Foundations of Privacy

Lecture 7

Plan of the lecture

- Solution of the exercise
- Brief recall of the Laplacian mechanism
- Discrete queries and Geometric Mechanism
- Truncated mechanisms
- Utility
- Optimal Mechanisms

Plan of the lecture

- Solution of the exercise
- Brief recall of the Laplacian mechanism
- Discrete queries and Geometric Mechanism
- Truncated mechanisms
- Utility
- Optimal Mechanisms

Plan of the lecture

- Solution of the exercise
- **Brief recall of the Laplacian mechanism**
- Discrete queries and Geometric Mechanism
- Truncated mechanisms
- Utility
- Optimal Mechanisms

A typical oblivious differentially private mechanism: Laplacian noise

- Randomized mechanism for a query $f : \mathcal{X} \rightarrow \mathcal{Y}$.
- A typical randomized method: **add Laplacian noise.** If the exact answer is y, the reported answer is z, with a probability density function defined as:

$$dP_y(z) = c\, e^{-\frac{|z-y|}{\Delta f}\varepsilon}$$

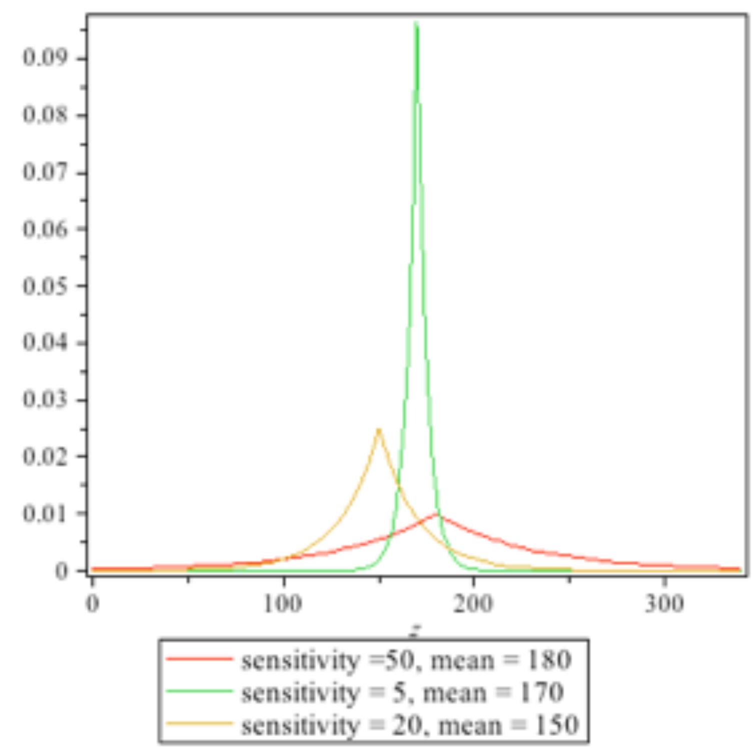
where Δf is the *sensitivity* of f:

$$\Delta f = \max_{x \sim x' \in \mathcal{X}} |f(x) - f(x')|$$

($x \sim x'$ means x and x' are adjacent, i.e., they differ only for one record)

and c is a normalization factor:

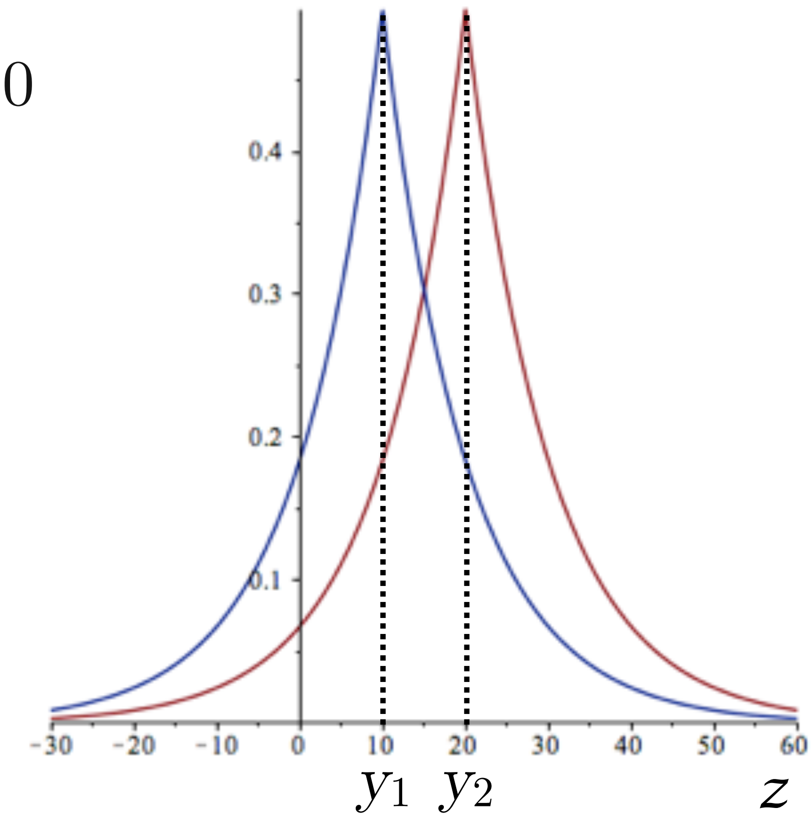
$$c = \frac{\varepsilon}{2\,\Delta f}$$



Example

- $\varepsilon = 1$
- $\Delta_f = |f(x_1) - f(x_2)| = 10$
- $y_1 = f(x_1) = 10,\ y_1 = f(x_2) = 20$

