Privacy in Statistics and Machine Learning — Spring 2023
Lecture 4: Differential Privacy I

Adam Smith

1 Defining "Privacy"

Having seen reconstruction attacks, we now want to get a handle on what it means that some set of statistics are actually ok to release—that they don't expose individuals' data (too much?) to attacks like the ones of the last two lectures. The question isn't new. Researchers in statistics, computer science, and information theory have been tackling variations on it since the 1960's [War65], and a many algorithms and techniques were developed that resist specific suites of attacks.

However, our goal will be to find a general criterion we can use to reason about many different kinds of released information, and about a broad class of attacks. In fact, what we really want is a clear sense in which we've prevented "all reasonable" attacks. *Differential privacy* provides one approach to this conundrum. Before we get to it, however, it is helpful to see an example of something that does *not* meet our desiderata.

k-Anonymity [Swe02] is one popular approach to reasoning about the privacy implications of publishing statistical tables. It applies only to specific kinds of information, called *generalized microdata*. This means a table of individual records, where each entry is either the original record's entry (a specific person's real age, for example) or a set of possible values for that entry (often an interval, like 30–34). Figure 1 gives an example of such a table with age and zip code data. The basic idea of k-anonymity is to divide attributes into 'non-sensitive' attributes—assumed to be available to an attacker—and 'sensitive' ones—assumed to be unknown—and to ensure that *every record matches at least $k-1$ others in the nonsensitive attributes.* That is, given an integer k, a table is k-anonymous if, when we delete the sensitive attributes and leave only the non-sensitive ones, each row appears at least k times.[1] The table of Figure 1 is 4-anonymous.

	Non-Sensitive			Sensitive
	Zip code	Age	Nationality	Condition
1	130**	<30	*	AIDS
2	130**	<30	*	Heart Disease
3	130**	<30	*	Viral Infection
4	130**	<30	*	Viral Infection
5	130**	≥40	*	Cancer
6	130**	≥40	*	Heart Disease
7	130**	≥40	*	Viral Infection
8	130**	≥40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Figure 1: A 4-anonymous table.

The idea behind this criterion is that it makes linkage attacks (like the Netflix example from Lecture 1 [NS08]) harder to carry out—if an attacker has access to another table that contains some of the non-sensitive attributes, then each record in a k-anonymous table will match at least k of the records in

[1] Sweeney's original notion [Swe02] was actually a bit more permissive: the condition did not have to hold simultaneously for all non-sensitive attributes, but only for those subsets of them, called *quasi-identifiers*, that were likely to apppear together in other tables. The simpler notion is good enough for our discussion.

the other table.

While k-anonymity is likely to make those specific attacks harder than they would be with raw data, a k-anonyous table can still leak lots of individual-level information. We can glean lots of information from the table in Figure 1: everyone in their 30's has cancer; our friend Alice, who's data we happen to know is in the table, cannot have visited the hospital because of a broken leg; etc. Of course, the example is simplistic (real hospital records don't look like the example in the table...) but it illustrates two important points: 1. Defending against one type of attack isn't sufficient, and 2. Criteria that limit the *form* of the output (in this case, the number of occurrences of each vector of non-sensitive attributes) do not necessarily constrain the *information* that is revealed.

Composition k-anonymity illustrates another important point, namely that when the same record is included in two (or more) data sets that are anonymized separately, the combination of the two might reveal far more than the two do individually [GKS08]. For example, consider the table of Figure 2. Suppose we know that Alice's record appears in both tables, and that she is 28 years old and lives in zip code 13012. Neither table on its own pins down her condition exactly (each one narrows it down to a few posibilities), but taken together they pin it down exactly.

This problem is known as *composition*—what happens when many different pieces of information are revealed about me? We return to this question in the next lecture.

