Foundations of Privacy

Lecture 2

Resume of previous class

- We saw various frameworks for anonymity that have been proposed in the past, based on the notion of quasi-identifier: k-anonymity, ℓ-diversity, p-closeness

- We saw that these methods are ineffective
  - everything can be a quasi identifier
    - attacks on large sparse datasets: Netfix prize attack
  - composition attacks
    - example of combination of queries
    - general problem of deterministic methods

- Solution: **randomization**

Exercise given previous time

Bob wants to find out whether Don is affected by a certain disease d. He knows Don's age and weight, and that Don is going to check in a hospital that maintains an anonymized database of all patients, and that can be queried with queries of the form:

- How many patients are affected by the disease d ?

- What is the average age and weight of the patients affected by the disease d?

Discuss whether Bob can determine, with high probability, whether Don has the disease. What kind of background information Don needs? What kind of queries should he ask?

Randomized mechanisms

- A randomized mechanism (for a certain query) reports an answer which is an approximation of the true answer and is generated randomly according to some probability distribution
- Randomized mechanisms are more robust to combination attacks than the deterministic ones
- However, we need to choose carefully the probability distribution, in order to get the desired degree of privacy, and in order to maintain a certain degree of utility for the query
- There is a trade-off between utility and privacy, but it is not strict: for a certain degree of privacy, one mechanism can give a better utility than another. It is therefore interesting to try to find the optimal mechanism (the mechanism with highest utility), among those that offer the desired degree of privacy.
- To solve the above problem, and more in general to reason about privacy and utility, we need formal, rigorous definitions of these notions.
- A definition of privacy that has become very popular: Differential Privacy [Cynthia Dwork, ICALP 2006]

Databases

- V is a set whose elements represent all possible **values of the records** ($v \in V$ can be a tuple, i.e. it can be composed by various fields). We assume that V contains a special element $\perp$ representing a dummy record, or the absence of the corresponding record.

- A **database** of n records is an element of V^n. We will represent the databases by x, x_1, x_2, ...

- We assume a probability distribution π on the databases. We will indicate by X the corresponding random variable.

- Two databases x_1, x_2 are **adjacent** if they differ for exactly one record. We will indicate this property with the notation $x_1 \sim x_2$
    - $x_1 \sim x_2$ represent the fact that x_1 and x_2 differ for the information relative to an individual. Either this individual has been added to x_2, or he has been removed from x_2, or has changed value.

- The number of records in which two databases x_1, x_2 differ from each other is called "Hamming distance" between x_1, x_2.

Queries

- (The answer to) a query f can be seen as a function from the set of databases $\mathcal{X} = V^n$ to a set of values $\mathcal{Y}$. Namely,

$$f : \mathcal{X} \to \mathcal{Y}$$

- $y = f(x)$ is the **true answer** of the query f on the database x.
- For a given f, the distribution π on $\mathcal{X}$ also induces a distribution on $\mathcal{Y}$. We will denote by Y the random variable associated to the distribution on $\mathcal{Y}$.

Randomized mechanisms

- A randomized mechanism for the query f is any probabilistic function $\mathcal{K}$ from $\mathcal{X}$ to a set of values $\mathcal{Z}$. Namely,

$$\mathcal{K} : \mathcal{X} \rightarrow \mathcal{D}\mathcal{Z}$$

  where $\mathcal{D}\mathcal{Z}$ represents the set of probability distributions on $\mathcal{Z}$.

- $\mathcal{Z}$ does not necessarily coincide with $\mathcal{Y}$.

- z drawn from $\mathcal{D}(x)$ is a **reported answer** of the query $\mathcal{K}$ on the database x.

- Note that π and $\mathcal{K}$ induce a probability distribution also on $\mathcal{Z}$. We will denote by Z the random variable associated to this probability distribution

Differential Privacy

- We are now ready to define **Differential Privacy** for a randomized mechanism $\mathcal{K}$.

- Let us first consider the discrete case. Namely, $\mathcal{K}(x)$ is discrete, for every database x.

- **Definition (Differential Privacy)** $\mathcal{K}$ is ε-differentially private if for every pair of databases $x_1, x_2 \in \mathcal{X}$ such that $x_1 \sim x_2$, and for every $z \in Z$, we have:

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

  where $p(Z = z | X = x)$ represents the conditional probability of z given x, namely the probability that on the database x the mechanism reports the answer z

- This definition therefore means that the value (or the presence) of an individual does not affect significantly the probability of getting a certain reported value.