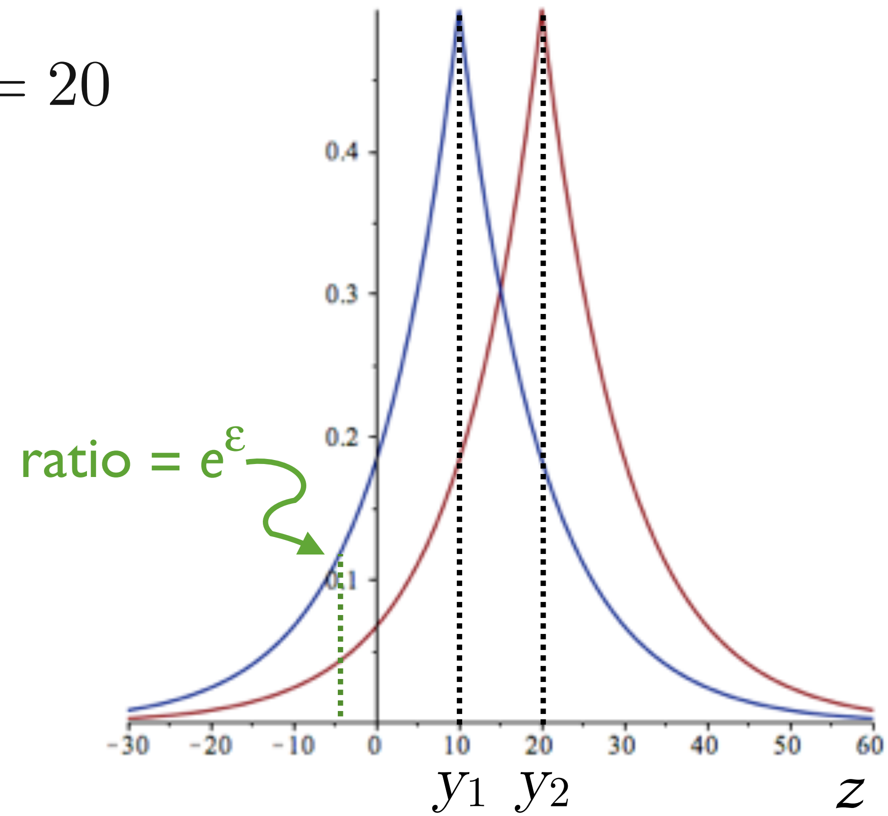
  Then:
- $dP_{y_1} = \frac{1}{2 \cdot 10} e^{\frac{|z-10|}{10}}$
- $dP_{y_2} = \frac{1}{2 \cdot 10} e^{\frac{|z-20|}{10}}$



Example

- $\varepsilon = 1$
- $\Delta_f = |f(x_1) - f(x_2)| = 10$
- $y_1 = f(x_1) = 10,\ y_1 = f(x_2) = 20$

  Then:
- $dP_{y_1} = \frac{1}{2 \cdot 10} e^{\frac{|z-10|}{10}}$
- $dP_{y_2} = \frac{1}{2 \cdot 10} e^{\frac{|z-20|}{10}}$



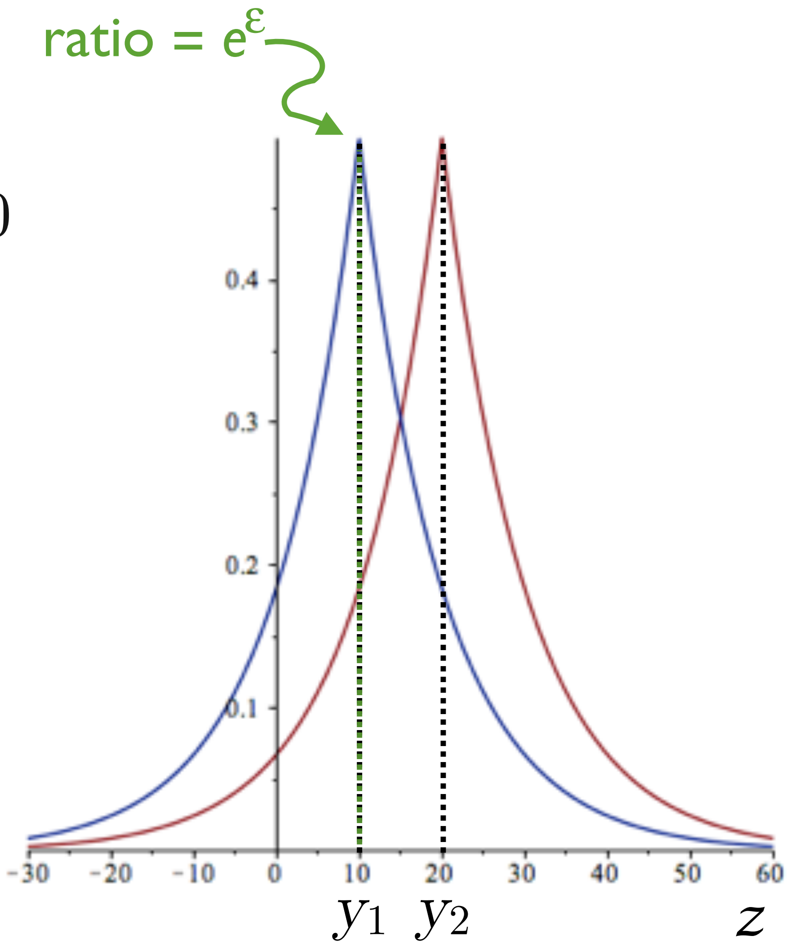
The ratio between these distribution is

- $= e^{\varepsilon}$ outside the interval $[y_1, y_2]$
- $\leq e^{\varepsilon}$ inside the interval $[y_1, y_2]$

Example

- $\varepsilon = 1$
- $\Delta_f = |f(x_1) - f(x_2)| = 10$
- $y_1 = f(x_1) = 10,\ y_1 = f(x_2) = 20$

  Then:
- $dP_{y_1} = \frac{1}{2 \cdot 10} e^{\frac{|z-10|}{10}}$
- $dP_{y_2} = \frac{1}{2 \cdot 10} e^{\frac{|z-20|}{10}}$



