δ is negligible, there are theoretical distinctions between (ε,0)- and (ε,δ)- differential privacy.
  - **(ε,0)-differential privacy**: for every run of the mechanism $M(x)$, the output observed is (almost) equally likely to be observed on every neighboring database, simultaneously.
  - **(ε,δ)- differential privacy**: given an output $\xi \sim M(x)$ it may be possible to find a database y such that ξ is much more likely to be produced on y than it is when the database is x.

    The **privacy loss** (divergence) incurred by observation ξ:

$$\mathcal{L}^{(\xi)}_{\mathcal{M}(x)\|\mathcal{M}(y)} = \ln\left(\frac{\Pr[\mathcal{M}(x)=\xi]}{\Pr[\mathcal{M}(y)=\xi]}\right)$$

(ε,δ)- differential privacy ensures that for all adjacent x, y, the absolute value of the privacy loss will be bounded by ε with probability at least 1-δ.

2. Basic Techniques

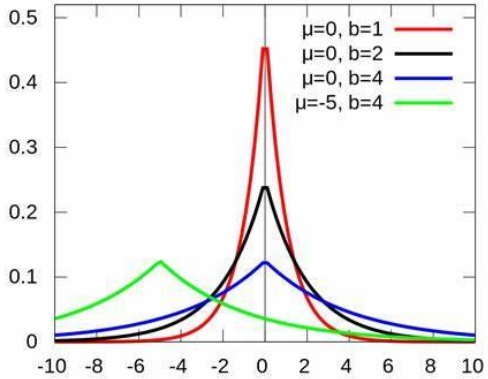
For real (vector) valued queries.

Laplace Mechanism

Definition (The Laplace Distribution). The Laplace Distribution (centered at 0) with scale b is the distribution with probability density function:

$$Lap(x|b) = \frac{1}{2b} exp(-\frac{|x|}{b})$$

The variance of this distribution is $\sigma^2 = 2b^2$. We will sometimes write $Lap(b)$ to denote the Laplace distribution with scale b, and will sometimes abuse notation and write $Lap(b)$ simply to denote a random variable $X \sim Lap(b)$.



Laplace Mechanism

Definition (l_1-sensitivity). The l_1-sensitivity of a function $f : \mathbb{N}^{|x|} \rightarrow \mathbb{R}^k$ is:

$$\Delta f = \max_{\substack{x,y \in \mathbb{N}^{|x|} \\ ||x-y||_1 = 1}} ||f(x) - f(y)||_1$$

The l_1 sensitivity of a function f captures the magnitude by which a single individual's data can change the function f in the worst case, and therefore, intuitively, the uncertainty in the response that we must introduce in order to hide the participation of a single individual. Indeed, we will formalize this intuition: the sensitivity of a function gives an upper bound on how much we must perturb its output to preserve privacy. One noise distribution naturally lends itself to differential privacy.

The larger Δf is, the greater the noise should be;

The smaller Δf is, the smaller the noise should be.

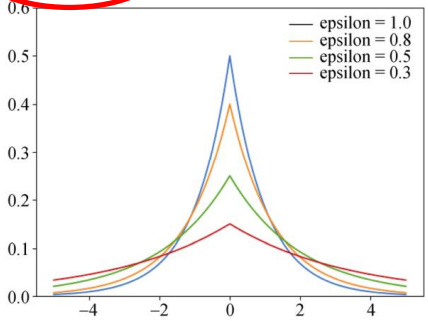
Laplace Mechanism

Definition (The Laplace Mechanism). Given any function $f : \mathbb{N}^{|x|} \to \mathbb{R}^k$, the Laplace mechanism is defined as:

$$\mathcal{M}_L(x, f(\cdot), \epsilon) = f(x) + (Y_1, ..., Y_k)$$

where Y_i are i.i.d. random variables drawn from $Lap(\frac{\triangle f}{\epsilon})$.

- k represents the dimension of the query.
- ε is an artificially defined parameter, the degree of privacy preserving in DP.
- Δf is the sensitivity of function f.



Theorem The Laplace mechanism preserves $(\epsilon, 0)$-differential privacy.

Laplace Mechanism

Proof. Let $x \in \mathbb{N}^{|\mathcal{X}|}$ and $y \in \mathbb{N}^{|\mathcal{X}|}$ be such that $\|x - y\|_1 \leq 1$, and let $f(\cdot)$ be some function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$. Let p_x denote the probability density function of $\mathcal{M}_L(x, f, \varepsilon)$, and let p_y denote the probability density function of $\mathcal{M}_L(y, f, \varepsilon)$. We compare the two at some arbitrary point $z \in \mathbb{R}^k$

$$
\begin{aligned}
\frac{p_x(z)}{p_y(z)} &= \prod_{i=1}^{k} \left(\frac{\exp(-\frac{\varepsilon|f(x)_i - z_i|}{\Delta f})}{\exp(-\frac{\varepsilon|f(y)_i - z_i|}{\Delta f})} \right) \\
&= \prod_{i=1}^{k} \exp\left(\frac{\varepsilon(|f(y)_i - z_i| - |f(x)_i - z_i|)}{\Delta f} \right) \\
&\leq \prod_{i=1}^{k} \exp\left(\frac{\varepsilon|f(x)_i - f(y)_i|}{\Delta f} \right) \\
&= \exp\left(\frac{\varepsilon \cdot \|f(x) - f(y)\|_1}{\Delta f} \right) \\
&\leq \exp(\varepsilon),
\end{aligned}
$$

where the first inequality follows from the triangle inequality, and the last follows from the definition of sensitivity and the fact that $\|x - y\|_1 \leq 1$. That $\frac{p_x(z)}{p_y(z)} \geq \exp(-\varepsilon)$ follows by symmetry. □

Accuracy of Laplace Mechanism

- Bounds on how accurate is the answer respect to the true answer

Theorem 3.8. Let $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, and let $y = \mathcal{M}_L(x, f(\cdot), \varepsilon)$. Then $\forall \delta \in (0, 1]$:

$$\Pr\left[\|f(x) - y\|_\infty \geq \ln\left(\frac{k}{\delta}\right) \cdot \left(\frac{\Delta f}{\varepsilon}\right)\right] \leq \delta$$

Proof. We have:

$$\begin{aligned}
\Pr\left[\|f(x) - y\|_\infty \geq \ln\left(\frac{k}{\delta}\right) \cdot \left(\frac{\Delta f}{\varepsilon}\right)\right] &= \Pr\left[\max_{i \in [k]} |Y_i| \geq \ln\left(\frac{k}{\delta}\right) \cdot \left(\frac{\Delta f}{\varepsilon}\right)\right] \\
&\leq k \cdot \Pr\left[|Y_i| \geq \ln\left(\frac{k}{\delta}\right) \cdot \left(\frac{\Delta f}{\varepsilon}\right)\right] \\
&= k \cdot \left(\frac{\delta}{k}\right) \\
&= \delta
\end{aligned}$$

Fact 3.7. If $Y \sim \text{Lap}(b)$, then:

$$\Pr[|Y| \geq t \cdot b] = \exp(-t).$$

Example

- First Names: calculate which first names, from a list of 10,000 potential names, were the most common ones.
  - This is a histogram query with $\Delta f = 1$
  - We can simultaneously calculate the frequency of all 10, 000 names with (1, 0)-differential privacy, and with probability 95%, no estimate will be off by more than an additive error of $\ln(10000/0.05) \approx 12.2$.
- Most Common Medical Condition: which condition is the most common in the medical histories of a set of respondents.
  - This is a histogram query with Δf=m (large!)
  - The m noisy counts themselves could not be released.

Differentially Private Selection

- The space of outcomes is discrete and the task is to produce a “best” answer.
- **Report Noisy Max**: add independently generated Laplace noise Lap($1/\varepsilon$) to each count and **return the index of the largest noisy count**.
- The Report Noisy Max algorithm is (ε, 0)-differentially private.

