CS 860: Algorithms for Private Data Analysis Fall 2020

Lecture 4 — Intro to Differential Privacy, Part 2

Prof. Gautam Kamath *Scribe: Gautam Kamath*

Today, we continue with some of the core fundamentals of differential privacy. We start by presenting arguably the most important algorithm in differential privacy: the Laplace mechanism.

Laplace Mechanism

Content in this section is based heavily off of Section 3.3 of [DR14].

Last time, we saw our first differentially private algorithm: randomized response. At its core, this is useful for privatizing the value of a single bit: whether an individual's private data is 0 or 1 (though it can be generalized to categorical data). While the privatized result can be used for whatever other query we wish to answer, this is indirect and often lossy. Our first focus today, the *Laplace mechanism*, will directly address any sort of numeric query. Before we introduce the algorithm itself, we will require the important concept of the *sensitivity* of a function (in particular, the ℓ_1 sensitivity).

Definition 1. *Let $f : \mathcal{X}^n \to \mathbb{R}^k$. The ℓ_1-sensitivity of f is*

$$\Delta^{(f)} = \max_{X,X'} \|f(X) - f(X')\|_1,$$

where X and X' are neighbouring databases.

When the function we are discussing is clear from context, we will drop f and just use Δ for the ℓ_1-sensitivity.

The sensitivity is a rather natural quantity to consider in the context of differential privacy. Indeed, recall that differential privacy attemps to mask the contributions of any one individual. Upper bounding "how much" the function can change by modifying a single datum is thus well motivated intuitively, and we will see how we exploit it technically. I put "how much" in quotes, since it may seem mysterious why we consider the ℓ_1-sensitivity of the function, and not the ℓ_2-sensitivity or some other notion. The answer is that we use it for technical reasons, though ℓ_2-sensitivity is the right notion in other settings (say, for the *Gaussian mechanism*, rather than the Laplace mechanism). Note that these are identical in the univariate setting (i.e., when $k = 1$), but may vary in the multivariate setting (up to a factor of $\sqrt{k}$).

As a simple running example, we will consider the function $f = \frac{1}{n}\sum_{i=1}^n X_i$, where $X_i \in \{0, 1\}$. It is not hard to verify that the sensitivity of this function is $1/n$, realized when any bit is flipped.

As the name of the mechanism suggests, the *Laplace distribution* will be a key component of the Laplace mechanism.

Definition 2. *The* Laplace distribution *with location and scale parameters* 0 *and b, respectively, has the following density:*

$$p(x) = \frac{1}{2b}\exp\left(-\frac{|x|}{b}\right).$$

Note that the variance of this distribution is $2b^2$. Some visualizations of the density of the Laplace distribution are provided in Figure 1. It can be seen as a symmetrization of the exponential distribution, which is only supported on $x \in [0, \infty)$ and has density $\propto \exp(-cx)$, versus the Laplace distribution which is supported on $x \in \mathbb{R}$ and has density $\propto \exp(-c|x|)$. As another potentially familiar point of comparison, the Gaussian distribution is also supported on $\mathbb{R}$, and has density $\propto \exp(-cx^2)$. We can see the Gaussian distribution has lighter tails than the Laplace distribution, meaning that it enjoys somewhat stronger concentration (though both tails decay at least exponentially).

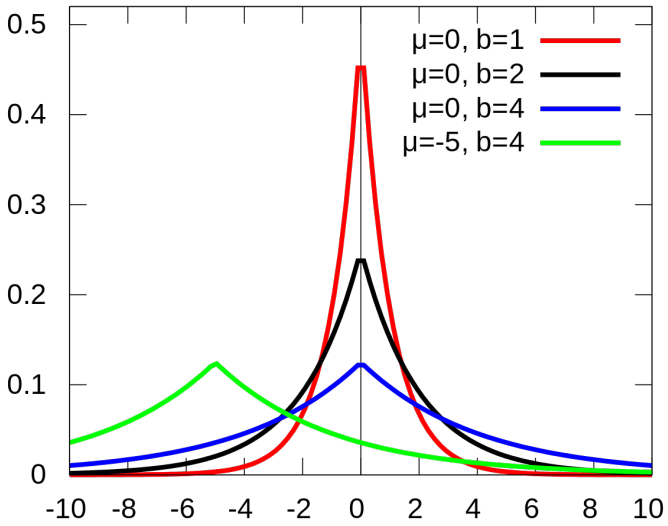

Figure 1: Figure from Wikipedia. Laplace distributions with various parameters.