	Non-Sensitive			Sensitive
	Zip code	Age	Nationality	Condition
1	130**	<35	*	AIDS
2	130**	<35	*	Tuberculosis
3	130**	<35	*	Flu
4	130**	<35	*	Tuberculosis
5	130**	<35	*	Cancer
6	130**	<35	*	Cancer
7	130**	≥35	*	Cancer
8	130**	≥35	*	Cancer
9	130**	≥35	*	Cancer
10	130**	≥35	*	Tuberculosis
11	130**	≥35	*	Viral Infection
12	130**	≥35	*	Viral Infection

Figure 2: A 6-anonymous table.

Attacks on k-anonymization in practice To see how these types of observations—and much more subtle attacks—play out in practice, see [Coh22], which gives general principles and uses them to break the anonymity of a specific release of educational data.

Form versus process (or syntax versus semantics) Perhaps the most important lesson we can draw from the examples above is that, to come up with a general approach to privacy of statistical data, it isn't enough to restrict the form of the outputs we generate. k-anonymity specifies a set of acceptable outputs, but doesn't substantially restrict *how* they are produced.

2 A First Example: Randomized Response

Let's recall the randomized reponse mechanism from Lecture 1. Suppose that our data set consists of a single bit $x_i \in \{0, 1\}$ for each of n individuals. For each person, we'll generate a biaed random bit Y_i as follows: roll a fair die. If the die comes up 1,2,3, or 4, we set $Y_i = x_i$. Otherwise, we set Y_i to be the

Algorithm 1: Basic Randomized Repsonse, RR_{basic}

Input: Data set of n bits: $\mathbf{x} = (x_1, ..., x_n) \in \{0,1\}^n$

Output: Bits $Y_1, ..., Y_n$

for $i = 1$ *to* n **do**

  $Y_i = \begin{cases} x_i & \text{w.p. } 2/3 \\ 1 - x_i & \text{w.p. } 1/3 \end{cases}$;

  return $(Y_1, ..., Y_n)$

opposite value to x_i (that iis, $Y_i = 1 - x_i$). The algorithm's output is the list of values $(Y_1, ..., Y_n)$. Let RR_{basic} denote the resulting algorithm (spelled out in Algorithm 1).

What sort of privacy does RR_{basic} provide? There are many ways to answer the question, but one way is to think of a sort of *plausible deniability*. For any individual i, seeing a particular value of Y_i in the output doesn't give an outsider much information about whether $x_i = 0$ or $x_i = 1$. For any particular output y_i, we have:

$$\frac{1}{2} \leq \frac{\mathbb{P}\left(Y_i = y_i \mid x_i = 1\right)}{\mathbb{P}\left(Y_i = y_i \mid x_i = 0\right)} \leq 2 \tag{1}$$

In other words, the outcome would have been roughly as likely if we had changed person i's record from one to 0 or vice-versa (assuming everyone else's records were unchanged).

Proposition 2.1. *There is a procedure that, given the outputs $Y_1, ..., Y_n$ from randomized response on input $x_1, ..., x_n$, returns an estiamte A such that*

$$\sqrt{\mathbb{E}\left(\left(A - \sum_{i=1}^{n} x_i\right)^2\right)} = O(\sqrt{n}) .$$

Exercise 2.2. Prove Proposition 2.1.

Now a factor of 2 maybe not be quite satisfactory. But we can get it to be arbitrarily close to 1 by changing the mechanism a bit. Suppose we want that odds ration to be bounded by e^{ε} for small number $\varepsilon > 0$. (Recall that $e^{\varepsilon} \approx 1 + \varepsilon$ when ε is close to 0.) Algorithm 2 gives a version which takes data in an arbitrary set $\mathcal{U}$, along with a predicate φ that maps each record to a bit.