- In both examples the utility of the response is directly related to the noise values generated (on the same scale and in the same units).

For private selection from a discrete set of candidate outputs.
But adding noise directly to the computed quantity can completely destroy its utility.

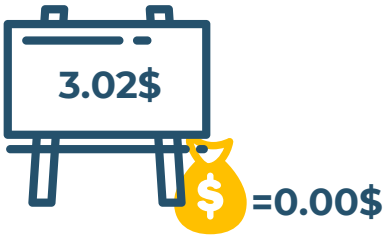
What is the optimal price?

A: 1.00$

B: 1.00$

C: 1.00$

D: 3.01$



Exponential Mechanism

- **Discrete-valued functions:** $f(x) \in \mathcal{R} = \{r_1, r_2, \ldots, r_k\}$
  - $\mathcal{R}$ is an arbitrary range
- **Utility function:**
  - Each output $r \in \mathcal{R}$ has a utility for database x, $u(x, r)$
- $\Delta u \equiv \max_{r \in \mathcal{R}} \max_{x,y:\|x-y\|_1 \leq 1} |u(x, r) - u(y, r)|.$
- **Privacy loss**

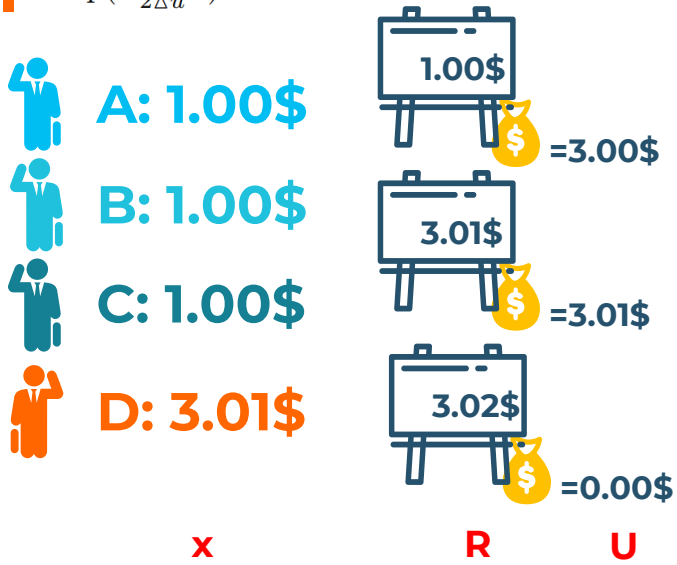
$$\ln\left(\frac{\exp(\varepsilon u(x, r)/\Delta u)}{\exp(\varepsilon u(y, r)/\Delta u)}\right) = \varepsilon[u(x, r) - u(y, r)]/\Delta u) \leq \varepsilon.$$

Output r with probability $\propto e^{\frac{\varepsilon u(x,r)}{2\Delta u}}$

Make high utility outputs exponentially more likely at a rate that depends on the sensitivity of $u(x, r)$

Exponential Mechanism

Definition (The Exponential Mechanism). The exponential mechanism $M_E(x, u, R)$ selects and outputs an element r ∈ R with probability proportional to $\exp(\frac{\epsilon u(x,y)}{2\Delta u})$.



It is important that the range of *potential* prices is independent of the actual bids.

Exponential Mechanism

Definition (The Exponential Mechanism). The exponential mechanism $M_E(x, u, R)$ selects and outputs an element r $\in$ R with probability proportional to $\exp(\frac{\epsilon u(x,y)}{2\triangle u})$.

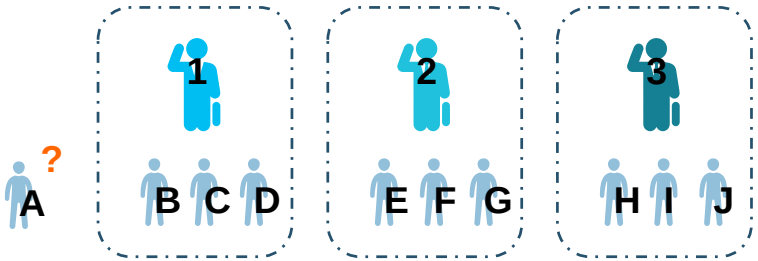


Table ballot dataset and probability distribution

	Total Votes	$\epsilon = 5$	$\epsilon = 0.5$	$\epsilon = 0$
1	4	0.9	0.39	0.33
2	3	0.045	0.3	0.33
3	3	0.045	0.3	0.33

Proof?

Exponential Mechanism

Theorem The exponential mechanism preserves $(\varepsilon, 0)$-differential privacy.

Proof. For clarity, we assume the range $\mathcal{R}$ of the exponential mechanism is finite, but this is not necessary. As in all differential privacy proofs, we consider the ratio of the probability that an instantiation of the exponential mechanism outputs some element $r \in \mathcal{R}$ on two neighboring databases $x \in \mathbb{N}^{|\mathcal{X}|}$ and $y \in \mathbb{N}^{|\mathcal{X}|}$ (i.e., $\|x - y\|_1 \leq 1$).

Exponential Mechanism

$$\frac{\Pr[\mathcal{M}_E(x,u,\mathcal{R})=r]}{\Pr[\mathcal{M}_E(y,u,\mathcal{R})=r]} = \frac{\left(\frac{\exp(\frac{\varepsilon u(x,r)}{2\Delta u})}{\sum_{r'\in\mathcal{R}}\exp(\frac{\varepsilon u(x,r')}{2\Delta u})}\right)}{\left(\frac{\exp(\frac{\varepsilon u(y,r)}{2\Delta u})}{\sum_{r'\in\mathcal{R}}\exp(\frac{\varepsilon u(y,r')}{2\Delta u})}\right)}$$

$$= \left(\frac{\exp(\frac{\varepsilon u(x,r)}{2\Delta u})}{\exp(\frac{\varepsilon u(y,r)}{2\Delta u})}\right)\cdot\left(\frac{\sum_{r'\in\mathcal{R}}\exp(\frac{\varepsilon u(y,r')}{2\Delta u})}{\sum_{r'\in\mathcal{R}}\exp(\frac{\varepsilon u(x,r')}{2\Delta u})}\right)$$

$$= \exp\left(\frac{\varepsilon(u(x,r')-u(y,r'))}{2\Delta u}\right) \cdot\left(\frac{\sum_{r'\in\mathcal{R}}\exp(\frac{\varepsilon u(y,r')}{2\Delta u})}{\sum_{r'\in\mathcal{R}}\exp(\frac{\varepsilon u(x,r')}{2\Delta u})}\right)$$

$$\leq \exp\left(\frac{\varepsilon}{2}\right)\cdot\exp\left(\frac{\varepsilon}{2}\right)\cdot\left(\frac{\sum_{r'\in\mathcal{R}}\exp(\frac{\varepsilon u(x,r')}{2\Delta u})}{\sum_{r'\in\mathcal{R}}\exp(\frac{\varepsilon u(x,r')}{2\Delta u})}\right)$$

$$= \exp(\varepsilon).$$

Similarly, $\frac{\Pr[\mathcal{M}_E(y,u)=r]}{\Pr[\mathcal{M}_E(x,u)=r]} \geq \exp(-\varepsilon)$ by symmetry. □

Accuracy of Exponential Mechanism

- The exponential mechanism can often give strong utility guarantees.
- How good is the output?
  - It will be highly unlikely that the returned element r has a utility score that is inferior to $OPT_u(x)$ by more than an additive factor of $O((\Delta u/\varepsilon)\log|\mathrm{R}|)$.

