ISYE 8813: Algorithmic Foundations of Ethical ML **Fall 2021**

Lecture 3: Definition of Differential Privacy

Lecturer: Juba Ziani *Scribe: Juba Ziani*

1 Preliminaries

Basic notations:

- Database x.
- n individuals in the database
- Data universe is $\mathcal{X}$.

How do we think about a database?

- Formally, a database x is a vector in $\mathcal{X}^n$. I.e., think of x_i representing the data of the i-th persone in the database. For example:

$$\begin{pmatrix} Ash & M & - \\ Summer & F & + \\ \vdots & \vdots & \vdots \\ Homer & M & - \end{pmatrix}$$

  Here, $\mathcal{X} = S \times \{M, F\} \times 0, 1$ (Cartesian product), where S is the set of possible names. You can think of $-$ and $+$ as a label that indicates whether an individual has a specific attribute. For example, covid vaccination setting: $+$ can mean vaccinated, $-$ non-vaccinated.

- Alternative representation: histogram form. The histogram form records how many of each data type are present in the database. For example, if $\mathcal{X} = +, -$, and the data is given by

$$\begin{pmatrix} Ash & - \\ Summer & + \\ Homer & - \end{pmatrix},$$

  then the histogram form is $x = (1, 2)$, where x_1 records the number of $+$'s and x_2 records the number of $-$'s. Note that in histogram form, the database x lives in $\mathbb{N}^{|\mathcal{X}|}$: the vector has one entry for each possible element in $\mathcal{X}$, and each entry is an integer in $\mathbb{N}$.

Remark 1 (Multiset of rows VS histogram form)**.** *We will often use the histogram notation in this course, as it is more mathematically convenient: indeed, it allows us to reason about databases comprised of integer values, as opposed to thinking about data from a possibly complicated universe $\mathcal{X}$ (think of $\mathcal{X}$ being the Cartesian product of many different features or attributes). However, the multiset of rows is the more natural and intuitive interpretation of what a real dataset looks like; the multiset representation is often the one that is used in practical implementations. Generally, the multiset representation is much more compact than the histogram one: you can represent such a database as a table with n rows and $|\mathcal{X}|$ columns (size $n\|\mathcal{X}|$), versus the histogram representation has size $\sim n^{|\mathcal{X}|}$.*

2 Neighboring databases

As discussed earlier in this lecture, differential privacy measures and bounds how much outcomes change across two close-by, **neighboring** databases, that only change in the data of one individual. To introduce DP formally, we first need to introduce the formal notion of neighboring databases. To do so, we need a metric/distance function.

Definition 1 (ℓ_1-norm)**.** *The ℓ_1-norm of a database $x \in \mathbb{N}^{|\mathcal{X}|}$ (in histogram form) is denoted $\|x\|_1$, and is given by*

$$\|x\|_1 = \sum_{i=1}^{|\mathcal{X}|} x_i.$$

We can now measure the distance between two databases x and y, as $\|x - y\|_1$. We can then formally introduce the notion of neighboring databases:

Definition 2 (Neighboring databases)**.** *Two databases x and y are neighboring if and only if*

$$\|x - y\|_1 \leq 1.$$

Informally, two databases are neighboring (in histogram notation) when they differ by at most 1 record. For example, $(2, 3)$ and $(2, 2)$ are neighboring: they correspond to databases (in multiset form) of the form $(+ + - - -)$ and $(+ + - -)$, which only differ in the addition or deletion of one $-$ entry.

Remark 2 (Neighboring databases in multiset form)**.** *In multiset form, we generally say that two databases are neighboring if they differ by at most 1 row. For example,*

$$\begin{pmatrix} Ash & M & - \\ Summer & F & + \\ Homer & M & - \end{pmatrix},$$

and

$$\begin{pmatrix} Juba & M & + \\ Summer & F & + \\ Homer & M & - \end{pmatrix},$$

are neighboring.