With the Laplace distribution in hand, we are ready to introduce the Laplace mechanism. It is very simple to state: add noise to the statistic of magnitude proportional to its sensitivity.

Definition 3. *Let* $f : \mathcal{X}^n \to \mathbb{R}^k$. *The* Laplace mechanism *is defined as*

$$M(X) = f(X) + (Y_1, \ldots, Y_k),$$

where the Y_i *are independent* Laplace(Δ/ε) *random variables.*

Let us apply this to our running example of $f = \frac{1}{n}\sum_{i=1}^n X_i$. This is a simple application of Definition 3, where $k = 1$. As we previously established, $\Delta = 1/n$. Therefore, the Laplace mechanism run on a dataset is $\tilde{p} = f(X) + Y$, where Y is Laplace$(1/\varepsilon n)$. Recalling that we previously defined $p = f(X)$, we have that $\mathbf{E}[\tilde{p}] = p$, by linearity of expectations and since $\mathbf{E}[Y] = 0$. Computing the variance, we have $\mathbf{Var}[\tilde{p}] = \mathbf{Var}[Y] = O(1/\varepsilon^2 n^2)$, and using Chebyshev's inequality gives that $|\tilde{p} - p| \leq O(1/\varepsilon n)$ with reasonable probability.[1] We can compare this with the accuracy of ε-randomized response, which was $O(1/\varepsilon\sqrt{n})$ – the Laplace mechanism's error is quadratically smaller in n.

It remains to show that the Laplace mechanism is differentially private.

[1]Note that one can easily get a high probability bound by examining the tails of the distribution – the error will exceed $O(\log(1/\beta)/\varepsilon n)$ with probability $\leq \beta$.

Theorem 4. *The Laplace mechanism is ε-differentially private.*

Proof. Let X and Y be any neighbouring databases, differing in any one entry. We let $p_X(z)$ and $p_Y(z)$ be the probability density functions of $M(X)$ and $M(Y)$ evaluated at a point $z \in \mathbb{R}^k$. To prove differential privacy, we will show that that their ratio is bounded above by $\exp(\varepsilon)$, for an arbitrary choice of z and neighboring X and Y.

$$\begin{aligned}
\frac{p_X(z)}{p_Y(z)} &= \frac{\prod_{i=1}^k \exp\left(-\frac{\varepsilon|f(X)_i - z_i|}{\Delta}\right)}{\prod_{i=1}^k \exp\left(-\frac{\varepsilon|f(Y)_i - z_i|}{\Delta}\right)} \\
&= \prod_{i=1}^k \exp\left(-\frac{\varepsilon(|f(X)_i - z_i| - |f(Y)_i - z_i|)}{\Delta}\right) \\
&\leq \prod_{i=1}^k \exp\left(-\frac{\varepsilon|f(Y)_i - f(X)_i|}{\Delta}\right) \\
&= \exp\left(\frac{\varepsilon \sum_{i=1}^k |f(X)_i - f(Y)_i|}{\Delta}\right) \\
&= \exp\left(\frac{\varepsilon\|f(X) - f(Y)\|_1}{\Delta}\right) \\
&\leq \exp(\varepsilon).
\end{aligned}$$

The first inequality is the triangle inequality, and the last uses the definition of ℓ_1-sensitivity. □

Counting Queries

We'll apply this in a few different scenarios. First, let's look at *counting queries*. This is essentially the non-normalized version of our running example we have used so far (though the term counting query is sometimes used interchangeably for both). Specifically, we can ask the question "How many people in the dataset have property P?" If we just ask one question like this, the analysis follows very similarly to before. Each individual will have a bit $X_i \in \{0, 1\}$ indicating whether or not this is true about them, and the function f we consider is their sum. The sensitivity is 1, and thus an ε-differential privatization of this statistic would be $f(X) + \text{Laplace}(1/\varepsilon)$. This introduces error to this query on the order of $O(1/\varepsilon)$, independent of the size of the database.

