DS 102 Data, Inference, and Decisions **Spring 2020**

Lecture 29: Differential Privacy

Lecturer: Moritz Hardt

29.1 Differential Privacy: Definition

We finished the last lecture by discussing randomized response, an old technique which intuitively provided a reasonable privacy guarantee. In this lecture, we extend this idea to a much more general setting, and introduce a definition which enables quantifying the level of privacy.

In particular, the topic of this lecture is **differential privacy**, which is a mathematical privacy notion aimed at privacy-preserving statistical data analysis. The main intuition that differential privacy formalizes is the following:

"Whether or not you're in the data set should have little effect on the output of the analysis."

The idea is that any given individual should be willing to participate in the statistical analysis because their participation in the study does not change the outcome of the study by very much; their personal information cannot be recovered because similar results would be obtained if that individual never participated in the first place.

We will need the notion of *neighboring data sets*. Two data sets D and D' are called neighboring if they differ in at most one data record. For example, D could be the GWAS test population which includes Moritz's DNA, and D' could be the same data set but with Moritz's DNA removed.

Differential privacy can be informally defined as follows: a randomized algorithm $\mathcal{A}(\cdot)$ is differentially private if for all neighboring data sets D, D' and all events S:

$$\mathbb{P}(\mathcal{A}(D) \in S) \approx \mathbb{P}(\mathcal{A}(D') \in S).$$

The probabilities are taken over the *randomness of the algorithm*, and not over the randomness of the data sets; we treat the data sets as fixed. It remains to define "$\approx$". To do so, we will introduce a parameter ϵ which quantifies the similarity of the distributions of $\mathcal{A}(D)$ and $\mathcal{A}(D')$. This parameter ϵ will also act as the strength of the privacy guarantee.

Formally, we call a randomized algorithm $\mathcal{A}(\cdot)$ ϵ-differentially private if for all neighboring data sets D, D' and all events S:

$$\mathbb{P}(\mathcal{A}(D) \in S) \leq e^{\epsilon}\mathbb{P}(\mathcal{A}(D') \in S),$$

where $\epsilon \geq 0$. Usually ϵ is some small constant between 0 and 1 (e.g. 0.01), and for such small ϵ, we can approximate the exponential well by $e^{\epsilon} \approx 1 + \epsilon$. Therefore, differential privacy says that $\mathbb{P}(\mathcal{A}(D) \in S)$ can be only a little bit larger than $\mathbb{P}(\mathcal{A}(D') \in S)$ (e.g. $1.01 \cdot \mathbb{P}(\mathcal{A}(D') \in S)$).

The following pictures illustrates this definition.

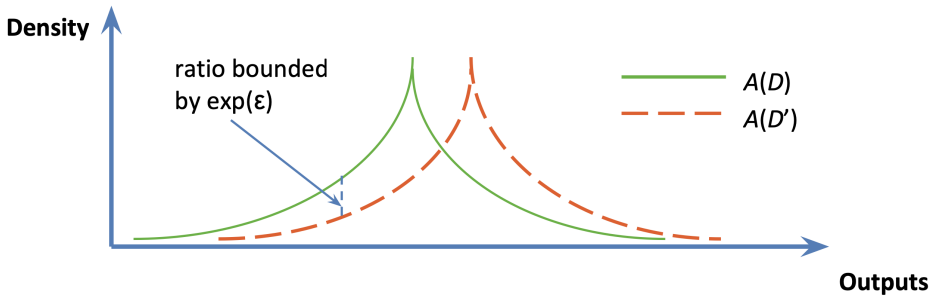

Figure 29.1: Illustration of the definition of differential privacy.

Smaller ϵ implies more privacy. In particular, $\epsilon = 0$ means that the distributions of $\mathcal{A}(D)$ and $\mathcal{A}(D')$ have to match exactly, in which case the output of the randomized algorithm $\mathcal{A}(\cdot)$ has to be completely independent of D. Bigger ϵ allows $\mathcal{A}(\cdot)$ to be less private and more sensitive to the input data set.

Differential privacy should be interpreted as limiting the harm resulting from participation—whether you're in or out is about the same. It enables population-level inferences (e.g. evaluating if smoking causes cancer), rather than inferences about a single person (e.g. revealing the identity of the richest person in the U.S.). It is worth keeping in mind that, although private, such inferences might still be harmful to you. For example, if private data analysis reveals that smoking causes cancer, your insurance premium might go up if you're a smoker.