3 Differential Privacy, formally

As we have seen earlier, the idea behind differential privacy is that a mechanism that computes a statistic on two databases that are close to each other (in the sense that they are neighboring, as defined formally above) should have almost indistinguishable outcomes across both databases. This means that a single agent's data has very little effect on and is minimally encoded in the outcome of the mechanism, hence the outcome of the mechanism provides very little insights onto what a single agent's data was. Differential privacy formally quantifies what it means for outcomes of a mechanism to be close. I.e.,

Definition 3 (Differentially Private Mechanism). *A randomized mechanism $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow Range(\mathcal{M})$ is ε-differentially private if and only for all outcomes $S \in Range(\mathcal{M})$, and for all neighboring databases $(x, y) \in \mathbb{N}^{|\mathcal{X}|}$,*

$$Pr(\mathcal{M}(x) \in S) \leq e^{\varepsilon} \Pr(\mathcal{M}(y) \in S),$$

where the probabilities are taken of the randomness of the mechanism $\mathcal{M}$.

There is a lot to unpack here, so let us go through the definition more carefully:

- $\mathcal{M}(x)$ is the outcome of the mechanism on database x, and $\mathcal{M}(y)$ the outcome of the mechanism on database y.
- S can be thought of a bad event or outcome. We think of S as a bad event in the sense that when the outcome is in S, we are worried that we can distinguish between x and y, and learn what a single agent's data was.
- We aim to prevent such bad events by enforcing that the probability of having an outcome in S is *almost* the same when running $\mathcal{M}$ on x and on y. Then, it is hard to distinguish between x and y: the outcome event S was almost equally likely to come from those two databases.
- We control how close those two probabilities $Pr(\mathcal{M}(x) \in S)$ and $\Pr(\mathcal{M}(y) \in S)$ are to each other through the e^{ε} term (we will get back to the meaning of ε later in this lecture).
- The definition is symmetric: you can invert the roles of x and y. So for every x and y that neighbor each other, it also tells you that

  $$Pr(\mathcal{M}(y) \in S) \leq e^{\varepsilon} \Pr(\mathcal{M}(x) \in S).$$

  So in the end you get a bound that goes both ways: $\Pr(\mathcal{M}(x) \in S)$ cannot be much bigger than $\Pr(\mathcal{M}(x) \in S)$, but also $\Pr(\mathcal{M}(y) \in S)$ cannot be much bigger than $\Pr(\mathcal{M}(y) \in S)$. Therefore they have to be close to each other.

Further, we note that this is a **worst-case** definition. It must holds for the worst-case/over **all** possible outcomes S, and for the worst-case/over **all** possible databases x and y with $\|x - y\|_1 \leq 1$. Why do we take such a worst-case/what happens otherwise?

1. If the definition did not hold for all x and y, there can be databases for which some of the agents are not provided any privacy guarantees.

2. If the definition did not hold for all S, there could be some very small probability that the outcome of the mechanism is in S for database x, but this probability could be very different and much larger for outcome y. Now if I see that the outcome of my mechanism is in set S, I know that there is a very good chance the original database was y/I can distinguish between them.

What role does ε play? As mentioned above, the definition depends on a parameter ε. This parameter is what controls the *level* or *amount* of privacy that we are guaranteeing. The way to think about this parameter is the following:

- When $\epsilon = 0$, the definition requires that $Pr(\mathcal{M}(x) \in S) \leq e^{\varepsilon} \Pr(\mathcal{M}(y) \in S) = \Pr(\mathcal{M}(y) \in S)$. But by symmetry, it also requires that $Pr(\mathcal{M}(y) \in S) \leq \Pr(\mathcal{M}(x) \in S)$. Hence, the definition requires that for all neighboring databases x and y, and outcome sets S,
$$\Pr(\mathcal{M}(x) \in S) = \Pr(\mathcal{M}(y) \in S).$$
I.e., the distribution of outcomes of the mechanisms are the same for **all** databases. In other words, the outcome of the mechanism **does not depend on the data itself**, which guarantees perfect privacy. However, since you are not using the data at all, your accuracy is infinitely bad/you cannot learn anything useful. **More privacy means less accuracy.**

- When $\varepsilon \to +\infty$, we also have that the e^{ε} term grows to infinity. Therefore, the constraint $Pr(\mathcal{M}(x) \in S) \leq e^{\varepsilon} \Pr(\mathcal{M}(y) \in S)$ becomes very trivial/is always satisfied. I.e., our mechanism is completely unconstrained and we have no privacy protection; the distributions of outcomes over two neighboring databases x and y can be vastly different.