$$\Pr\left[u(\mathcal{M}_E(x,u,\mathcal{R})) \leq \mathrm{OPT}_u(x) - \frac{2\Delta u}{\varepsilon}\left(\ln\left(\frac{|\mathcal{R}|}{|\mathcal{R}_{\mathrm{OPT}}|}\right)+t\right)\right] \leq e^{-t}$$

$$|\mathcal{R}_{\mathrm{OPT}}| \geq 1,$$

$$\Pr\left[u(\mathcal{M}_E(x,u,\mathcal{R})) \leq \mathrm{OPT}_u(x) - \frac{2\Delta u}{\varepsilon}\left(\ln\left(|\mathcal{R}|\right)+t\right)\right] \leq e^{-t}$$

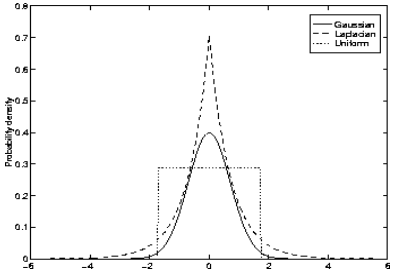
Example

- Best of Two: determining which of exactly two medical conditions A and B is more common.
  - Truth: 0 for A, c>0 for B
  - Utility: tied to the actual counts
  - Δu=1, $|R| = 2$
  - The probability of observing (wrong) outcome A is at most

$$2e^{-c(\varepsilon/(2\Delta u))} = 2e^{-c\varepsilon/2}$$

Laplace versus Gauss

- An alternative to adding Laplacian noise is to add Gaussian noise.
  - Rather than scaling the noise to the ℓ_1 sensitivity Δf, we instead scale to the ℓ_2 sensitivity.
  - Add zero-mean Gaussian noise with variance b in each of the k coordinates.
  - Gaussian noise is a common noise.
  - Sum of two Gaussians is a Gaussian



Theorem 3.22. Let $\varepsilon \in (0,1)$ be arbitrary. For $c^2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geq c\Delta_2(f)/\varepsilon$ is (ε, δ)-differentially private.

Post-processing

- Differential privacy is immune to post-processing

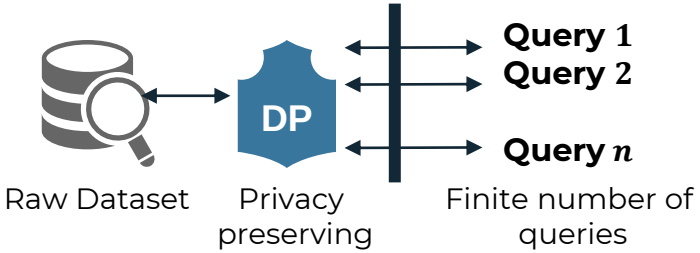
Proposition 2.1 (Post-Processing). Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R$ be a randomized algorithm that is (ε, δ)-differentially private. Let $f : R \to R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R'$ is (ε, δ)-differentially private.

- A data analyst, without additional knowledge about the private database, cannot compute a function of the output of a private algorithm M and make it less differentially private.
- The composition of a **data-independent** mapping f with an (ε,δ)- differentially private algorithm M is also (ε,δ)-differentially private.

Why Composition?

- Reasoning about privacy of a complex algorithm is hard.
- Helps software design
  - – If building blocks are proven to be private, it would be easy to reason about privacy of a complex algorithm built entirely using these building blocks.

Composition Theorems

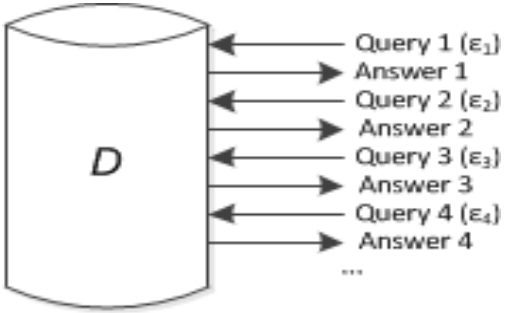


- The parameters ε and δ will necessarily degrade — consider repeatedly computing the same statistic.

A bound on the number of queries

- In order to ensure utility, a statistical database must leak some information about each individual
- We can only hope to bound the amount of disclosure
- Hence, there is a limit on number of queries that can be answered

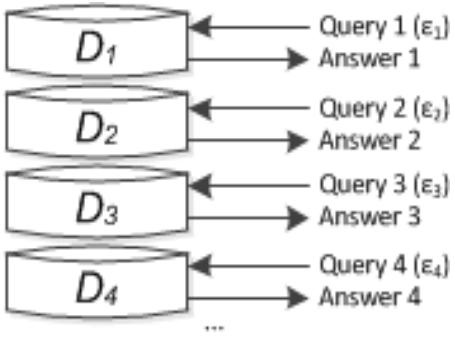
Composition theorems



Sequential composition

$\sum_i \varepsilon_i$ –differential privacy

Relating to the same individual



Parallel composition

$\max(\varepsilon_i)$–differential privacy

Composition Theorems (Sequential)

Theorem Let $\mathcal{M}_1 : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}_1$ be an ε_1-differentially private algorithm, and let $\mathcal{M}_2 : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}_2$ be an ε_2-differentially private algorithm. Then their combination, defined to be $\mathcal{M}_{1,2} : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}_1 \times \mathcal{R}_2$ by the mapping: $\mathcal{M}_{1,2}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x))$ is $\varepsilon_1 + \varepsilon_2$-differentially private.

Proof. Let $x, y \in \mathbb{N}^{|\mathcal{X}|}$ be such that $\|x - y\|_1 \leq 1$. Fix any $(r_1, r_2) \in \mathcal{R}_1 \times \mathcal{R}_2$. Then:

$$\begin{aligned}
\frac{\Pr[\mathcal{M}_{1,2}(x) = (r_1, r_2)]}{\Pr[\mathcal{M}_{1,2}(y) = (r_1, r_2)]} &= \frac{\Pr[\mathcal{M}_1(x) = r_1] \Pr[\mathcal{M}_2(x) = r_2]}{\Pr[\mathcal{M}_1(y) = r_1] \Pr[\mathcal{M}_2(y) = r_2]} \\
&= \left(\frac{\Pr[\mathcal{M}_1(x) = r_1]}{\Pr[\mathcal{M}_1(y) = r_1]}\right)\left(\frac{\Pr[\mathcal{M}_2(x) = r_1]}{\Pr[\mathcal{M}_2(y) = r_1]}\right) \\
&\leq \exp(\varepsilon_1)\exp(\varepsilon_2) \\
&= \exp(\varepsilon_1 + \varepsilon_2)
\end{aligned}$$

By symmetry, $\frac{\Pr[\mathcal{M}_{1,2}(x)=(r_1,r_2)]}{\Pr[\mathcal{M}_{1,2}(y)=(r_1,r_2)]} \geq \exp(-(\varepsilon_1 + \varepsilon_2))$. □

Composition Theorems (Sequential)

- Epsilons and the deltas add up:

Theorem 3.16. Let $\mathcal{M}_i : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}_i$ be an $(\varepsilon_i, \delta_i)$-differentially private algorithm for $i \in [k]$. Then if $\mathcal{M}_{[k]} : \mathbb{N}^{|\mathcal{X}|} \to \prod_{i=1}^{k} \mathcal{R}_i$ is defined to be $\mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), \ldots, \mathcal{M}_k(x))$, then $\mathcal{M}_{[k]}$ is $(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i)$-differentially private.

Group Privacy

» The strength of the privacy guarantee drops linearly with the size of the group:

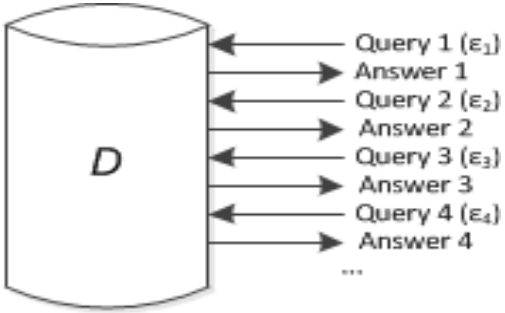
- This addresses, for example, the question of privacy in surveys that include multiple family members.