Algorithm 2: Randomized Response, RR_{ε}

Input: Data set of n bits: $\mathbf{x} = (x_1, ..., x_n) \in \mathcal{U}^n$, predicate $\varphi : \mathcal{U} \to \{0,1\}$, and a parameter $\varepsilon > 0$

Output: Bits $Y_1, ..., Y_n$

for $i = 1$ *to* n **do**

  $Y_i = \begin{cases} \varphi(x_i) & \text{w.p. } \frac{e^{\varepsilon}}{e^{\varepsilon}+1} \\ \varphi(1 - x_i) & \text{w.p. } \frac{1}{e^{\varepsilon}+1} \end{cases}$;

  return $(Y_1, ..., Y_n)$;

Even though no particular $\varphi(x_i)$ can be learned with confidence, when n is large we can use the Y_i's to estimate the proportion of records that satisfu φ. Figure 5(a) shows the accuracy of randomized response for estimating for estimating the proportion of individuals in some population who satisfy some predicate, for different values of n. For $\varepsilon = 0.5$, the accuracy becomes reasonable for n in the low thousands.

Proposition 2.3. *There is a procedure that, given the outputs $Y_1, ..., Y_n$ from randomized response (Alg. 2) on input $x_1, ..., x_n$, returns an estimate A such that*

$$\sqrt{\mathbb{E}\left(\left(A - \sum_{i=1}^{n} \varphi(x_i)\right)^2\right)} \leq \frac{e^{\varepsilon/2}}{e^{\varepsilon} - 1}\sqrt{n}\,.$$

For bounded ε (say, less than 1), this bound is $\Theta\left(\frac{\sqrt{n}}{\varepsilon}\right)$.

Exercise 2.4. Prove Proposition 2.3. (*Hint:* For which constants a, b do we have $\mathbb{E}\,(aY_i - b) = x_i$?)

Exercise 2.5. Strengthen Proposition 2.3 as follows: show that there is a constant $c > 0$ such that, for every $t > 1$, the probability that $\left|A - \sum_{i=1}^{n} \varphi(x_i)\right| \geq t\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1}\sqrt{n}$ is at most $2\exp(-ct^2)$. (*Hint:* Write A as a sum of independent random variables and apply a Chernoff bound.) (The exact form of the function of ε isn't really important here. Anything expression that scales as $\Theta\left(\frac{1}{\varepsilon}\right)$ will do.

Exercise 2.6. It is typical to analyze a mechanism like randomized response in terms of how well it does at estimating the *average* $\frac{1}{n}\sum_{i=1}^{n} \varphi(x_i)$, instead of the sum (since if we add more data from the same population, the average should stay more or less the same). We can use $\frac{A}{n}$ (where A is the estimate from Prop. 2.3) to estimate the average, and its standard deviation will $\Theta\left(\frac{1}{\varepsilon\sqrt{n}}\right)$ (for $\varepsilon \leq 1$).

Suppose we want the standard deviation of this estimate to be at most α. For a privacy parameter ε, how large a dataset $n(\alpha, \varepsilon)$ do we need? (Write ann explicit function for $n(\alpha, \varepsilon)$, as well as a simple asymptotic expression for the setting where $\varepsilon \leq 1$). If we halve α, what will happen to $n(\alpha, \varepsilon)$? What if we halve ε when ε is small?

3 Differential Privacy

Let $\mathcal{U}$ be the set of all possible records for each individual. A dataset $\mathbf{x}$ is thus a multiset[2] of values in $\mathcal{U}$. When the size n is fixed, we may think of it as a list $\mathbf{x} = (x_1, ..., x_n) \in \mathcal{U}^n$. (It is often conventient to view it as a *histogram*, that is, a function $\mathcal{U} \to \mathbb{N}$ that counts the number of occurrences of each possible record in $\mathcal{U}$. We will return to this view later; for now, we'll stick with lists.)

A Thought Experiment The main idea of DP is to consider a thought experiment in which we compare how an algorithm behaves on a data set $\mathbf{x}$ with the way it behaves on a hypothetical dataset $\mathbf{x}'$ in which one person's record has been replaced with some other value.