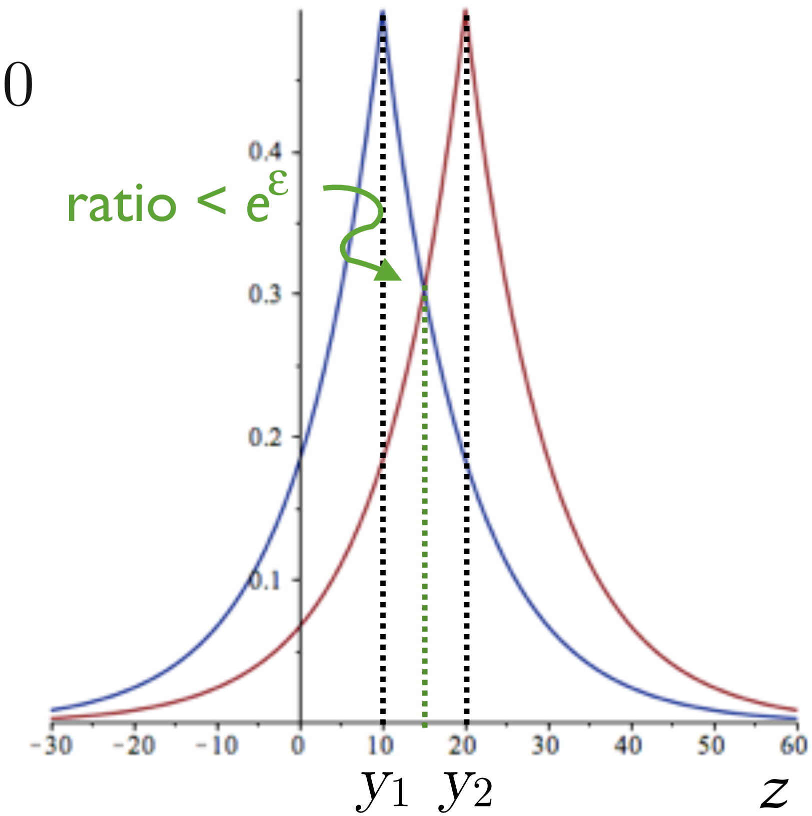
The ratio between these distribution is

- $= e^{\varepsilon}$ outside the interval $[y_1, y_2]$
- $\leq e^{\varepsilon}$ inside the interval $[y_1, y_2]$

Example

- $\varepsilon = 1$
- $\Delta_f = |f(x_1) - f(x_2)| = 10$
- $y_1 = f(x_1) = 10, \ y_1 = f(x_2) = 20$

  Then:
- $dP_{y_1} = \frac{1}{2 \cdot 10} e^{\frac{|z-10|}{10}}$
- $dP_{y_2} = \frac{1}{2 \cdot 10} e^{\frac{|z-20|}{10}}$



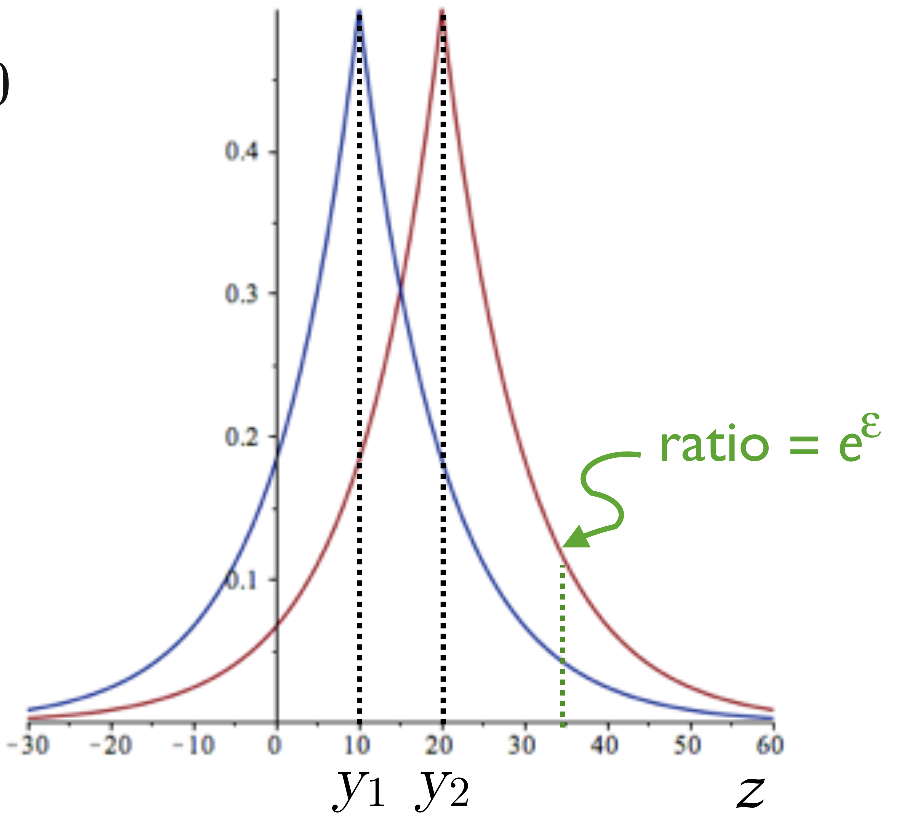
The ratio between these distribution is

- $= e^{\varepsilon}$ outside the interval $[y_1, y_2]$
- $\leq e^{\varepsilon}$ inside the interval $[y_1, y_2]$

Example

- $\varepsilon = 1$
- $\Delta_f = |f(x_1) - f(x_2)| = 10$
- $y_1 = f(x_1) = 10,\ y_1 = f(x_2) = 20$

  Then:
- $dP_{y_1} = \frac{1}{2 \cdot 10} e^{\frac{|z-10|}{10}}$
- $dP_{y_2} = \frac{1}{2 \cdot 10} e^{\frac{|z-20|}{10}}$



The ratio between these distribution is

- $= e^{\varepsilon}$ outside the interval $[y_1, y_2]$
- $\leq e^{\varepsilon}$ inside the interval $[y_1, y_2]$