Theorem 2.2. Any $(\varepsilon, 0)$-differentially private mechanism $\mathcal{M}$ is $(k\varepsilon, 0)$-differentially private for groups of size k. That is, for all $\|x - y\|_1 \leq k$ and all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(k\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}],$$

where the probability space is over the coin flips of the mechanism $\mathcal{M}$.

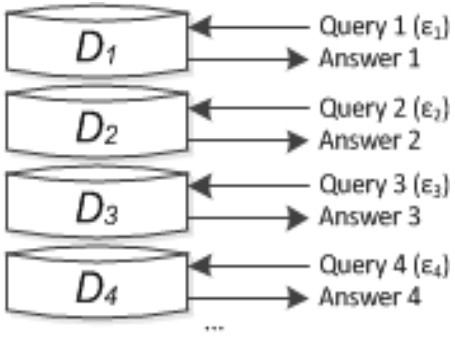
Composition theorems



Sequential composition

$\sum_i \varepsilon_i$ –differential privacy

Relating to the same individual



Parallel composition

$\max(\varepsilon_i)$–differential privacy

Advanced Composition

- To allow the parameters to degrade more slowly.
- To handle more complicated forms of composition.

- **k-fold adaptive composition**: allow the adversary A to choose the databases (parallel composition), mechanisms, and the parameters adaptively depending on the outputs of previous mechanisms.

- $\mathcal{F}$ be a family of mechanisms, Experiment 0 and Experiment 1:

Experiment b for family $\mathcal{F}$ and adversary A:

For $i = 1, \ldots, k$:

1. A outputs two adjacent databases x_i^0 and x_i^1, a mechanism $\mathcal{M}_i \in \mathcal{F}$, and parameters w_i.
2. A receives $y_i \in_R \mathcal{M}_i(w_i, x_{i,b})$.

+Bob -Bob

the adversary "can't tell", given the output of all k mechanisms

Advanced Composition

- A's view of the experiment to be A's coin tosses and all of the mechanism outputs $(y_1, \ldots, y_k)$.

Definition 3.7. We say that the family $\mathcal{F}$ of database access mechanisms satisfies ε-*differential privacy under* k-*fold adaptive composition* if for every adversary A, we have $D_\infty(V^0 \| V^1) \leq \varepsilon$ where V^b denotes the view of A in k-fold Composition Experiment b above.

Theorem 3.20 (Advanced Composition)**.** For all $\varepsilon, \delta, \delta' \geq 0$, the class of (ε, δ)-differentially private mechanisms satisfies $(\varepsilon', k\delta + \delta')$-differential privacy under k-fold adaptive composition for:

$$\varepsilon' = \sqrt{2k \ln(1/\delta')}\varepsilon + k\varepsilon(e^\varepsilon - 1).$$

Advanced Composition

Theorem 3.20 (Advanced Composition). For all $\varepsilon, \delta, \delta' \geq 0$, the class of (ε, δ)-differentially private mechanisms satisfies $(\varepsilon', k\delta + \delta')$-differential privacy under k-fold adaptive composition for:

$$\varepsilon' = \sqrt{2k \ln(1/\delta')}\varepsilon + k\varepsilon(e^{\varepsilon} - 1).$$

Corollary 3.21. Given target privacy parameters $0 < \varepsilon' < 1$ and $\delta' > 0$, to ensure $(\varepsilon', k\delta + \delta')$ cumulative privacy loss over k mechanisms, it suffices that each mechanism is (ε, δ)-differentially private, where

$$\varepsilon = \frac{\varepsilon'}{2\sqrt{2k \ln(1/\delta')}}.$$

Example

- Over the course of his lifetime, Bob is a member of k = 10, 000 $(\varepsilon_0, 0)$ –differentially private databases. No coordination among these databases.
  - What should be the value of ε_0 so that, over the course of his lifetime, Bob's cumulative privacy loss is bounded by ε= 1 with probability at least $1 - e^{-32}$?

Example

- Over the course of his lifetime, Bob is a member of k = 10, 000 $(\varepsilon_0, 0)$ –differentially private databases. No coordination among these databases.
  - What should be the value of ε_0 so that, over the course of his lifetime, Bob's cumulative privacy loss is bounded by ε= 1 with probability at least $1 - e^{-32}$?
  - taking $\delta' = e^{-32}$, it suffices to have$\varepsilon_0 < 1/801$

What differential privacy promises

An Economic View

- DP promises to protect individuals from any additional harm that they might face due to their data being in the private database x that they would not have faced had their data not been part of x.
- Although individuals may indeed face harm once the results $M(x)$ of a DP mechanism M have been released, differential privacy promises that the probability of harm was not significantly increased (an $\exp(\varepsilon) \sim (1+\varepsilon)$ factor) by their choice to participate.

What differential privacy does not promise

- DP is an extremely strong guarantee, but it does not promise unconditional freedom from harm.
- DP does not create privacy where none previously exists.
- DP does not guarantee that what one believes to be one's secrets will remain secret.
- DP only ensures that one's participation in a survey will not in itself be disclosed, nor will participation lead to disclosure of any specifics that one has contributed to the survey.

Qualitative Properties of DP

- **Automatic neutralization of linkage attacks**: including all those attempted with all past, present, and future datasets and other auxiliary information.
- **Quantification of privacy loss**
- **Composition**: permits the analysis and control of cumulative privacy loss over multiple computations.
- **Group privacy**: permits the analysis and control of privacy loss incurred by groups, such as families.

Granularity of Privacy

- Be careful about the level of granularity at which privacy is being promised.
  - DP promises that the behavior of an algorithm will be roughly unchanged even if **a single entry** in the database is modified.
  - **What constitutes a single entry in the database?**

Granularity of Privacy

- **Edge** differential privacy

G:  G': 

Two graphs are **neighbors** if they differ in ***one edge***.

- **Node** differential privacy

G:  G': 

Two graphs are **neighbors** if one can be obtained from the other by deleting ***a node and its adjacent edges***.

Granularity of Privacy

- **Edge** differential privacy

G:  G': 

Weaker but sometimes sufficient. No data analyst should be able to conclude anything about the existence of any subset of $1/\varepsilon$ edges in the graph.

G:  G': 

Two graphs are **neighbors** if one can be obtained from the other by deleting ***a node and its adjacent edges***.

All Small Epsilons Are Alike

- The nature of the privacy guarantees with differing but small epsilons are quite similar.
  - When " is small, failing to be $(\varepsilon, 0)$-differentially private is not necessarily alarming, the mechanism may be $(2\varepsilon, 0)$-differentially private.
- What of large values for ε?
  - Merely means there exist neighboring databases x or y and an output o for which $P(M(x) = o)/P(M(y) = o)$ is large.
  - The output o might be very unlikely (addressed by (ε, δ)-dp).
  - Databases x and y might be terribly contrived.
  - The adversary may not have the right auxiliary information to recognize that a revealing output has occurred.
- The failure to be $(\varepsilon, 0)$ or (ε, δ)- differentially private may range from effectively meaningless privacy breaches to complete revelation of the entire database.

