Privacy Engineering (CO408)

Week 4 - Differential Privacy

1 Some definitions

Let's first recall some important definitions. Note: for the sake of simplicity and brevity, we won't be completely formal. If you are interested in a fully rigorous description, you can read Chapter 2 and 3 of *The Algorithmic Foundations of Differential Privacy* by C. Dwork and A. Roth (download here).

Definition. A *dataset* is a table in which every row (or record) contains some attributes relative to a specific user. All rows correspond to different users. We have seen several such tables in the previous classes, but here we usually don't have a column for IDs. We call $\mathcal{D}$ the family of all datasets.

Definition. We say that two different datasets D_1 and D_2 are *neighboring* if they differ by exactly one row, which means $D_1 = D_2 \cup \{r\}$ or $D_2 = D_1 \cup \{r\}$ (where r is some row). Observe that two neighboring datasets always have different size (and, specifically, $|D_1| - |D_2| = \pm 1$).

2 Sensitivity (univariate)

Definition. Let $f \colon \mathcal{D} \to \mathbb{R}$ be a function. The *global sensitivity* of f is

$$\Delta f = \max_{D_1, D_2} |f(D_1) - f(D_2)|,$$

where D_1 and D_2 can be **any** arbitrary **neighboring** datasets in $\mathcal{D}$.

In real world scenarios, f will be a *query* that a data analyst sends to the data curator of a certain dataset. For example, $f(D)$ could be the query "How many users in D are males?". The global 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. However, observe that the global sensitivity does **not** depend on the specific dataset D. The global sensitivity is computed over **all** possible datasets in $\mathcal{D}$. This is why it's called *global* sensitivity.

For example, suppose you have a dataset D with one positive integer per user (for space reasons, we write D as a vector rather than a column):

$$D = (4, 16, 3, 1, 1, 1, 9, 5, 3, 23).$$

Define $f(D) := \min D$, so that $f(D) = 1$. Observe that adding or removing one record (i.e. element) from D will not change the value of $f(D)$ at all. However, it is **not true** that $\Delta f = 0$. The value of Δf must be computed over *all* datasets.

Exercise 2.1. Let f be the function that computes the minimum element in a dataset of positive integers. Show that $\Delta f = \infty$. That is, the global sensitivity of f is unbounded.

Solution. Your answer here.

In the class we have seen that, for any $f\colon \mathcal{D} \to \mathbb{R}$, the mechanism $M\colon \mathcal{D} \to \mathbb{R}$ defined by

$$M(D) = f(D) + \mathrm{Lap}(\Delta f/\varepsilon)$$

is ε-DP. This is called the *Laplace mechanism*. However, when Δf is unbounded (or very large), we cannot apply the simple Laplace mechanism without destroying utility. Fortunately, in many cases it is still possible to use other mechanisms. However, these mechanisms are often very sophisticated, and are beyond the scope of this course.

3 Sensitivity (multivariate)

In the class we have already seen an example of how to optimise the Laplace mechanism for histograms, in order to achieve much better utility. The main idea was that adding/removing a user can affect the count of only one bucket. In this exercise we will see a generalized version of this optimization.

So far we have considered only univariate functions, i.e. functions taking values in $\mathbb{R}$. Now we want to deal with multivariate functions, i.e. functions taking values in $\mathbb{R}^k$. These functions have the form

$$f(D) = (f_1(D), f_2(D), \ldots, f_k(D)).$$

Intuitively, we can see f as a list of different queries $f_1, \ldots, f_k$.

Definition. We can generalize the notion of global sensitivity to a list of queries. Consider any multivariate function $f\colon \mathcal{D} \to \mathbb{R}^k$, with $f(D) = (f_1(D), f_2(D), \ldots, f_k(D))$. Then we generalize the definition of global sensitivity as follows:

$$\Delta f = \max ||f(D_1) - f(D_2)||_1 = \max \sum_{i=1}^{k} |f_i(D_1) - f_i(D_2)|.$$

where D_1 and D_2 can be **any** arbitrary **neighboring** datasets in $\mathcal{D}$.

The Laplace mechanism can be generalized immediately to multivariate functions:

Theorem. Take a multivariate function $f\colon \mathcal{D} \to \mathbb{R}^k$. Consider the mechanism $M\colon \mathcal{D} \to \mathbb{R}^k$ defined by

$$M(D) = (f_1(D) + \mathrm{Lap}(\Delta f/\varepsilon), \ldots, f_k(D) + \mathrm{Lap}(\Delta f/\varepsilon).$$

The mechanism M is ε-DP.