Properties of differential privacy

- Two important properties that have made differential privacy so successful:
  - Independence from the prior
  - Compositionality

Independence from the prior

- The distribution π on the databases is called prior, meaning: *before* the reported answer
- π represents the knowledge that a potential adversary (aka user, in the case of DP) has about the database (before knowing the answer of $\mathcal{K}$)
- We note that the definition of DP does not depend on π. This is a very good property, because it means that we can design mechanisms that satisfy DP without taking the knowledge of the adversary into account: the same mechanism will be good for all adversaries.

Compositionality

- Differential privacy is compositional, namely: given two mechanisms $\mathcal{K}_1$ and $\mathcal{K}_2$ on $\mathcal{X}$ that are respectively ε_1 and ε_2-differentially private, their composition $\mathcal{K}_1 \times \mathcal{K}_2$ is $(\varepsilon_1 + \varepsilon_2)$-differentially private.

  Note: $\mathcal{K}_1 \times \mathcal{K}_2$ is defined by the following property: if $\mathcal{K}_1(x)$ reports z_1 and $\mathcal{K}_2(x)$ reports z_2, then $(\mathcal{K}_1 \times \mathcal{K}_2)(x)$ reports (z_1, z_2).

  Proof: exercise

- **Privacy budget**: An user is given an initial budget α. Each time he asks a query, answered by ε-differentially private mechanism, his budget is decreased by ε. When his budget is exhausted, he is not allowed to ask queries anymore.

Bayesian interpretation

- Let X_i be the random variable representing the value of the individual i, and let X_{others} be the random variable representing the value of all the other individuals in the database.
  Similarly, let x_i and x_{others} represent possible values for X_i and X_{others}.
  Note that (x_i, x_{others}) represents and element in $\mathcal{X}$.
  Analogously, let π_i represent the component of the prior distribution that concerns the value of the individual i.

- ε-differential privacy in the discrete case is equivalently characterized by the following property: For all $(x_i, x_{others}) \in \mathcal{X}$, for all $z \in Z$, and for all π_i,

$$p(X_i = x_i | X_{others} = x_{others}, Z = z) \leq e^{\varepsilon} p(X_i = x_i | X_{others} = x_{others})$$

  Namely: assuming that the adversary knows the value of all the other individuals in the database, the reported answer does not increase significantly his probabilistic knowledge of the value of i, with respect to his prior knowledge

  Note: $p(X_i = x_i | X_{others} = x_{others})$ is called *prior* of x_i, and $p(X_i = x_i | X_{others} = x_{others}, Z = z)$ is called *posterior* of x_i.

Differential Privacy

- Let us now consider the continuous case. Namely, $\mathcal{K}(x)$ is a probability density function on $\mathcal{Z}$. The only thing that changes is that we consider a measurable subset $\mathcal{S}$ of $\mathcal{Z}$ instead than a single z:

- **Definition (Differential Privacy)** $\mathcal{K}$ is ε-differentially private if for every pair of databases $x_1, x_2 \in \mathcal{X}$ such that $x_1 \sim x_2$, and for every measurable $\mathcal{S} \subseteq Z$, we have:

  $$p(Z \in \mathcal{S} | X = x_1) \leq e^{\varepsilon} p(Z \in \mathcal{S} | X = x_2)$$

  where $p(Z \in \mathcal{S} | X = x)$ represents the probability that on the database x the mechanism reports an answer in $\mathcal{S}$

- This definition therefore means that the value (or the presence) of an individual does not affect significantly the probability that the reported value satisfy a certain property.