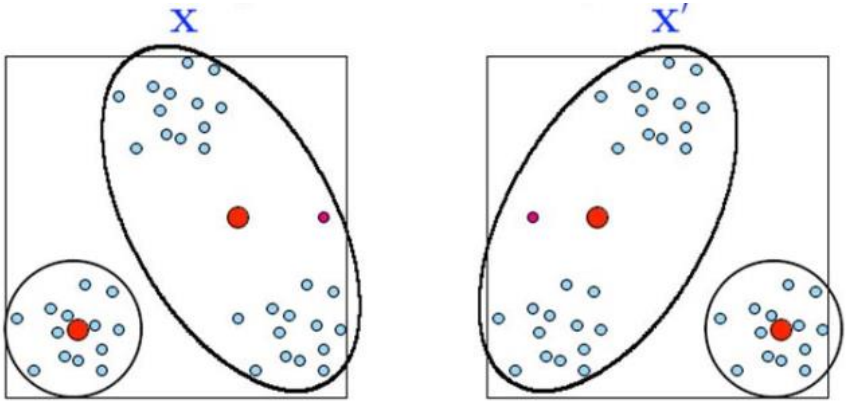
3. Analysis

Alternatives to Global Sensitivity

- Local Sensitivity
- Smooth Sensitivity

Examples of High Global Sensitivity

Database entries: points in a metric space.



- Global sensitivity of clusters is roughly the diameter of the space.
- But intuitively, if clustering is “good”, cluster centers should be insensitive.

Examples of High Global Sensitivity

- Median, Max, Min…

$$D = \{\underbrace{0, 0, \ldots, 0}_{\frac{n-1}{2}}, 0, \underbrace{k, k, \ldots, k}_{\frac{n-1}{2}}\} \quad D' = \{\underbrace{0, 0, \ldots, 0}_{\frac{n-1}{2}}, k, \underbrace{k, k, \ldots, k}_{\frac{n-1}{2}}\}$$

- Global sensitivity is k! Perturb by $Lap(k/\epsilon)$ gives no utility at all.
- But for most neighbor databases x, x', $|median(x) -$

$$D = \{1, 2, 2, \ldots, \frac{k}{2}, \frac{k}{2}, \frac{k}{2}, \frac{k}{2} + 1, \ldots, k, k\}$$

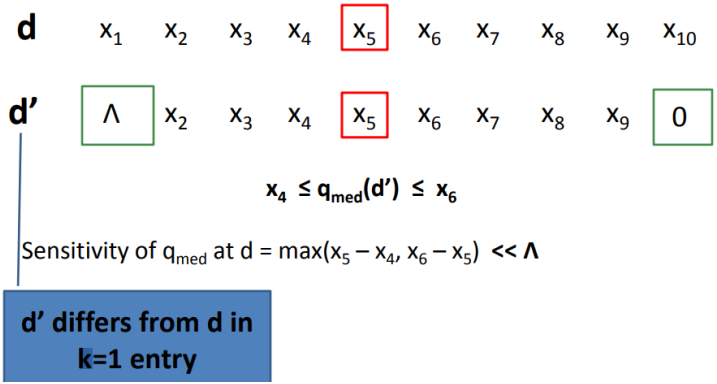
Local Sensitivity

- Local sensitivity of median

For sorted $X = x_1, x_2, \ldots, x_n$,

$$LS_{median}(X) = \max(x_m - x_{m-1}, x_{m+1} - x_m),$$

where n is odd and $m = (n+1)/2$.



Global Sensitivity vs. Local Sensitivity

- Global sensitivity depends on the function
- **Local sensitivity** of query q at point D: [Nissim et al., STOC 2007]

$$LS_q(D) = \max_{D' \in N(D)} \left| q(D) - q(D') \right|$$

- Reminder: $GS_q(D) = \max_D LS_q(D)$

Goal: add less noise when local sensitivity is small

- Problem: can **leak information** by **amount of noise**

Nissim, Kobbi, Sofya Raskhodnikova, and Adam Smith. "Smooth sensitivity and sampling in private data analysis." *symposium on the theory of computing* (2007): 75-84.

Noise proportional to Local Sensitivity

- $d_1 = \{0, 0, 0, 0, 0, 0, \Lambda, \Lambda, \Lambda, \Lambda\}$
  $q_{med}(d_1) = 0$
  $LS_{qmed}(d_1) = 0 \Rightarrow$ Noise sampled from Lap(0)

- $d_2 = \{0, 0, 0, 0, 0, \Lambda, \Lambda, \Lambda, \Lambda, \Lambda\}$
  $q_{med}(d_2) = 0$
  $LS_{qmed}(d_2) = \Lambda \Rightarrow$ Noise sampled from Lap(Λ/ε)

$$\Pr[\text{answer} > 0 \mid d_2] > 0 \quad \Pr[\text{answer} > 0 \mid d_1] = 0 \quad \text{implies} \quad \frac{\Pr[\text{answer} > 0 \mid d_2] > 0}{\Pr[\text{answer} > 0 \mid d_1] = 0} = \infty$$

Local Sensitivity

- Local sensitivity is defined to particular data D, but adding $\boldsymbol{Lap\left(LS_f(D)/\varepsilon\right)}$ **does not guarantee** $\boldsymbol{\varepsilon - DP}$**.**
  - E.g., $LS_{qmed}(d)$ has very high sensitivity

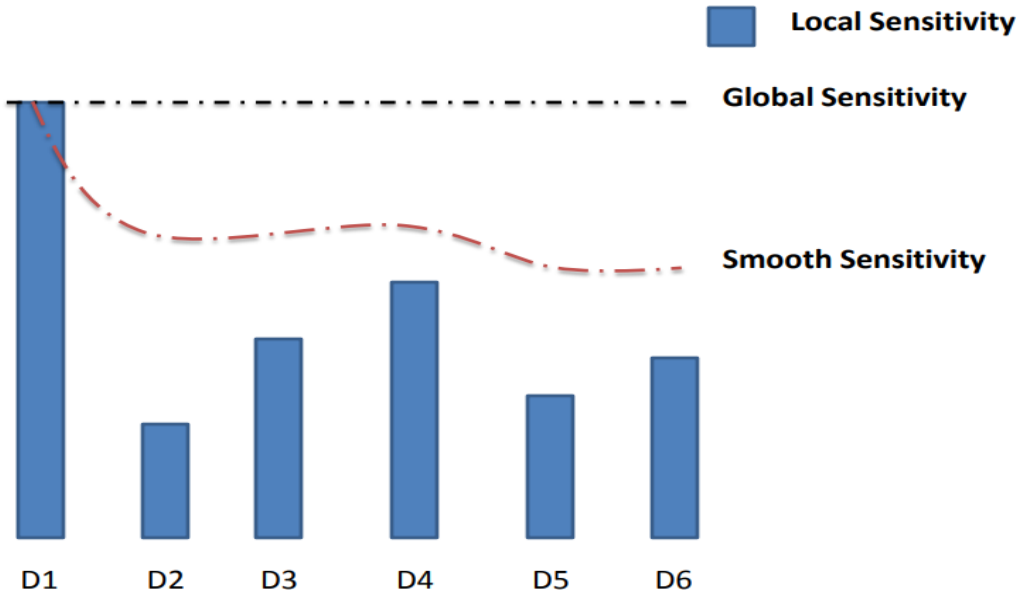
- Problem: can leak information by amount of noise

The noise magnitude has to be an insensitive function.

- Solution: Smooth the sensitivity of f on the database. (It should not change quickly in any neighborhood of its input space.)

Instance-specific Noise

» Solution: Smooth the upper bound of LS_f, that adding noise proportional to S_f is safe.