Gaussian noise

A gaussian noise would not satisfy differential privacy (although it satisfies a more relaxed form of privacy called (ε,δ)-privacy)

In fact, the formula for gaussian noise would be

$$c\ e^{-\frac{(y-z)^2}{\sigma}\varepsilon}$$

and we can easily check that it does not satisfy DP for any value of σ

Sensitivity of the query in a Laplacian

- The sensitivity of the query and the level of privacy ε determine how uniform the noise is:
  - higher sensitivity ⇒ more uniform noise
  - smaller ε ⇒ more privacy, more uniform noise
- Intuitively, the more uniform is the noise, the less useful is the mechanism (the reported answer is less precise)
- To reduce the sensitivity of the query, we often assume that the database contains a minimum number of individuals
- **Example:** consider the query "What is the average age of the people in the DB ?". Assume that the age can vary from 0 to 120. Check the sensitivity in the following two cases:
  - the DB contains at least 100 records, or
  - there is no restriction.

Plan of the lecture

- Solution of the exercise
- Brief recall of the Laplacian mechanism
- Discrete queries and Geometric Mechanism
- Truncated mechanisms
- Utility
- Optimal Mechanisms

The geometric mechanism

- The Laplacian noise is typically used in the case that $\mathcal{Y}$ (the set of true answers of the query) is a **dense** numerical set, like the Reals or the Rationals.

- If $\mathcal{Y}$ is a **discrete** numerical set, like the Integers, then the typical mechanism used in this case is the **geometric mechanism**, which is a sort of discrete Laplacian.

- In the geometric mechanism, the probability distribution of the noise is:

$$p(z|y) = c\, e^{-\frac{|z-y|}{\Delta f}\varepsilon}$$

  - In this expression, c is a normalization factor, defined so to obtain a probability distribution,

  - Δf is the sensitivity of query f

Example: Counting Queries

- Counting queries are typical examples of discrete queries. They are of the form: How many individuals in the database satisfy the property $\mathcal{P}$?
  - Examples:
    - How many individuals are affected by diabetes?
    - How many diabetic people are obese?
- Question: what is the sensitivity of a counting query?

Normalization constant in a geometric mechanism

- In the geometric mechanism, the probability distribution of the noise is:

$$p(z|y) = c\, e^{-\frac{|z-y|}{\Delta f}\varepsilon}$$

As usual, we can compute c (the normalization factor) by imposing that the sum of the probability on all Z is 1. It turns out that

$$c = \frac{1-\alpha}{1+\alpha} \qquad \text{where} \qquad \alpha = e^{-\frac{\varepsilon}{\Delta f}}$$

$$\text{hence} \qquad p(z|y) = \frac{1-\alpha}{1+\alpha}\, \alpha^{|z-y|}$$

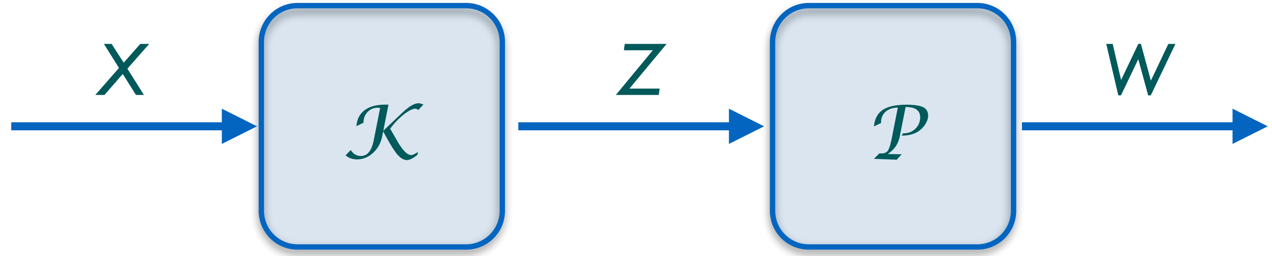
- **Examples:** Compute the geometric mechanism for the following queries:
  - “ How many diabetic people weight more than 100 kilos ? ”
  - “ What is the max weight (in kilos) of a diabetic person ? ”

Truncated geometric mechanism

- Often $\mathcal{Y}$ (the set of the true answers) does not coincide with the whole set of integers, but it is just subset, for instance an interval [a,b].
- With the geometric mechanism, however, the set of reported answers Z is always the whole set of integers
- It is often considered that it does not make much sense to report answers outside $\mathcal{Y}$. If $\mathcal{Y}$ is an interval [a,b], we can **truncate** the mechanism, i.e., set $\mathcal{Z} = \mathcal{Y}$, and transfer on the extremes a and b all the probability that (according to the geometric mechanism) would fall outside the interval: The probability that would fall to the left of a is transferred into a, and probability that would fall to the right of b is transferred into b.
- The same considerations hold for the Laplacian (truncated Laplacian)
- Exercise: Compute the truncated geometric mechanism for a counting query if the interval is [0,100]

Post-processing

- Post-processing a mechanism $\mathcal{K}$ consists in composing $\mathcal{K}$ with another function $\mathcal{P}$
  - $\mathcal{P}$ can be probabilistic or deterministic
  - $\mathcal{K}$ can be oblivious or not — it does not matter for the theorem below



Theorem: Post processing does not harm privacy. Namely, if $\mathcal{K}$ is ε-differentially private, then also $\mathcal{P} \circ \mathcal{K}$ is ε-differentially private

Truncation

- Truncation is a typical example of post-processing
- In fact, assume that the true answer is in the interval [a,b]. Then truncation can be defined as follows: If the reported is smaller than a, then it gets remapped into a, and if it is greater than b, then it gets remapped into b.
- Because of the above theorem, truncation does not decrease the level of privacy.