Observe that we need to add noise proportional to $\Delta f/\varepsilon$ to *each* entry.

This generalized mechanism is very powerful, as it offers a simple way to preserve ε-DP when we answer many queries. For example, suppose that $f_1, \ldots, f_k$ is a list of counts, so that each f_i has global sensitivity 1. Then it's immediate to show that $\Delta(f_1, \ldots, f_k) \leq \sum_{i=1}^{k} \Delta f_i = \sum_{i=1}^{k} 1 = k$. So, using this generalized Laplace mechanism to release $(f_1, \ldots, f_k)$ with ε-DP, it's sufficient to add $\mathrm{Lap}(k/\varepsilon)$ noise to each f_i.

Observe that this is exactly what we could have done using the composability theorem to split evenly the total privacy budget ε across k different queries! Indeed, we would have added noise $\mathrm{Lap}(1/\frac{\varepsilon}{k})$, which is precisely what the generalized Laplace mechanism tells us to do.

But sometimes the sensitivity of the vector $\Delta(f_1, \ldots, f_k)$ is much lower than the sum of the sensitivities for each entry! In these cases, the Laplace mechanism for multivariate functions achieves much better utility than a simple application of the composability theorem (i.e. distributing the budget). We will now see some examples.

Exercise 3.1. Consider a survey with the following 3 questions:

- Have you read a Harry Potter book (Yes/No)?

- How many books in the Harry Potter series do you own (0 to 7)?
- What is your gender (Male, Female)?

So we our datasets looks like this:

read_HP	**owned_HP_books**	**gender**
Yes	3	Male
No	1	Female
Yes	5	Female
⋮	⋮	⋮

State and briefly explain the global sensitivity for each of the following differentially private queries on the collected Harry Potter dataset.

i) How many people have read a Harry Potter book?

ii) How many males and how many females are in the dataset (one query with a tuple result)?

iii) A query consisting of both of the previous queries i.e. both (i) and (ii).

iv) A query consisting of the following 4 queries. How many males have read a Harry Potter book? How many males have not read a Harry Potter book? How many females have read a Harry Potter book? How many females have not read a Harry Potter book?

v) The total number of Harry Potter books owned?

vi) The average number of books owned?

Solution. Your answer here.

Exercise 3.2 In the class we have seen an optimised mechanism for histograms. Specifically, we saw that we can release a differentially private histogram by adding noise proportional to $1/\varepsilon$ to the count of each bin. Prove formally that this method provides ε-DP.

Solution. Your answer here.

4 Definition of differential privacy (again)

In the class we have seen this definition of DP:

Definition. Let $M : \mathcal{D} \to \mathbb{R}$ be a randomized mechanism. M is ε-differentially private if, for any neighboring datasets $D, D' \in \mathcal{D}$ and any $y \in \mathbb{R}$, we have:

$$\Pr[M(D) = y] \leq e^{\varepsilon} \Pr[M(D') = y].$$

This is actually a simplified definition. The true definition is different in two ways:

- In the simplified definition, M takes values in $\mathbb{R}$ (the real numbers), but actually it could take values in any set Y. The definition doesn't change at all in this case (we just replace $\mathbb{R}$ with Y). We have already seen an example of this when we considered mechanisms with values in $\mathbb{R}^k$. It is convenient to think of Y generically as range(M), i.e. the set of all possible outputs of M.
- The inequality between probabilities as presented so far works well when range(M) is discrete (e.g. finite), but it doesn't work anymore when the range of M is continuous (like in the Laplace mechanism). In this case, we must replace "$M(D) = y$" with "$M(D) \in S$", where S is a *subset* of range(M) (not an element).

The general definition is:

Definition. Let $M : \mathcal{D} \to Y$ be a randomized mechanism. M is ε-differentially private if, for any neighboring datasets $D, D' \in \mathcal{D}$ and any $S \subset \text{range}(M)$, we have:

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \Pr[M(D') \in S].$$

This definition works well for mechanisms that take values in any set, both discrete and continuous.

To get an idea of what this means, you can consider the Laplace mechanism for a function f and replace S by any interval, e.g. $(4.7, 11.2)$. Then the definition ensures that:

$$\Pr[4.7 < f(D) + \text{Lap}(\Delta f/\varepsilon) < 11.2] \leq e^{\varepsilon} \Pr[4.7 < f(D') + \text{Lap}(\Delta f/\varepsilon) < 11.2].$$

5 Post-processing

Suppose you are the data curator, and an analyst sends you a counting query. You want to make sure that the output is differentially private, so you use the Laplace mechanism to answer the query. However, the Laplace mechanism adds noise according to a continuous Laplace distribution. This means that the output of the mechanism is not necessarily an integer, but can be any real number, such as 4.719. The analyst doesn't know anything about DP, so you are afraid that he could get a bit confused if he received a float output for a counting query. You would like to round the differentially private output to the nearest integer. But are you sure that the result will still be differentially private?