Smooth Upper Bound

Compute a “smoothed” version of local sensitivity

- Design sensitivity function $S(D)$

$S(D)$ is an β-smooth upper bound on $LS_f(D)$ if:

- For all D: $S(D) \geq LS_f(D)$
- For all neighbors D and D': $S(X) \leq e^{\beta} S(D')$

Smooth Sensitivity Mechanism

Definition 1 (Smooth sensitivity). *For $\beta > 0$, the β-smooth sensitivity of f is:*

$$S^*_{f,\beta}(D) = \max_{D' \subset X} LS_f(D') \exp(-\beta d(D', D)).$$

- $S^*_{f,\beta}(D)$ is the smallest β-smooth upper bound on LS_f

Calibrating Noise to Sensitivity

- $A(x) = f(x) + N(x) \cdot Z$
- Z is a random variable drawn from a noise distribution
- $N(x)$ is the noise magnitude

Calibrating Noise to Sensitivity

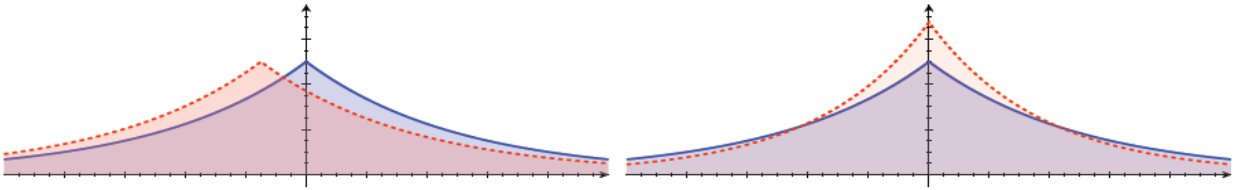
Admissible Noise Distribution

- The noise distribution doesn't change much under translation (sliding) and scaling (dilation).
- A distribution satisfying the two properties can be used to add noise proportional to $S(D)$:

Definition 2.5 (Admissible Noise Distribution)**.** *A probability distribution on $\mathbb{R}^d$, given by a density function h, is* (α, β)-admissible *(with respect to ℓ_1) if, for $\alpha = \alpha(\epsilon, \delta), \beta = \beta(\epsilon, \delta)$, the following two conditions hold for all $\Delta \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}$ satisfying $\|\Delta\|_1 \leq \alpha$ and $|\lambda| \leq \beta$, and for all measurable subsets $\mathcal{S} \subseteq \mathbb{R}^d$:*

Sliding Property:
$$\Pr_{Z\sim h}\left[Z \in \mathcal{S}\right] \leq e^{\frac{\epsilon}{2}} \cdot \Pr_{Z\sim h}\left[Z \in \mathcal{S} + \Delta\right] + \frac{\delta}{2}.$$

Dilation Property:
$$\Pr_{Z\sim h}\left[Z \in \mathcal{S}\right] \leq e^{\frac{\epsilon}{2}} \cdot \Pr_{Z\sim h}\left[Z \in e^{\lambda} \cdot \mathcal{S}\right] + \frac{\delta}{2}.$$



Calibrating Noise to Sensitivity

- Let h be an (α, β)-admissible noise probability density function, and let Z be a fresh random variable sampled according to h. For a function $f: D^n \to R^d$, let $\boldsymbol{S: D^n \to R}$ **be a $\boldsymbol{\beta}$-smooth upper bound on the local sensitivity of $\boldsymbol{f}$**. Then **algorithm** $\boldsymbol{A(x) = f(x) + \frac{S(x)}{\alpha} \cdot Z}$ **is (ε, δ)-differentially private**.

- For two neighbor databases x and y, the output distribution $A(y)$ is a shifted and scaled version of $A(x)$. The sliding and dilation properties ensure that $\Pr[A(x) \in S]$ and $\Pr[A(y) \in S]$ are close for all sets S of outputs.

Calibrating Noise to Sensitivity

- **Example:** Laplace distribution
  - $Lap(b) = \frac{1}{2b} exp(-\frac{|x|}{b})$
  - $Lap(1)$ is (α, β)-admissible with $\alpha = \varepsilon/2$, $\beta = \frac{\varepsilon}{2\ln(\frac{1}{\delta})}$. In dimension d>1, one can use the product of Laplace distributions with $\beta = \Omega(\frac{\varepsilon}{\sqrt{d}\ln(\frac{1}{\delta})})$. The algorithm $x \to f(x) + \frac{2(S(x))}{\varepsilon} \cdot \eta$, where $\eta \sim Lap(1)$, is $(\boldsymbol{\varepsilon}, \boldsymbol{\delta})$-differentially private.

Calibrating Noise to Sensitivity

- Sensitivity
  - Global sensitivity of query q:

$$GS_q = \max_{D,D'} \left\| q(D) - q(D') \right\|_1$$

  - Local sensitivity of query q at point:

$$LS_q(D) = \max_{D'} \left\| q(D) - q(D') \right\|_1$$

  - Smooth sensitivity (Much smaller than global sensitivity):

$$S_q^*(D) = \max_{D'} \{ LS_q(D) \exp(-\epsilon dist(D, D')) \}$$

Calibrating Noise to Sensitivity

- Sensitivity
  - Global sensitivity of query q:

$$GS_q = \max_{D,D'} \left\| q(D) - q(D') \right\|_1 \quad E.g., Noise\ Z \sim Lap(GS_q/\varepsilon)$$

For sum queries, principle component analysis, k-means, learning ID3 decision trees, statistical learning, histograms, singular value decomposition …

  - Local sensitivity of query q at point:

$$LS_q(D) = \max_{D'} \left\| q(D) - q(D') \right\|_1$$

  - Smooth sensitivity (Much smaller than global sensitivity):

$$S_q^*(D) = \max_{D'} \{ LS_q(D) \exp(-\epsilon dist(D, D')) \}$$

$$E.g., Noise\ Z \sim Lap(S_q^*/\varepsilon)$$

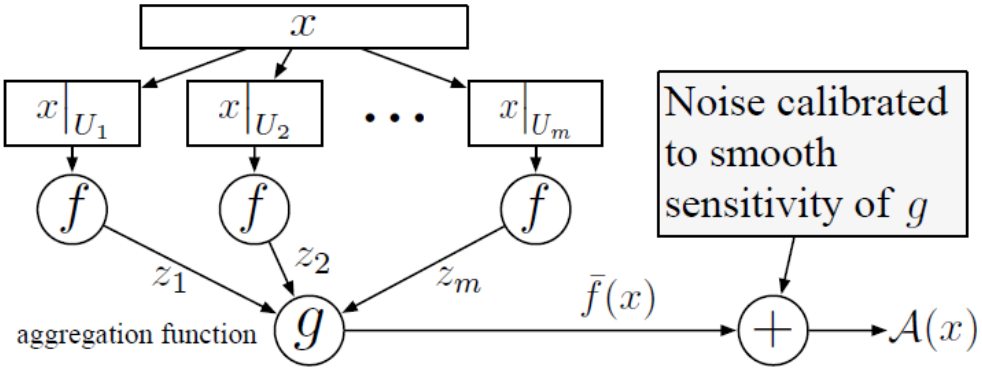
Median, Max, Min, number of triangles in a graph, cluster centers…

Smooth Sensitivity Mechanism

- In order to use it for releasing a function f, one needs to design an efficient algorithm for computing or approximating the smooth sensitivity of f.
- For a given function, such an algorithm might not exist or might be difficult to design. (e.g., the cluster centers for various clustering problems and learning the mixtures of Gaussians)
- In the interactive model, it implies that the smooth sensitivity for all allowed user requests has to be analyzed and known in advance.

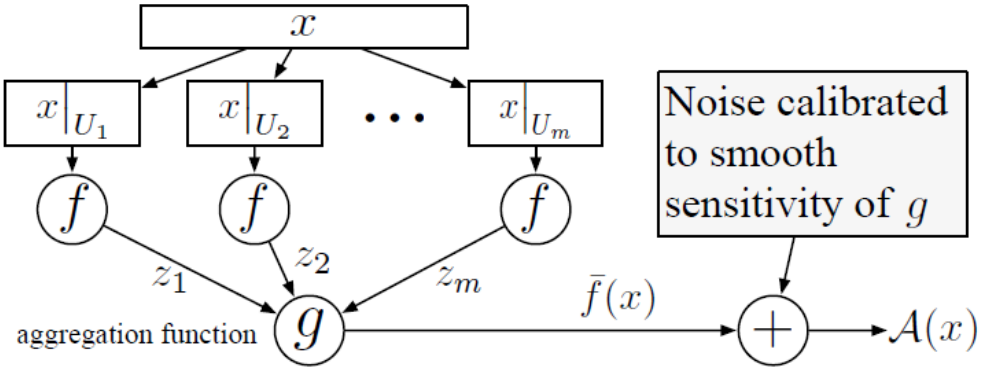
Sample and Aggregate

- The sample and aggregate framework provides an efficient database access mechanism that treats the query function as a black box. It can be applied without an explicit computation of smooth sensitivity.
- Sample and aggregate works by replacing f with a related function $\bar{f}$ for which smooth sensitivity is low and efficiently computable.