Examples of mechanisms

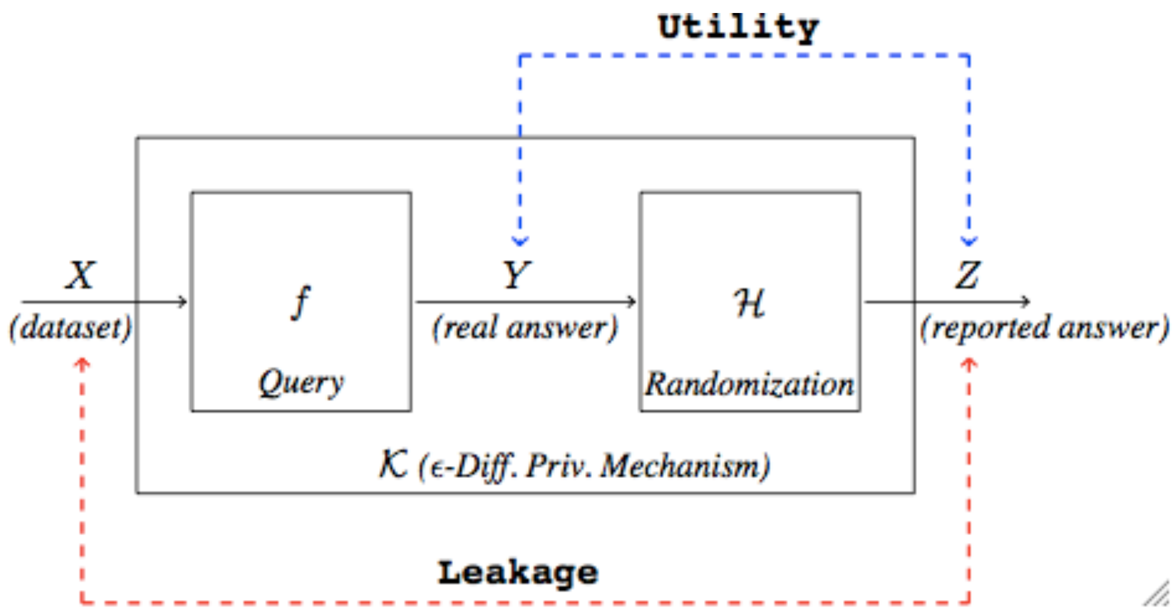
Let us assume that we have databases containing as values V the heights of people, in cm, ranging from 50 to 250 (integers). Let us assume that the query is: the average age of the people in the data base, rounded to the next integer.

- The mechanism that always reports the true answer is not differentially private, for any ε
- The mechanism that always reports 150 is differentially private in the strongest sense ($\varepsilon = 1$), but totally useless
- The mechanism that reports 100 if the true answer is less than 150, and 200 otherwise, is a bit more useful, but it is not differentially private, for any ε
- The mechanism that reports the true answer with probability $\varepsilon/(200 + \varepsilon)$, and every other integer in $[50,250]$ with probability $1/(200 + \varepsilon)$, is ε-differentially private, and, intuitively, relatively useful. We will study its utility later on.

Oblivious Mechanisms

- Given $f : \mathcal{X} \rightarrow \mathcal{Y}$ and $\mathcal{K} : \mathcal{X} \rightarrow \mathcal{Z}$, we say that $\mathcal{K}$ is oblivious if it depends only on $\mathcal{Y}$ (not on $\mathcal{X}$)

- If $\mathcal{K}$ is oblivious, it can be seen as the composition of f and a randomized mechanism $\mathcal{H}$ (noise) defined on the exact answers $\mathcal{K} = f \times \mathcal{H}$



- Privacy concerns the information flow between the databases and the reported answers, while utility concerns the information flow between the correct answer and the reported answer

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:

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

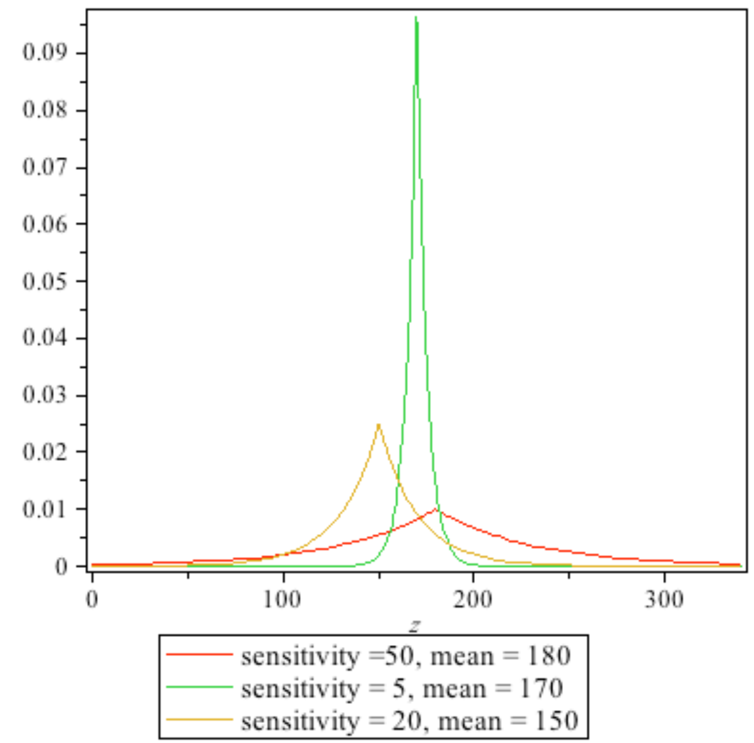
where Δf is the *sensitivity* of f:

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

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

and c is a normalization factor:

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



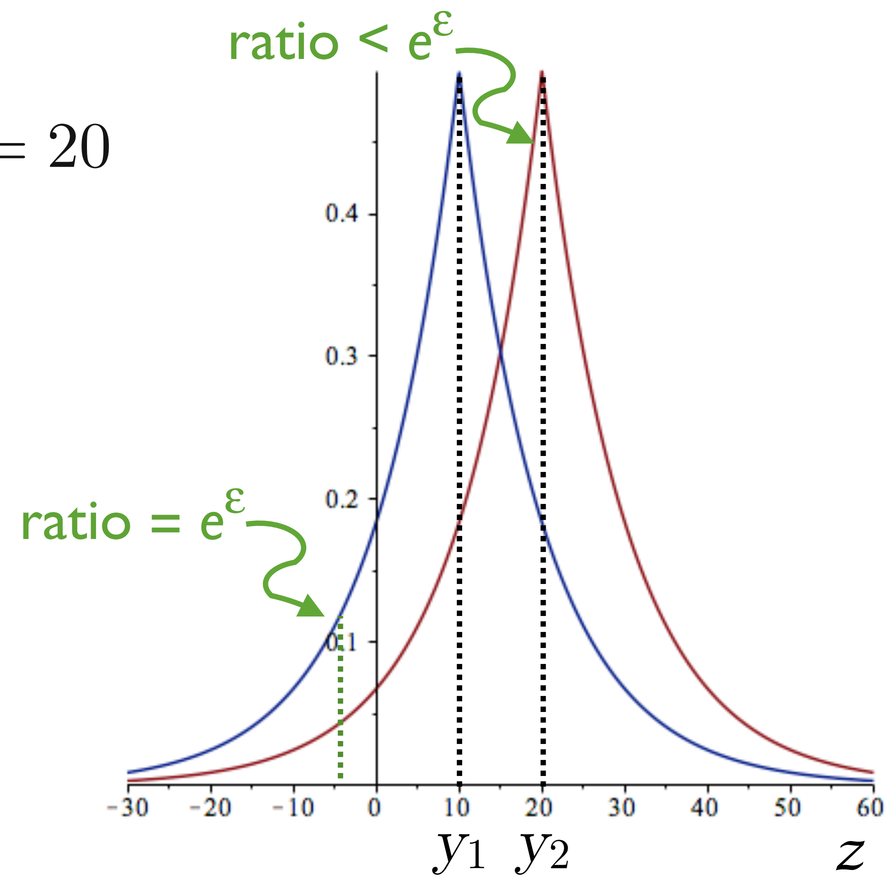
Sensitivity of the query

- The sensitivity of the query and the level of privacy ε determine the amount of noise of the mechanism:
  - higher sensitivity ⇒ more noise
  - smaller ε ⇒ more privacy, more noise
- Intuitively, the more the mechanism is noisy, the less useful it is (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.

Example of Laplacian Mechanism

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

  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 $[y_1, y_2]$

Laplacian mechanism

The probability density function of a Laplacian mechanism is:

$$p(Z = z|X = x) = dP_{f(x)}(z) = c\,e^{-\frac{|z-f(x)|}{\Delta f}\varepsilon}$$

where $c = \dfrac{\varepsilon}{2\,\Delta f}$

Theorem: The Laplacian mechanism is ε–differentially private

Proof: Let $x_1 \sim x_2$ and $y_1 = f(x_1), y_2 = f(x_2)$ We have:

$$\begin{aligned}
\frac{p(Z=z|X=x_1)}{p(Z=z|X=x_2)} &= \frac{c\,e^{-\frac{|z-f(x_1)|}{\Delta f}\varepsilon}}{c\,e^{-\frac{|z-f(x_2)|}{\Delta f}\varepsilon}} \\
&= e^{\frac{|z-y_2|}{\Delta f}\varepsilon - \frac{|z-y_1|}{\Delta f}\varepsilon} \\
&\leq e^{\frac{|y_1-y_2|}{\Delta f}\varepsilon} \\
&\leq e^{\varepsilon}
\end{aligned}$$

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

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 ? "