Plan of the lecture

- Solution of the exercise
- Brief recall of the Laplacian mechanism
- Discrete queries and Geometric Mechanism
- Truncated mechanisms
- Utility
- Optimal Mechanisms

Utility

- When a user sees the reported value z of the mechanism, he may take z as it is, or, based on his prior knowledge, he may guess another value w. We say that the user remaps z into w.
  Summarizing, we have:

- $\mathcal{X}$, the set of databases, with associated random variable X

- $\mathcal{Y}$, the set of true answers to the query f. Associated random variable Y

- $\mathcal{Z}$, the set of reported answers to the query f (after we apply the noise). Associated random variable Z

- $\mathcal{W}$, the set of guesses. Associated random variable W. $\mathcal{W}$ often coincides with $\mathcal{Y}$, but W usually does not coincide with Y.

Utility

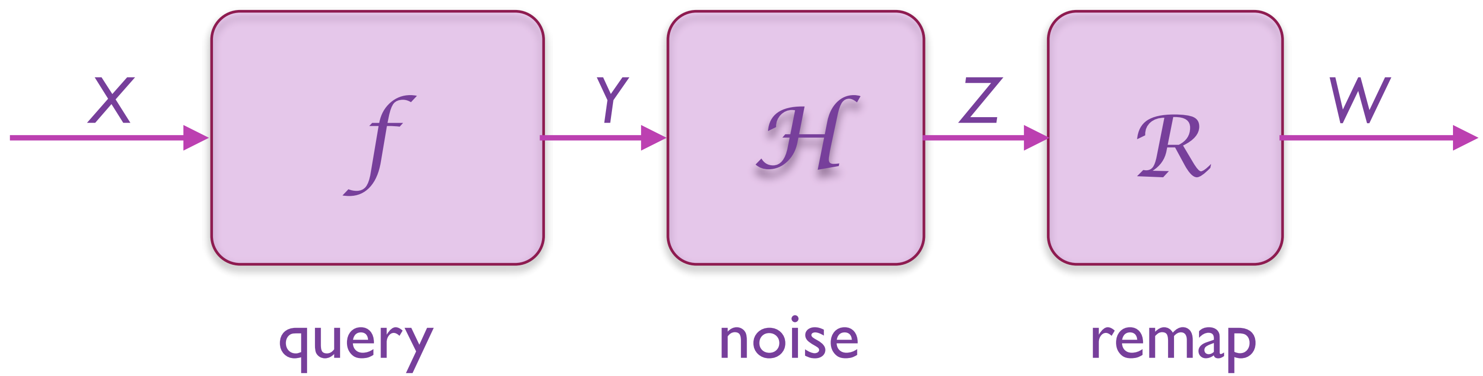
- When a user sees the reported value z of the mechanism, he may take z as it is, or, based on his prior knowledge, he may guess another value w. We say that the user remaps z into w.
  Summarizing, we have:

- $\mathcal{X}$, the set of databases, with associated random variable X

- $\mathcal{Y}$, the set of true answers to the query f. Associated random variable Y

- $\mathcal{Z}$, the set of reported answers to the query f (after we apply the noise). Associated random variable Z

- $\mathcal{W}$, the set of guesses. Associated random variable W. $\mathcal{W}$ often coincides with $\mathcal{Y}$, but W usually does not coincide with Y.



Schema for an oblivious mechanism. In a non-oblivious one Z depend also on X.

Utility

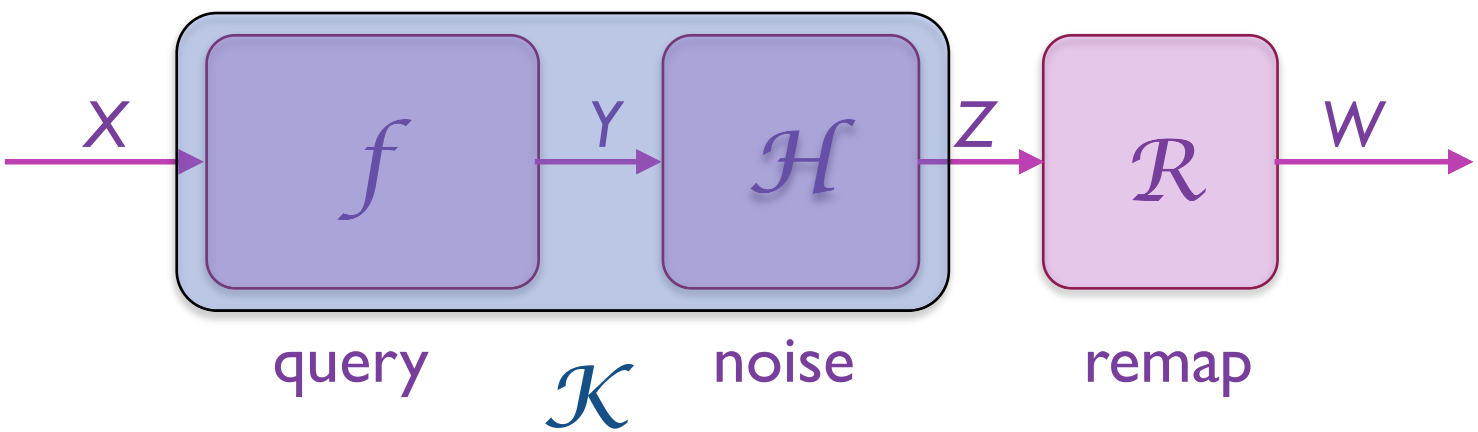
- When a user sees the reported value z of the mechanism, he may take z as it is, or, based on his prior knowledge, he may guess another value w. We say that the user remaps z into w.
  Summarizing, we have:

- $\mathcal{X}$, the set of databases, with associated random variable X

- $\mathcal{Y}$, the set of true answers to the query f. Associated random variable Y

- $\mathcal{Z}$, the set of reported answers to the query f (after we apply the noise). Associated random variable Z

- $\mathcal{W}$, the set of guesses. Associated random variable W. $\mathcal{W}$ often coincides with $\mathcal{Y}$, but W usually does not coincide with Y.



Schema for an oblivious mechanism. In a non-oblivious one Z depend also on X.