29.2 Differential Privacy in the Query Model

Differential privacy is often studied in the query model. There are two parties, one being a data analyst, and the other being a trusted curator who has access to a data set D. The trusted curator could be, for example, the NIH, and the data analyst could be a researcher who wants to analyze the private data set D.

The data analyst submits queries to the trusted curator (i.e. questions about the data set D), and the trusted curator returns answers in a differentially private way, typically by adding noise to the true answer to the posed query. The goal is to minimize the magnitude of this noise, while maintaining differential privacy (for a fixed privacy level).

The most common model of queries, which have also been widely studied in the privacy literature, are called *statistical queries*. Assume the data set D is a subset of some universe X. For example, X could be all binary strings of length d, i.e. $X = \{0,1\}^d$, where each entry of the string corresponds to a property of an individual, and we have d individuals in total. A statistical query is then a mapping from X to $[0,1]$, $q : X \to [0,1]$. The true answer to query q on data set D is then $q(D) := \sum_{x \in D} q(x)$. This answer is between 0 and $n = |D|$. For example, a query could be: "How many people in D smoke and have cancer?" The answer is naturally between 0 and $n = |D|$.

Note that, by the definition of statistical queries, if D and D' are neighboring data sets, then

$$|q(D) - q(D')| \leq 1.$$

The quantity $|q(D) - q(D')|$ is called *query sensitivity*, and it determines how much noise we need to add to guarantee that the answers to the queries are differentially private. In this lecture we will just assume that sensitivity is equal to 1.

The first algorithm for answering statistical queries in a differentially private way was the **Laplace mechanism**. Given a query q, it outputs $q(D) + \xi$, where ξ is noise distributed according to the Laplace distribution, $\xi \sim \text{Lap}(1/\epsilon)$. The density of this output, which is a Laplace distribution centered at $q(D)$, is:

$$p(x) = \frac{\epsilon}{2} e^{-\epsilon|x-q(D)|}.$$

Another mechanism for differential privacy is randomized response. Suppose n individuals have sensitive -1/1 bits $D = (b_1, b_2, \ldots, b_n)$. Randomized response computes noisy bits $b'_i \sim \text{Bern}(\frac{1}{2} + \epsilon b_i)$, and releases $RR(D) = \sum_{i=1}^{n} b'_i$. One can show that for small enough ϵ (e.g. $\epsilon < 1/4$), the randomized response mechanism is 2ϵ-differentially private.

29.3 Local Differential Privacy

In the randomized response mechanism, we added noise to each sensitive bit individually, rather than to the aggregated answer to the query on the whole data set D. This idea falls under *local differential privacy*.

More generally, in local differential privacy each individual computes the randomization locally, before sending their privatized input to the aggregation step. In particular, in randomized response, the noisy bit b'_i is already differentially private before the bits are aggregated. This is in contrast with "central differential privacy", where sensitive data is made private only after it has been aggregated (e.g. as we saw in the Laplace mechanism example).

The central approach generally adds less noise, however the local approach ensures privacy even when the data curator is untrusted. For example, if the data aggregator is an untrusted company, one might wish to randomize their data before sending it over to the company.

29.4 Composition Guarantees for Differential Privacy

An important property of differential privacy is *composition*. If we have k ϵ-differentially private algorithms, and we combine them in any way we like (e.g. we compose them sequentially or in parallel), we are guaranteed that the output of the composition is $k\epsilon$-differentially private.

For example, the curator might receive queries sequentially $q_1, q_2, \ldots, q_k$, and after receiving each query, they output $q_i(D) + \text{Lap}(k/\epsilon)$. This ensures overall ϵ-differential privacy. Notice that the noise magnitude scales linearly with the number of queries.

We prove this composition fact by induction. First, we claim that $\mathcal{A}(D) = (\mathcal{A}_1(D), \mathcal{A}_2(D))$ is $(\epsilon_1 + \epsilon_2)$-differentially private, if $\mathcal{A}_i(D)$ is ϵ_i-differentially private. For simplicity, we will assume that $\mathcal{A}_i(D)$ are discrete random variables. This is just for the sake of a simpler proof; all claims we make are morally correct even if $\mathcal{A}_i(D)$ is continuous.