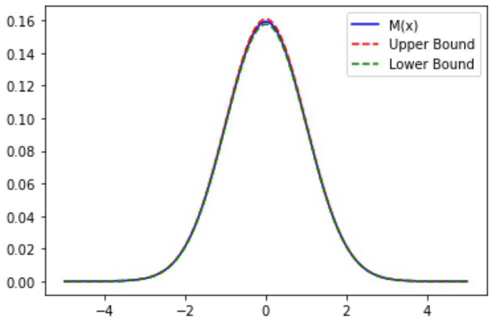
- The level of privacy continuously evolves between $\varepsilon = 0$ and $\varepsilon = +\infty$. As ε becomes bigger and bigger, our privacy guarantee becomes weaker and weaker. See Figure 1 to see how the choice of ϵ affects the privacy constraints.

- When ε is small, we remark that $\exp(\varepsilon) \sim 1 + \varepsilon$. This regime of small values of ε is the one in which we preferably want ε to be. Outside of this regime, $\exp(\varepsilon)$ becomes much larger than $1 + \varepsilon$, and the privacy guarantee we obtain is too slack.

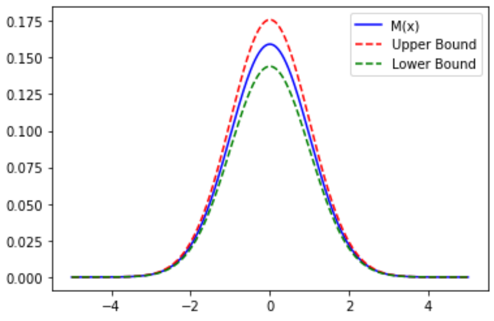
This definition of differential privacy can be made less stringent by relaxing it by an additive factor:

Definition 4 (Approximate Differential Privacy). *A randomized mechanism $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to Range(\mathcal{M})$ is (ε, δ)-differentially private if and only for all outcomes $S \in Range(\mathcal{M})$, and for all neighboring databases $(x, y) \in \mathbb{N}^{|\mathcal{X}|}$,*

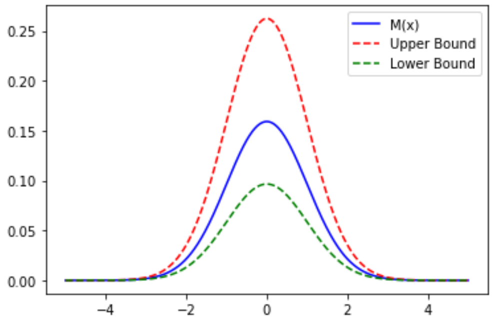
$$Pr(\mathcal{M}(x) \in S) \leq e^{\varepsilon} \Pr(\mathcal{M}(y) \in S) + \delta, \tag{1}$$



(a) $\varepsilon = 0.01$



(b) $\varepsilon = 0.1$



(c) $\varepsilon = 0.5$

Figure 1: Lower and upper bounds on the distribution of outcomes for $M(y)$ for ε-DP, where y neighbors x

where the probabilities are taken of the randomness of the mechanism $\mathcal{M}$.

Note here that we think of δ as a very small parameter, generally $\delta << 1/n$. (For $\delta \sim 1/n$, we are in the case where the trivial mechanism that outputs an element in the database uniformly at random is $(0, \delta)$-DP. This is bad because it is obviously not privacy-preserving: it reveals a full entry in the database.)

- The major difference with approximate DP is that it allows the mechanism designer to relax the privacy guarantee for low probability events. E.g., imagine that $Pr(\mathcal{M}(x) \in S) \leq \delta$ on the current database. Then Equation (1) is satisfied; i.e., we do not need to protect database x on an event S that happens with very low probability.
- This also allows for situations where $Pr(\mathcal{M}(x) \in S) = 0$ but $Pr(\mathcal{M}(y) \in S) = \delta$. Then, in the very low probability case where we see an outcome in S, we can distinguish x and y: we know that seeing S was impossible on database x, so the database can be y. But this happens with very low probability δ/almost never.
- In the first problem set, we will aim to understand a bit more carefully what this δ means from a probabilistic point of view.