What if we wanted to answer many queries? The way we defined the Laplace mechanism this makes this easy to reason about. Suppose we had k counting queries $f = (f_1, \ldots, f_k)$, which are all specified in advance. We would simply output the vector $f(X)+Y$, where the Y_i's are i.i.d. Laplace random variables. But what scale parameter should we use for the Y_i's? Each individual counting query f_j has sensitivity 1, but we are using the same dataset to answer all queries, so changing a single individual may affect the result of many queries at once. Consider, for example, the swapping of two individuals: one who satisfies no properties, and one who satisfies every property. This swap would change the result of every query by 1, and therefore the overall ℓ_1 sensitivity is k. Let's analyze this slightly more mathematically. Since $f(X) = \sum(f_1(X_i), \ldots, f_k(X_i))$, if neighbouring datasets X and Y differ in that one contains x and the other contains y, the ℓ_1 difference can be

written as $\sum_j |f_j(x) - f_j(y)|$, as the common terms cancel. This can be upper bounded as follows: $\sum_j |f_j(x) - f_j(y)| \leq \sum_j 1 = k$.

With this sensitivity bound $\Delta = k$ in hand, we can add $Y_i \sim \text{Laplace}(k/\varepsilon)$ noise to each coordinate, answering each counting query with error of magnitude $O(k/\varepsilon)$.

Some discussion is in order. First, this method of answering k counting queries required us to specify all the queries in advance – in other words, a *non-adaptive* setting. We will later see that similar guarantees are achieveable in the adaptive setting, where the choice of a query may depend on previous ones. Secondly, let's compare this with the Dinur-Nissim attacks [DN03] discussed in previous lectures. That showed that if the analyst asks $\Omega(n)$ counting queries, defended by the curator using noise of magnitude $O(\sqrt{n})$, the analyst can reconstruct the database and cause blatant non-privacy. On the other hand, the above strategy shows that, if the analyst asks $O(n)$ counting queries and the curator adds noise of magnitude $O(n/\varepsilon)$, then privacy is preserved. This seems to be a huge gap in the two results: are there stronger attacks, which allow the adversary to succeed even with more noise? Or can we add less noise and still preserve privacy? Fortunately, the latter is true, and it is possible to add less noise via better analysis (as well as a slight relaxation of the definition of differential privacy), using something called *advanced composition*.

Histograms

Another natural type of query is a *histogram query*. With counting queries, we had to be pessimistic – changing a single individual could affect the results of every query at once. But certain *structures* of queries might allow us to perform better sensitivity analysis. Suppose each individual in the dataset has some categorical feature: for example, let's say the person's age (rounded down to the nearest whole number). We would like to answer questions like "How many people in the dataset are X years old?" While this is similar to the counting queries example, an individual here can not have more than one age. Our function f will be $(f_0, f_1, \ldots, f_{k-1})$, where f_i asks how many people are i years old. It is not hard to argue that the ℓ_1-sensitivity of this function is 2: changing any individual's age would result in one count decrementing and another count incremening. More formally, similar to before we consider neighbouring datasets X and Y, where the difference is that one dataset has x and the other has it replaced by y. Then the ℓ_1 sensitivity is equal to $\|e_a - e_b\|_1 = 2$, where e_j is the j-th standard basis vector (having a 1 in the jth position and 0 elsewhere), and f_a and f_b are the functions which evaluate to 1 on x and y. As such, the Laplace mechanism prescribes outputting $f(X) + Y$, where $Y_i \sim \text{Laplace}(2/\varepsilon)$, where the magnitude is independent of the number of "bins" k.

How much error does this incur? As before, we observe that any individual count will have error on the order of $O(1/\varepsilon)$. However, we can also reason about the error incurred in all counts simultaneously! We can use the following basic fact about the Laplace distribution:

Fact 5. *If* $Y \sim \text{Laplace}(b)$*, then*

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

This can be verified simply by integrating the PDF of the Laplace distribution. Now, for the ith bin, the error in the count is exactly Y_i, and we have that $\Pr[|Y_i| \geq 2\log(k/\beta)/\varepsilon] \leq \beta/k$. Taking a union bound over all bins, it means that the probability that *any* bin has error $\geq 2\log(k/\beta)/\varepsilon$ is at most β. Stated differently: the magnitude of the error scales only logarithmically with the number of bins, in contrast to the linear relationship when our counting queries were arbitrary.