Fix some arbitrary neighboring data sets D and D', and any two outputs r_1 in the range of $\mathcal{A}_1$ and r_2 in the range of $\mathcal{A}_2$. Then, we have:

$$\frac{\mathbb{P}(\mathcal{A}(D) = (r_1, r_2))}{\mathbb{P}(\mathcal{A}(D') = (r_1, r_2))} = \frac{\mathbb{P}(\mathcal{A}_1(D) = r_1)\mathbb{P}(\mathcal{A}_2(D) = r_2)}{\mathbb{P}(\mathcal{A}_1(D') = r_1)\mathbb{P}(\mathcal{A}_2(D') = r_2)} = \left(\frac{\mathbb{P}(\mathcal{A}_1(D) = r_1)}{\mathbb{P}(\mathcal{A}_1(D') = r_1)}\right)\left(\frac{\mathbb{P}(\mathcal{A}_2(D) = r_2)}{\mathbb{P}(\mathcal{A}_2(D') = r_2)}\right) \leq e^{\epsilon_1}e^{\epsilon_2},$$

where in the last step we use the assumption that $\mathcal{A}_i$ is ϵ_i-differentially private. The general case, when k algorithms are composed, follows from the base case by induction.

29.4.1 Approximate Differenital Privacy

In fact, one can show even better composition properties under a relaxation of differential privacy. This relaxation is usually referred to as *approximate* differential privacy.

We say a randomized algorithm $\mathcal{A}(\cdot)$ is (ϵ, δ)-differentially private if for all neighboring data sets D, D' and all events S:

$$\mathbb{P}(\mathcal{A}(D) \in S) \leq e^{\epsilon}\mathbb{P}(\mathcal{A}(D') \in S) + \delta.$$

Differential privacy is equivalent to approximate differential privacy when $\delta = 0$. Because δ is an additive term rather than multiplicative, it is typically much smaller than ϵ (think $\epsilon = 0.01$ and $\delta = o(1/|D|)$).

It is useful to think of approximate differential privacy as differential privacy with δ chance of failure.

Note that, if $\delta \geq 1/|D|$, one can easily break privacy. For example, consider $\mathcal{A}(D)$ that selects a single entry (i.e. individual) uniformly at random in the database D and just releases it without any further randomization. Clearly this violates any reasonable privacy notion.

Although approximate differential privacy is weaker than differential privacy, it allows much better composition. In particular, composing k (ϵ, δ)-differentially private algorithms roughly results in $(O(\sqrt{k}\epsilon), k\delta)$-differential privacy. Usually δ is negligible so the $k\delta$ factor doesn't matter all that much, so now the privacy loss has factor $\sqrt{k}$ instead of k. In other words, with (ϵ, δ)-differential privacy one can gain a quadratic factor in the number of queries that can be answered privately. Also, one can show that the $\sqrt{k}$ factor is optimal and cannot be improved.

29.4.2 Multiplicative Weights Approach

The previous composition results allow non-trivial answers to a number of queries that is quadratic in the database size. We briefly discuss a multiplicative weights approach that allows handling huge query sets—it allows non-trivial answers to exponentially many queries.

In the context of this multiplicative weights algorithm, we will think of the database D as a vector with $|X|$ coordinates, one for each possible data point. The i-th coordinate of the vector is equal to the number of times the i-th point of space X appears in D. A statistical query now essentially becomes a vector in this histogram space.

The multiplicative weights algorithm we consider takes as input a histogram D and query set Q, and outputs a histogram D^* which satisfies differential privacy, but also $|q(D) - q(D^*)|$ is small for all $q \in Q$. We only sketch the steps of this algorithm.

The input is a histogram D and query set Q. The algorithm first initializes D_0 to be a uniform histogram. At every time step t, the algorithm finds a "bad" query $q \in D$ where $|q(D) - q(D_{t-1})|$ is too large. Then, it improves the histogram using this query q: $D_t = \text{MWUpdate}(D_{t-1}, q)$. After T rounds, it outputs $D^* = D_T$. It can be shown that one runs out of such "bad" queries quite quickly.

29.4.3 Application: Differentially Private Gradient Descent

Suppose we want to train a model on sensitive data using gradient descent. In standard gradient descent, the update rule is:

$$w_{t+1} = w_t - \alpha \nabla_w \text{loss}(w_t, D),$$

for some step size α. For example, we could look at the squared loss:

$$\text{loss}(w_t, D) = \sum_{(x,y) \in D} (\langle w_t, x \rangle - y)^2.$$

Because we can represent the loss as a sum over sensitive data points, each coordinate of $\nabla_w \text{loss}(w_t, D)$ is actually a statistical query. Hence, we can make gradient descent private using the techniques we saw earlier in this lecture (e.g. add Laplace noise to the gradients). Moreover, because we know the composition properties of differential privacy, we can adjust the noise scaling for any number of gradient descent steps we wish to run, such that the overall sequence $\{w_t\}$ is ϵ-differentially private.