Utility

- A gain function is a function

$$g : \mathcal{W} \times \mathcal{Y} \to \mathbb{R}$$

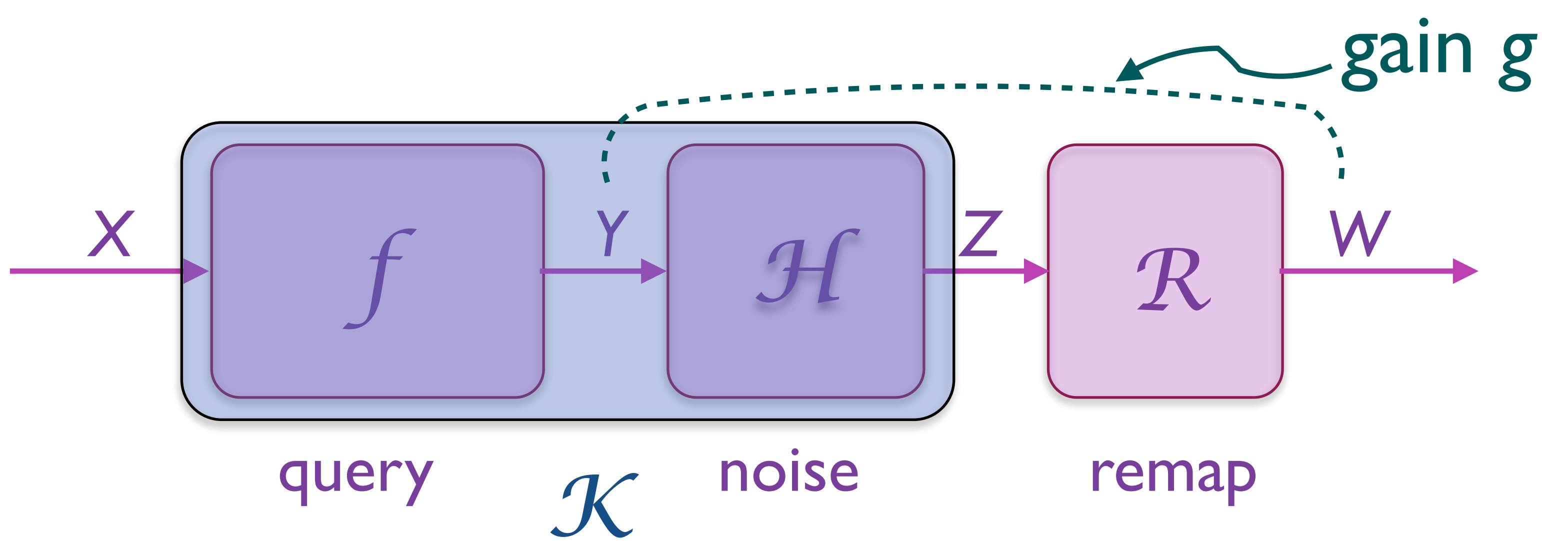
  that represents the usefulness of the guess w when the true answer is y.

- Often there is a notion of distance d between w and y, representing how well w approximates y. Formally:

$$d : \mathcal{W} \times \mathcal{Y} \to \mathbb{R}$$

- The gain g is usually assumed to be anti-monotonic with respect to d. Namely:

$$\text{if } d(w, y) \leq d(w', y), \text{ then } g(w, y) \geq g(w', y)$$



Schema for an oblivious mechanism. In a non-oblivious one Z depend also on X.

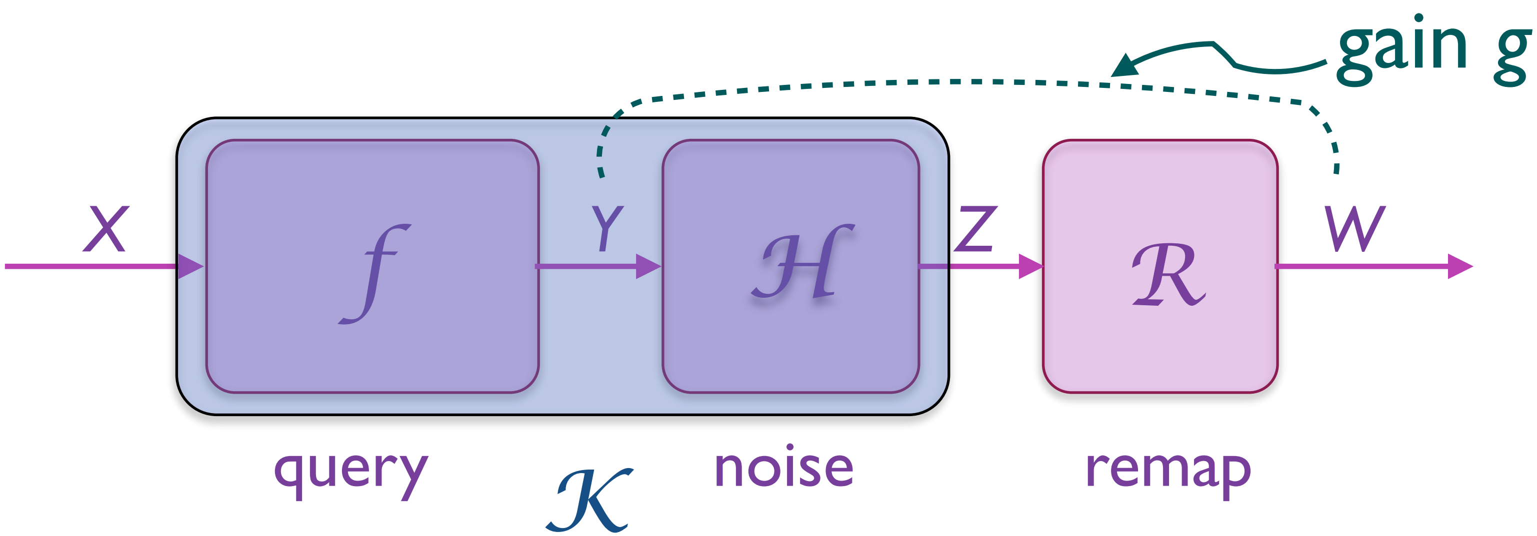
Utility

- Given a database x, consider the expected gain over all possible reported answers, for a certain remapping r. For an oblivious mechanism this is given by the formula:

$$\sum_z p_{\mathcal{H}}(z|f(x))g(r(z), f(x))$$

- For a generic (possibly non oblivious) mechanism, this is given by:

$$\sum_z p_{\mathcal{K}}(z|x)g(r(z), f(x))$$

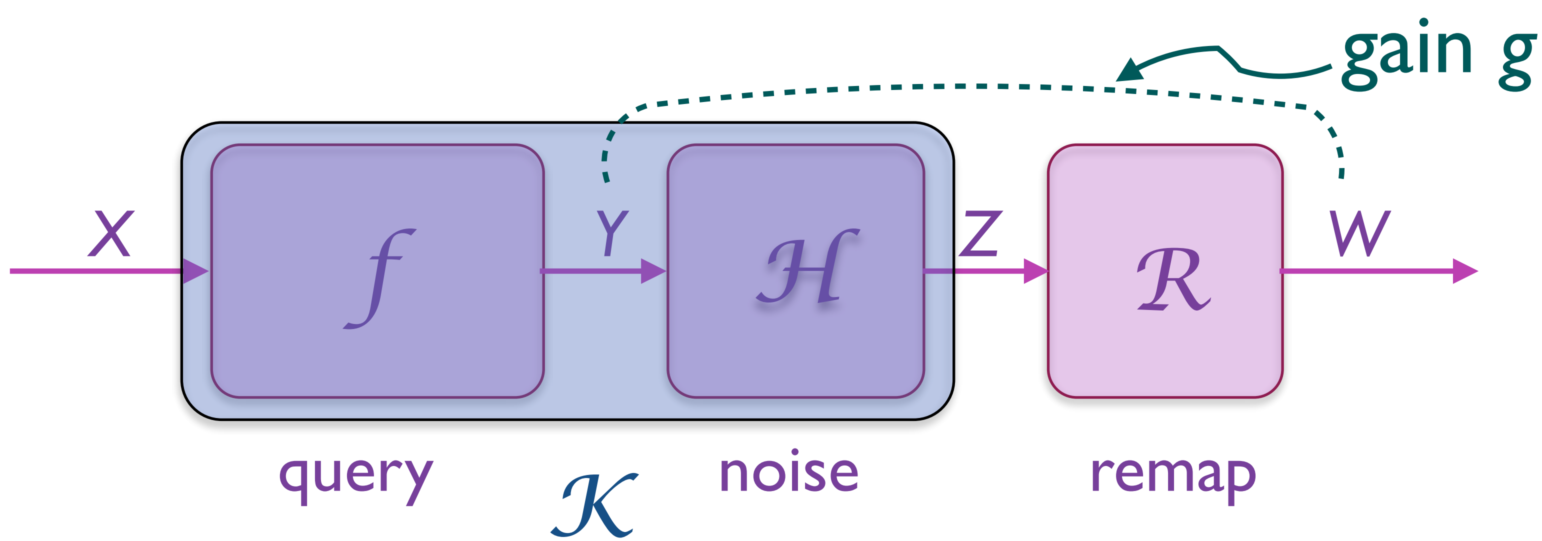


Schema for an oblivious mechanism. In a non-oblivious one *Z* depend also on *X*.

Utility

- The utility $\mathcal{U}$ of a mechanism is the maximum expected gain over all possible databases. The maximum is over all possible remappings: It is assumed that the user is rational and therefore makes the guesses that are the most useful to him. Note that $\mathcal{U}$ depends also on the prior π over $\mathcal{X}$ Formally, let us denote by r a remapping function. For an oblivious mechanism we have:

$$\mathcal{U}(\mathcal{K}, \pi, g) = \max_r \sum_x \pi(x) \sum_z p_{\mathcal{H}}(z|f(x)) g(r(z), f(x))$$



Schema for an oblivious mechanism. In a non-oblivious one Z depend also on X.