We say *two data sets are neighbors if they differ in one individual's record.* A simple way to model this is to think of the size n of data sets as fixed, and to consider two data sets adjacent if one record has been replaced with a different value. For example, if they differ in index i, we would have:

$$\begin{aligned} \mathbf{x} &= (x_1, x_2, ..., x_{i-1}, \quad x_i, \quad x_{i+1}, ..., x_n) \\ \mathbf{x}' &= (x_1, x_2, ..., x_{i-1}, \quad x_i', \quad x_{i+1}, ..., x_n) \end{aligned}$$

Now consider a randomized algorithm A. For each possible input data set $\mathbf{x}$, its output is a random variable $A(\mathbf{x})$. We say an algorithm is differentially private if running the algorithm on two neighboring data sets yields roughly the same distribution on outcomes. Specifically, we'll ask that for every set E of possible outcomes—for example, those outputs from a healthcare study that lead to individual i being

[2] A multiset is a set where we keep track of how many times each element appears.

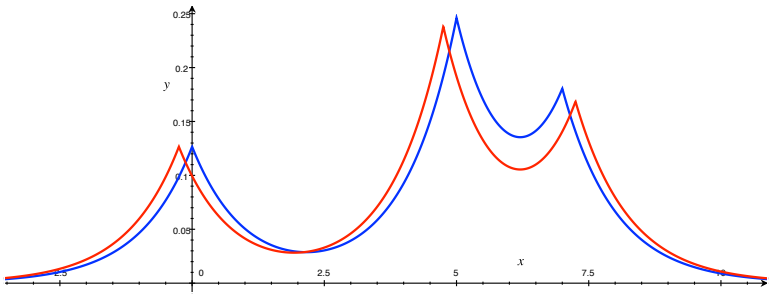


Figure 3: Two distributions P and Q that satisfy: for every event E, $P(E) \leq e^{1/4}Q(E)$ and $Q(E) \leq e^{1/4}P(E)$.

denied health insurance—the probability of an outcome in E should be the same under $A(\mathbf{x})$ and $A(\mathbf{x}')$, up to a small multplicative factor. In other words, the algorithm's outcomes should be about the same *whether or not individual i 's real data was used.*

Definition 3.1 (ε-DP with fixed-size data sets)**.** A randomized algorithm $A : \mathcal{U}^n \to \mathcal{Y}$ taking inputs in $\mathcal{U}^n$ is *ε-differentially private for size n data sets* if, for every pair of neighboring data sets $\mathbf{x}, \mathbf{x}'$, for all events[3] $E \subseteq \mathcal{Y}$:

$$\mathbb{P}\left(A(\mathbf{x}) \in E\right) \leq e^{\varepsilon} \cdot \mathbb{P}\left(A(\mathbf{x}') \in E\right). \tag{2}$$

The definition of DP uses the parameter ε to control how far apart the distributions of $A(\mathbf{x})$ and $A(\mathbf{x}')$ can be. For example, Figure 3 depicts two distributions that satisfy the criterion of Equation (2) with $\varepsilon = 1/4$. As ε gets smaller, the algorithm's output distributions can vary less. When $\varepsilon = 0$, the algorithm leaks nothing at all—its output distribution must be the same for all inputs.

In earlier sections, we pretty much already proved that randomized response (Alg. 2) is differentially private, without using that terminology. Let's go through the argument again, filling in the missing pieces.

Proposition 3.2. *RR_ε is ε-differentially private.*

Proof. Fix two neighboring data sets $\mathbf{x}$ and $\mathbf{x}'$, and let i be the position in which they differ (so that $x_i \neq x'_i$ but $x_j = x'_j$ for all $j \neq i$) . First, consider a particular outcome $\mathbf{y} = (y_1, ..., y_n)$. Because we make selections independently for each i, we have

$$\mathbb{P}\left(RR_\varepsilon(\mathbf{x}) = \mathbf{y}\right) = \mathbb{P}\left(Y_1 = y_1 \mid x_1\right) \cdot \mathbb{P}\left(Y_2 = y_2 \mid x_2\right) \cdots \mathbb{P}\left(Y_n = y_n \mid x_n\right) \tag{3}$$