Intuitively, if you have some private information and you modify it *without looking again at the original data*, then the modified information must stay *at least as private*.

Within the framework of DP, this intuition can actually be formalized with a theorem.

Theorem (Post-processing). Let $M\colon \mathcal{D} \to Y$ be an ε-DP mechanism. Let $g\colon Y \to Z$ be a deterministic function. Then $g \circ M$ is an ε-DP mechanism.

Exercise 5.1. Prove the Post-processing theorem.

Solution. Your answer here.

Exercise 5.2 Prove that rounding any output of an ε-DP mechanism preserves ε-DP.

Solution. Your answer here.

6 The actual guarantees of DP

Unlike other definitions of privacy such as k-anonymity, for DP it is harder to understand what are the actual guarantees provided in practice. In other words, what are the *semantics* of the formal DP definition?

This question can be answered in many ways. Here is a neat and simple answer.

Suppose that you compile a survey about satisfaction at workplace. You are told that your boss will be able to send queries about the resulting dataset, but only through a differentially private mechanism M with total privacy budget ε. Depending on the result of M, your boss will make some decisions. For example, if he learns that most employees hate him, he will decide to fire everybody. We can model the boss' decisions as a function of the outputs of M, that is $g\colon \text{range}(M) \to Z$. Here, Z is the space of all possible decisions. An element of Z could be anything, for example "Fire everybody" or "Give you a promotion" or "Buy a new coffee machine".

Now, differential privacy guarantees that *any* decision made by the boss would have *probably* been the same even if you didn't participate in the survey. In other words, your participation in the dataset is unlikely (as quantified by ε) to affect you in any way.

Formally, this is an immediate consequence of the post-processing theorem. Let D be the dataset with your record and D' the same dataset without your record. Let $S \subseteq Z$ be a set of decisions. S could contain only one decision, e.g. $S = \{\text{"Fire everybody"}\}$, or more. By the theorem we have that $g \circ M$ is ε-DP, which means

$$\Pr[g \circ M(D) \in S] \leq e^{\varepsilon} \Pr[g \circ M(D') \in S].$$

If $S = \{\text{"Fire everybody"}\}$ and ε is small enough, this means that:

$$\Pr[\text{You get fired} \mid \text{You're in the dataset}] \approx \Pr[\text{You get fired} \mid \text{You're not in the dataset}].$$

Note that this observation, just like everything in the context of differential privacy, does not depend in any way on the auxiliary information known to the boss (or anybody). That is, the guarantees of DP hold always in the same way, *no matter what auxiliary information is available.*

However, observe that DP does NOT guarantee that any decision made by the boss would have probably been the same even *if the boss didn't have access to the differentially private mechanism.* For example, the mechanism might allow the boss to learn that *most* of the employes in the HR department do not like him. At this point the angry boss might decide to fire everybody in the HR department. And if you are in the HR department, you will get fired as well, even if you did not participate in the survey (or you answered that you love your boss). In other words, even DP does NOT

ensure that these two probabilities are similar:

$$\Pr[\text{You get fired} \mid \text{We don't release any outputs}] \approx \Pr[\text{You get fired} \mid \text{We release DP outputs}].$$

Make sure that you compare this (non) equation with the previous one and understand the difference.

This is just like the "smoking causes cancer" example that we've seen in the class. DP cannot guarantee that nobody will learn from a DP mechanism that smoking is heavily correlated with having cancer. On the contrary, we *want* DP to allow this! DP simply guarantees that releasing these outputs will affect you in the same way whether you are in the data or not.

Exercise 6.1 (optional) Consider again the example with the workplace satisfaction survey. In the set of all possible decisions Z, you could have also the boss' decision(s) to increase/decrease your salary by a certain amount. We model this as a *utility* function $\text{sal}_\$ \colon Z \to [0, 10000]$. For each decision $z \in Z$, the value $\text{sal}_\$(z)$ is your new salary. Unfortunately, DP cannot guarantee that your salary will remain roughly the same as *before*. However, we can prove that the expected salary will be roughly the same whether you participate in the survey or not. How?

Solution. Your answer here.