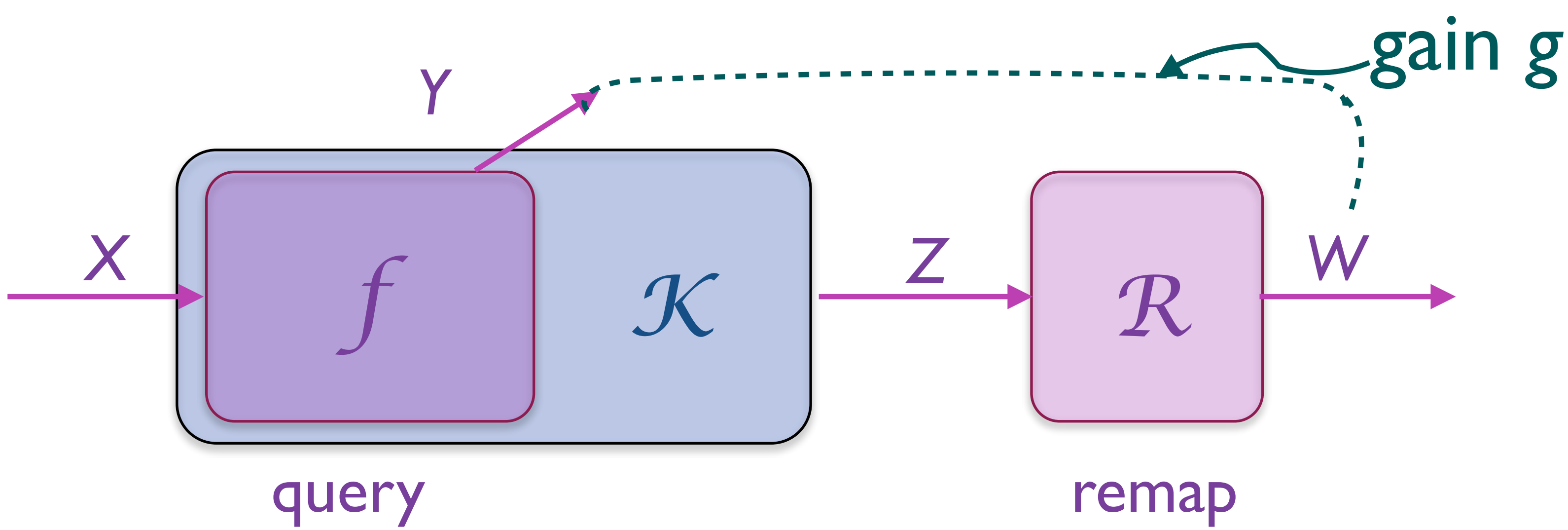
Utility

- The utility $\mathcal{U}$ of a mechanism is the maximum expected gain over all possible databases. The maximum is over all possible remappings: It is assumed that the user is rational and therefore makes the guesses that are the most useful to him. Note that $\mathcal{U}$ depends also on the prior π over $\mathcal{X}$ Formally, let us denote by r a remapping function. For an oblivious mechanism we have:

$$\mathcal{U}(\mathcal{K}, \pi, g) = \max_r \sum_x \pi(x) \sum_z p_{\mathcal{H}}(z|f(x)) g(r(z), f(x))$$

For a general (possibly non-oblivious) mechanism, we have:

$$\mathcal{U}(\mathcal{K}, \pi, g) = \max_r \sum_x \pi(x) \sum_z p_{\mathcal{K}}(z|x) g(r(z), f(x))$$



Example

The simplest gain function is the identity relation:

$$g(w,x) = \begin{cases} 1 & w = x \\ 0 & w \neq x \end{cases}$$

It represents the situation in which we are happy only if we guess the true answer.
With this gain function, the utility becomes (we give the formula for the oblivious case, the non-oblivious one is analogous):

$$\begin{aligned} \mathcal{U}(\mathcal{K},\pi,g) &= \max_r \textstyle\sum_x \pi(x) \sum_z p_{\mathcal{H}}(z|f(x))\; g(r(z), f(x)) \\ &= \max_r \textstyle\sum_y p_f(y) \sum_z p_{\mathcal{H}}(z|y)\; g(r(z), y) \\ &= \textstyle\sum_z \max_y (p_f(y)\; p_{\mathcal{H}}(z|y)) \end{aligned}$$

This utility function essentially gives the expected probability of guessing the true answer. It is the converse of the Bayes risk

Example

Another typical gain function is the converse of the distance:

$$g(w, x) = D - d(w, x)$$

where D is the maximum possible distance between reported answers and true answers (it works well for truncated mechanisms). If such maximum does not exists, we can take $D = 0$. The only problem is that we get negative gains With this gain function, the utility is the expected distance between our best guess and the true answer. It gives a measure of how good is the approximated of the true answer that we can get with the mechanism.

Plan of the lecture

- Solution of the exercise
- Brief recall of the Laplacian mechanism
- Discrete queries and Geometric Mechanism
- Truncated mechanisms
- Utility
- Optimal Mechanisms

Optimal mechanisms

- Given a prior π, and a privacy level ε, an ε-differentially private mechanism K is called optimal if it provides the best utility among all those which provide ε-differential privacy
- Note that the privacy does not depend on the prior, but the utility (in general) does.
- In the finite case the optimal mechanism can be computed with linear optimization techniques, where the variables are the conditional probabilities $p(z \mid y)$
  where y is the exact answer and z is the reported answer
- A mechanism is universally optimal if it is optimal for all priors π

Privacy vs utility: two fundamental results

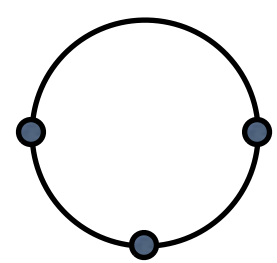
1. [Ghosh et al., STOC 2009]
   The geometric mechanism and the truncated geometric mechanism are universally optimal for counting queries and any anti-monotonic gain function

Privacy vs utility: two fundamental results

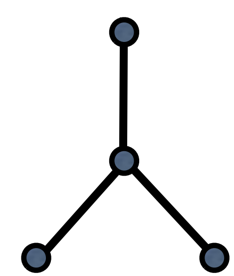
2. [Brenner and Nissim, STOC 2010] The counting queries are the only kind of queries for which a universally optimal mechanism exists

- This means that for other kind of queries one the optimal mechanism is relative to a specific user.
- The precise characterization is given in terms of the graph $(\mathcal{Y}, \sim)$ induced by $(\mathcal{X}, \sim)$

ok



not ok



not ok

Exercises

1. Define the noise density function for the Laplacian mechanism for the query "What is the percentile of the people in the DB who earn more than 10K Euro a month", assuming that the database contains at least 1000 elements.

2. Define the Laplacian truncated mechanism for the above query. Note that $\mathcal{Y}$ is the interval [0,100].

3. Prove that ε-differential privacy can be equivalently defined as follows

$\mathcal{K}$ is ε-differentially private if for every pair of databases $x_1, x_2 \in \mathcal{X}$ (not necessarily adjacents), and for every $z \in \mathcal{Z}$, we have:

$$p(Z = z | X = x_1) \leq e^{\varepsilon h(x_1, x_2)} p(Z = z | X = x_2)$$

where $h(x_1, x_2)$ represents the Hamming distance between x_1 and x_2

Exercises

4. Compute the utility of the geometric mechanism for a counting query, with privacy degree ε, on the uniform prior distribution, with the gain function defined as the identity relation.

5. Same exercise, but with the gain function defined as the converse of the distance.

6. Find a mechanism for the same counting query, with the same degree of privacy, but lower utility.

7. We saw that post-processing cannot decrease privacy. Can it decrease the utility? Motivate your answer.