Sample and Aggregate

» Bound the smooth sensitivity of the function $\bar{f}$ at x by the smooth sensitivity of the aggregation function g at $z = (z_1, \ldots, z_m)$. Changing a single point in the original database x will change very few small databases, and hence very few evaluations $z_1, \ldots, z_m$.



Reconstruction attacks

- Dataset: $D \in \{0,1\}^n$.
- Query: take $q \in \{0,1\}^n$, the query function specified by q is $f(D,q) = 1/n < q, D >$. (how many 1's are in the selected rows)
- Answer: $A(q), r(q)$
- Error/distortion: $E(A(q) - r(q))$

Goal:

Find out how inaccurate the answers for multiple counting queries must be in order to preserve privacy?

Blatant non-privacy

Definition 1 (Blatant non-privacy)**.** *A mechanism is blatantly non-private if an adversary can construct a candidate database c that agrees with the real database d in all but $o(n)$ entries, i.e., $\|c - d\|_0 \in o(n)$.*

- A mechanism is blatantly non-private if it permits a reconstruction attack that allows the adversary to correctly guess the secret bit of all but o(n) members of the database.

Reconstruction Attack

Theorem 1 (Reconstruction from exponential numbers of queries)**.** *Let* $D \in \{0,1\}^n$*, if we are given, for each* $q \in \{0,1\}^n$*, a value* $y_q \in \mathbb{R}$ *such that*

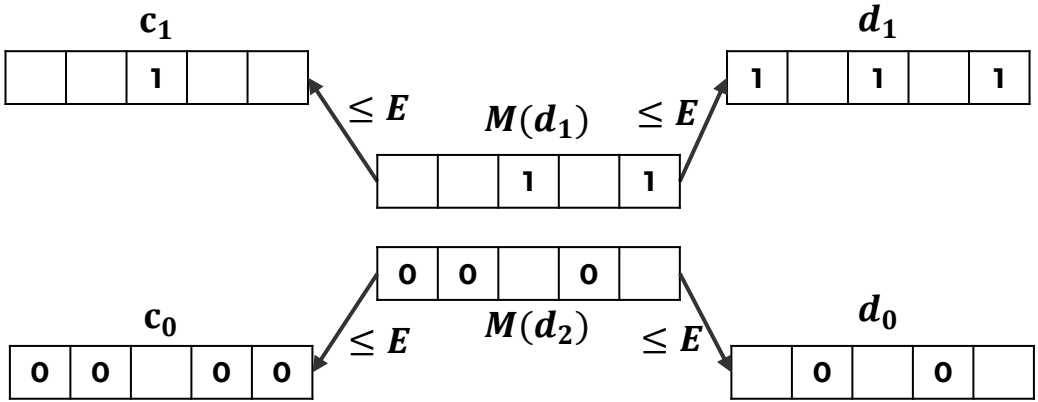
$$\left| y_q - \frac{\langle q, x \rangle}{n} \right| \leq \alpha.$$

Then one can use the y_q*'s to compute* $x' \in \{0,1\}^n$ *such that* x *and* x' *differ in at most* 4α *fraction of coordinates.*

Corollary 1. *If* $M(x)$ *is a mechanism that outputs values* y_q *as above with* $\alpha \leq o(n)$*, then* M *is blatantly non-private.*

Reconstruction Attack

1. **Estimate the number of 1s in all possible sets:** Query $\mathcal{M}$ on all subsets $S \subseteq [n]$.
2. **Rule out "distant" databases:** For every candidate database $c \in \{0,1\}^n$, if $\exists S \subseteq [n]$ such that $|\sum_{i \in S} c_i - \mathcal{M}(S)| > E$, then rule out c. If c is not ruled out, then output c and halt.



Reconstruction Attack

1. **Estimate the number of 1s in all possible sets:** Query $\mathcal{M}$ on all subsets $S \subseteq [n]$.
2. **Rule out "distant" databases:** For every candidate database $c \in \{0,1\}^n$, if $\exists S \subseteq [n]$ such that $|\sum_{i \in S} c_i - \mathcal{M}(S)| > E$, then rule out c. If c is not ruled out, then output c and halt.

Example: a privacy mechanism adding noise with magnitude always bounded by, say, *n*/401, permits an adversary to correctly reconstruct **X%** of the entries.

Reconstruction Attack (fewer queries)

Theorem 2 (Reconstruction from fewer queries)**.** *A mechanism which answers the counting queries specified by* $q_1, \ldots, q_m$ *to within error at most* $o(\sqrt{n})$ *is blatantly non-private, where* $m = O(n \log^2 n)$.

[Query Phase]
Let $t = n(\log n)^2$. For $1 \leq j \leq t$ choose uniformly at random $q_j \subseteq_R [n]$, and set $\tilde{a}_{q_j} \leftarrow \mathcal{A}(q_j)$.

[Weeding Phase]
Solve the following linear program with unknowns $c_1, \ldots, c_n$:

$$\begin{array}{ll} \tilde{a}_{q_j} - \mathcal{E} \leq \sum_{i \in q_j} c_i \leq \tilde{a}_{q_j} + \mathcal{E} & \text{for } 1 \leq j \leq t \\ 0 \leq c_i \leq 1 & \text{for } 1 \leq i \leq n \end{array} \quad (1)$$

[Rounding phase]
Let $c'_i = 1$ if $c_i > 1/2$ and $c'_i = 0$ otherwise. Output c'.

How many queries can we answer?

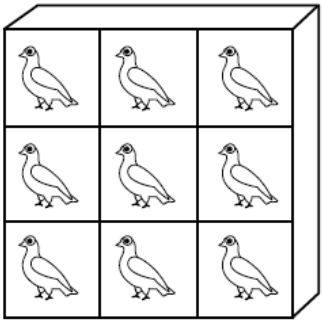
- On a database of size n
  - If the error is of order $o(n)$, for fixed values of ε and δ, it is possible to answer close to n^2 counting queries
  - If the error is of order $o(\sqrt{n})$, for fixed values of ε and δ, it is possible to answer close to n counting queries
  - It is possible to dramatically improve on these results: an exponential number of queries with noise only slightly larger than $o(\sqrt{n})$, by **coordinating the noise added to the individual responses.**

Lower bounds for differential privacy

- If the adversary has narrowed down the set of possible databases to a relatively small set S, where the L_1 distance between each pair of vectors is large. We can find a k-dimensional query that the true answers to the query look very different (response regions are disjoint) on the different vectors in S.
- Each element x in the set S gives rise to a vector $F(x)$ in answer space. The actual response will be a perturbation of this point in answer space.
- If with even moderate probability the (noisy) responses are "reasonably" close to the true answers, then ε cannot be very small (privacy can't be too good).

If $\mathcal{M}$ is an $(\varepsilon, 0)$-differentially private mechanism for F such that, $\forall 1 \leq i \leq 2^s$, $\Pr[\mathcal{M}(x_i) \in B_i] \geq 1/2$, then $\varepsilon \geq \frac{\ln(2)(s-1)}{\Delta}$.