When we compare this to the probability that $RR_\varepsilon(\mathbf{x}') = \mathbf{y}$, only one of the terms in the product will change. We thus get that

$$\frac{\mathbb{P}\left(RR_\varepsilon(\mathbf{x}') = \mathbf{y}\right)}{\mathbb{P}\left(RR_\varepsilon(\mathbf{x}) = \mathbf{y}\right)} = \frac{\mathbb{P}\left(Y_i = y_i \mid x'_i\right)}{\mathbb{P}\left(Y_i = y_i \mid x_i\right)} \tag{4}$$

This ratio is at most $\frac{e^\varepsilon}{e^\varepsilon+1} \Big/ \frac{1}{e^\varepsilon+1} = e^\varepsilon$.

Now let's take any subset $E \subseteq \mathcal{Y} = \{0,1\}^n$. The probability that $RR_\varepsilon(\mathbf{x})$ lies in E is just the sum over $\mathbf{y} \in \mathbf{E}$ of the probability that $RR_\varepsilon(\mathbf{x}) = \mathbf{y}$. We thus get

$$\mathbb{P}\left(RR_\varepsilon(\mathbf{x}) \in E\right) = \sum_{\mathbf{y} \in E} \mathbb{P}\left(RR_\varepsilon(\mathbf{x}) = \mathbf{y}\right) \overset{Eq.\ (4)}{\leq} \sum_{\mathbf{y} \in E} e^\varepsilon \cdot \mathbb{P}\left(RR_\varepsilon(\mathbf{x}') = \mathbf{y}\right) = e^\varepsilon \cdot \mathbb{P}\left(RR_\varepsilon(\mathbf{x}') \in E\right). \tag{5}$$

[3] In this course, it is generally fine to think of an event as any subset of the output set. In general, for uncountable output sets like $\mathbb{R}$, one restricts attention to a collection of "measurable" sets. Standard texts on probability discuss the issue in detail.

This completes the proof. □

The proof that randomized response is differentially private uses a useful trick that is true quite generally:

Exercise 3.3. Show that if the output space $\mathcal{Y}$ is discrete (so probabilities are just sums over individual elements), then an algorithm $A : \mathcal{U}^n \to \mathcal{Y}$ ε-DP if and only if for every particular output $a \in \mathcal{Y}$, we have $\mathbb{P}(A(\mathbf{x}) = a) \leq e^{\varepsilon}\mathbb{P}(A(\mathbf{x}') = a)$ (that is, neighboring data sets lead to each individual output with about the same probability). Similarly, if the distributions of $A(\mathbf{x})$ and $A(\mathbf{x}')$ both have probability densities (on $\mathbb{R}$, say), show that it suffices to have $f_{\mathbf{x}}(y) \leq e^{\varepsilon} f_{\mathbf{x}'}(y)$ for all possible outptus y, where $f_{\mathbf{x}}(y)$ and $f_{\mathbf{x}'}(y)$ are the two probability densities.

Exercise 3.4. (i) Suppose $\mathbf{x}$ and $\mathbf{x}'$ are neighbors. Let A be a randomized algorithm that is ε-differentially private for $\varepsilon = \ln(5/4) \approx 0.223$. Suppose A outputs real numbers, and suppose $\mathbb{P}(A(\mathbf{x}) \geq 14) = 0.2$. What range of values for $\mathbb{P}(A(\mathbf{x}') \geq 14)$ is possible? (ii) How would the answer change if you knew instead that $\mathbb{P}(A(\mathbf{x}) \geq 14) = 0.5$? [*Hint:* Consider the constraints on $\mathbb{P}(A(\mathbf{x}) < 14) = 1 - \mathbb{P}(A(\mathbf{x}) \geq 14)$.]

3.1 Discussion

There are a few points worht highlighting about the definition before we dive into more examples.