Properties of Differential Privacy

One of the reasons for the success of differential privacy is how "user friendly" it is. Specifically, it possesses a number of convenient properties that make it possible to think about differential privacy in a very modular fashion. We will discuss some of the most fundamental properties: closure under post-processing, group privacy, and basic composition.

Post-Processing

One convenient fact about differentially private algorithms is that once a quantity is privatized, it can't be "un-privatized," if the data is not used again. We used this already when we were analyzing randomized response.

Theorem 6. *Let $M : \mathcal{X}^n \to \mathcal{Y}$ be ε-differentially private, and let $F : \mathcal{Y} \to \mathcal{Z}$ be an arbitrary randomized mapping. Then $F \circ M$ is ε-differentially private.*

Proof. Since F is a randomized function, we can consider it to be a distribution over deterministic functions f. The privacy proof follows for every neighbouring dataset X, X' and $T \subseteq \mathcal{Y}$:

$$\begin{aligned}\Pr[F(M(X)) \in T] &= \mathbf{E}_{f\sim F}[\Pr[M(X) \in f^{-1}(T)]] \\ &\leq \mathbf{E}_{f\sim F}[e^{\varepsilon} \Pr[M(X') \in f^{-1}(T)]] \\ &= e^{\varepsilon} \Pr[F(M(X')) \in T].\end{aligned}$$

□

Group Privacy

So far, we've discussed differential privacy with respect to neighbouring datasets – ones which differ in exactly one entry. But one might wonder about datasets which differ in multiple entries. The definition of differential privacy allows for the guarantee to decay gracefully as the distance is increased.

Theorem 7. *Let $M : \mathcal{X}^n \to \mathcal{Y}$ be an ε-differentially private algorithm. Suppose X and X' are two datasets which differ in exactly k positions. Then for all $T \subseteq \mathcal{Y}$, we have*

$$\Pr[M(X) \in T] \leq \exp(k\varepsilon) \Pr[M(X') \in T].$$

Proof. The proof follows by what is known in the business as a "hybrid" argument. Let $X^{(0)} = X$, $X^{(k)} = X'$ – since they differ in k positions, there exists a sequence $X^{(0)}$ through $X^{(k)}$ such that each consecutive pair of datasets is neighbouring. Then, for all $T \subseteq \mathcal{Y}$:

$$\begin{aligned}\Pr[M(X^{(0)}) \in T] &\leq e^{\varepsilon} \Pr[M(X^{(1)}) \in T] \\ &\leq e^{2\varepsilon} \Pr[M(X^{(2)}) \in T] \\ &\cdots \\ &\leq e^{k\varepsilon} \Pr[M(X^{(k)}) \in T].\end{aligned}$$

□

(Basic) Composition

As a final but important property, we discuss *composition* of differentially private algorithms. Suppose you ran k differentially private algorithms on the same dataset, and released all of their results – how private is this as a whole? Essentially, the overall privacy guarantee decays by a factor of k. We already saw this when we considered the Laplace mechanism when the queries were chose in advance, but the following result holds for general differentially private algorithms, even when the queries are chosen adaptively!

Theorem 8. *Suppose $M = (M_1, \ldots, M_k)$ is a sequence of ε-differentially private algorithms, potentially chosen sequentially and adaptively. Then M is $k\varepsilon$-differentially private.*

Proof. Fix two neighbouring datasets X and X', and consider some sequence of outputs $y = (y_1, \ldots, y_k)$. Then we have

$$\begin{aligned}\frac{\Pr[M(X) = y]}{\Pr[M(X') = y]} &= \prod_{i=1}^{k} \frac{\Pr[M_i(X) = y_i | (M_1(X), \ldots, M_{i-1}(X)) = (y_1, \ldots, y_{i-1})]}{\Pr[M_i(X') = y_i | (M_1(X'), \ldots, M_{i-1}(X')) = (y_1, \ldots, y_{i-1})]} \\ &\leq \prod_{i=1}^{k} \exp(\varepsilon) \\ &= \exp(k\varepsilon).\end{aligned}$$

□

Surprisingly, it is possible to do better: we can get away with paying a factor of $O(\sqrt{k})$ in the privacy parameter, rather than the k given above, However, this will require a relaxation of the privacy notion, which we will leave for next lecture.

References

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, pages 202–210, New York, NY, USA, 2003. ACM.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.