THE PIGEONHOLE PRINCIPLE



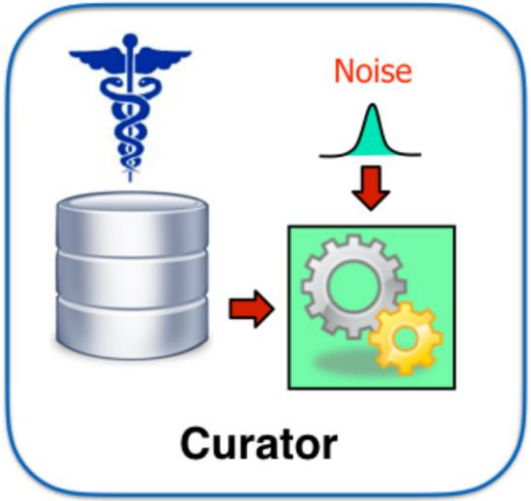
4. Distributed Differential Privacy

Multi-party Differential Privacy

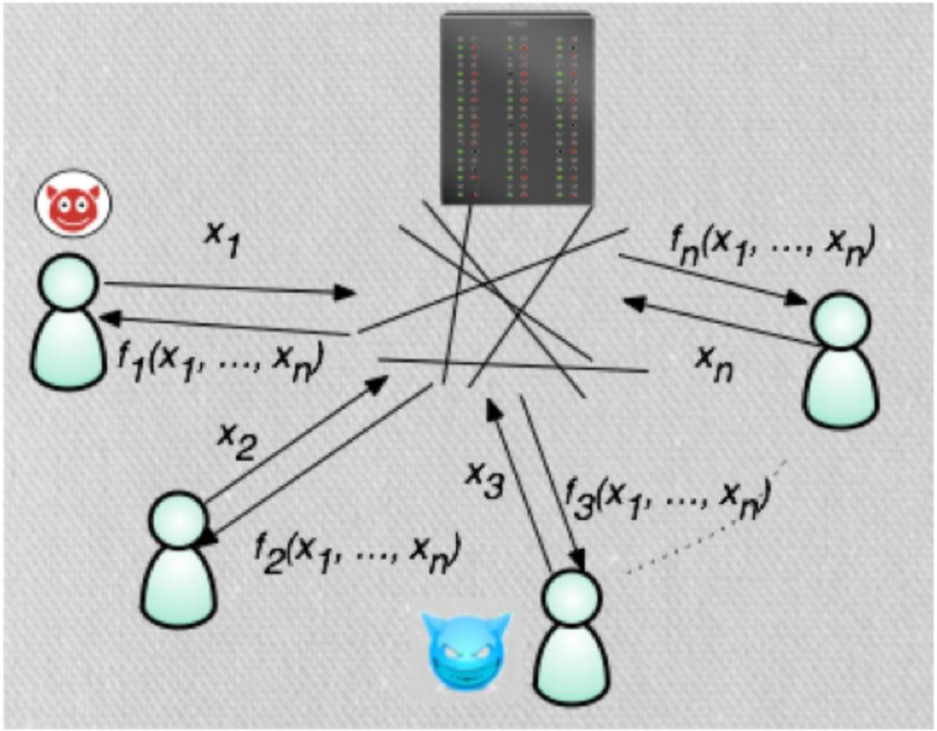
- Definition of multi-party differential privacy
- Two-party differential privacy

Differential Privacy

» In DP, we consider a curator model where there exists a trusted centralized party that holds the data and to which we ask out queries.

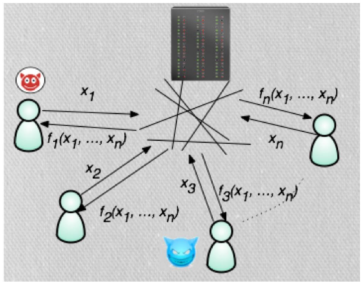


Multi-party Differential Privacy



Multi-party Setting

- Consider a model where the data is distributed among m parties $P_1, \ldots, P_m$.
- Assume the data is evenly split among the parties, each party has n/m rows of the dataset.
- Each party wants to guarantee privacy for its data against an adversary that may control other parties.



Adversaries

- We assume that the adversaries are:
  - Passive (honest-but-curious): they follow the specifies protocol but try to extract information from what they see.
  - Computationally unbounded: we will not restrict the capacity of the adversary.
  - Control several parties: an adversary can control $t \leq m - 1$ parties. We will focus on $t = m - 1$.

Protocol

- We consider a protocol as a sequence of rounds where:
  - Every party P_i selects a message to be broadcast based on its input (a part of x), internal coin tosses, and all messages received in previous rounds.
  - The output of the protocol is specified by a deterministic function of the transcript of messages exchanged.

$$(P_1, \ldots, P_m)(x)$$

Adversary view

- We are interested in a protection against an adversary that controls all the parties except the k-th one.
- The view of the adversary is then determined by the inputs and coin tosses of all parties other than P_k as well as the messages sent by P_k.

$$\text{View}_{P_{-k}}(P_{-k} \leftrightarrow (P_1, \ldots, P_m)(x)) \in T$$

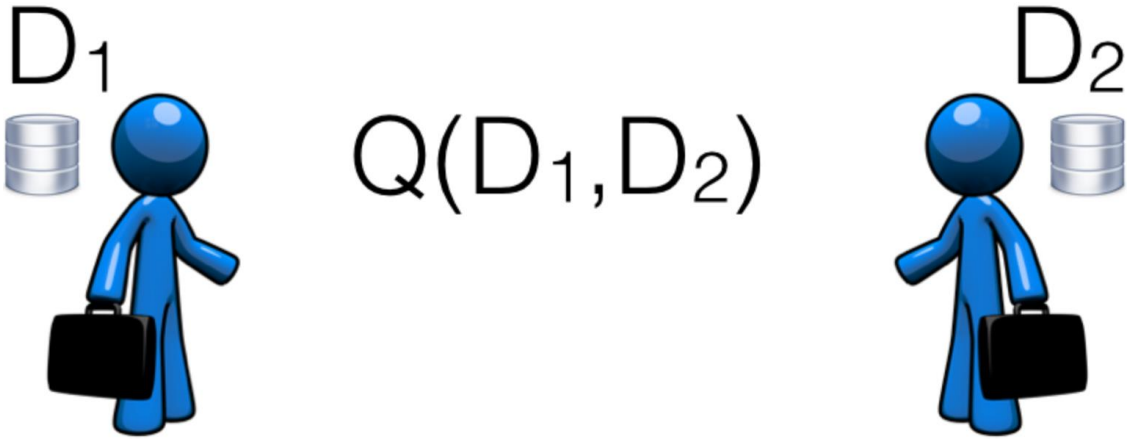
Multi-party differential privacy

Definition 1 (Multi-party differential privacy). *For a protocol* $P = (P_1, \ldots, P_m)$ *taking as inputs datasets* $(x_1, \ldots, x_m) \in ((\mathcal{X}^{n/m})^m)$, *we say that* P *is* (ϵ, δ)*–differentially private if for every* $k \in [m]$ *and every two datasets* $x, x' \in ((\mathcal{X}^{n/m})^m)$ *that differ on one row of* P_k*'s input (and are equal otherwise), the following holds for every set* T*:*

$$\Pr[View_{P_{-k}}(P_{-k} \leftrightarrow (P_1, \ldots, P_m)(x)) \in T] \leq$$
$$\exp(\epsilon) \Pr[View_{P_{-k}}(P_{-k} \leftrightarrow (P_1, \ldots, P_m)(x')) \in T] + \delta.$$

Two-party differential privacy

- Now consider the case of two parties that want to compute a common statistics.
- Each party has a database of size $n/2$.



Counting queries in the 2-party model

- How can we compute efficiently a counting query $q: \{0,1\}^n \to R$ in the 2-party model where $q(D) = \Sigma_{i=1}^{n} D_i$?
- Protocol:
  - Each party P_i computes $a_i = q(D_i) + Lap(2/\epsilon)$ and shares it.
  - Collect the results and compute $a = a_1 + a_2$.

THANKS!

Any questions?

You can find me at:

- zhanglan@ustc.edu.cn