First, differential privacy is a condition on the *algorithm*. It's not that we can point to a particular table or plot that was output and say "that's differentially private". We have to understand the process that mapped the data to the output in order to reason with any confidence.

Second, the parameter ε measures how much is leaked about (the presence of) any particular data point. You should think of it as small, but not too small. Setting ε to 1 or 0.1 or even 0.001 (if the data set size n is large enough) makes sense, but setting it to 2^{-100} would make it impossible to do anything useful. We'll return to this point in the future.

Finally, it is important to understand that this is not the *only* useful notion of privacy. In fact, even if you like the general framework, there are quite a few variations on the definitions, some of which we will discuss in this course.

One aspect in which definitions might differ is the way in which they measure how close the output distributions are on neighboring data sets (like *approximate differential privacy* or *concentrated differential privacy*). It can be tricky to think about which variations of the framework are most meaningful. We'll try to get a handle on that issue in Lecture 05.

Another source of variation comes from what data sets we take to be neighbrong. For example, if the data set consists of the graph of freindships in an online social network, where nodes are people and edges represent friendships, then "changing one person's data" might mean changing all the edges touching some node. This feels a bit weird since changing the data for node x will change the data for other nodes y (since the edge (x, y) is "about" both x and y). Exploring the meaning of such definitions and how to design algorithms that satisfy them might make a good course project!

4 A Second Example: The Laplace Mechanism

Another natural way to add randomness to a computation is to simply add noise to the output of some function f evaluated on the data. This function could just return a single real number (like a proportion or a sum), or it could be something more complex that returns a vector in $\mathbb{R}^d$ (such as the roughly 3 billion statistics produced by the US Census Bureau from its decennial census).

When does adding noise satisfy differential privacy? How does the choice of the function f we evaluate affect the amount of noise we must add? One basic idea is to look at how sensitive a function is to a change in one of its input records. We measure this via the *global sensitivity* of f:

Definition 4.1. Given a function $f : \mathcal{U}^n \to \mathbb{R}^d$, we define the *global sensitivity of f in the ℓ_1 norm* to be

$$GS_{f,\ell_1} = \sup_{\mathbf{x},\mathbf{x}' \text{ neighbors in } \mathcal{U}^n} \|f(\mathbf{x}) - f(\mathbf{x}')\|_1 . \tag{6}$$

(We often drop the subcript ℓ_1, and write simply GS_f.)

For some functions f, it is tricky to get our hands on the exact global sensitivity, and it is easier to work with an upper bound. A gunction has global sensitivity at most Δ (in the ℓ_1 norm) if for all pairs of neighboring data sets $\mathbf{x}, \mathbf{x}' \in \mathcal{U}^n$:

$$\|f(\mathbf{x}) - f(\mathbf{x}')\|_1 \leq \Delta . \tag{7}$$

The notation $\|\cdot\|_1$ refers to the ℓ_1 norm of a vector, which is sum of the absolute values of the vector's entries. For example, $\|(1, 0, -3)\|_1 = 4$, and $\|(1, 1, 1, 1, 1, 1)\|_1 = 6$. In 1 dimension, the ℓ_1 norm is just the absolute value.

Examples of global sensitivity A proportion $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \varphi(x_i)$, where $\varphi : \mathcal{U} \to \{0, 1\}$, has global sensitivity $GS_f = \frac{1}{n}$. The same is true if φ maps records to numbers in the interval $[0, 1]$.

To take another example, consider a *histogram*: given a data set $\mathbf{x} \in \mathcal{X}^n$ and a partition of $\mathcal{U}$ into d disjoint sets $B_1, ..., B_d$ (think of these as "bins" or "types" of items in $\mathcal{U}$), we count how many records there are of each type. $f(\mathbf{x}) = (n_1, n_2, ..., n_d)$ where $n_j = \#\left\{i : x_i \in B_j\right\}$. So, for example, if we wanted to compute the number of residents of each of the 50 US states from a census of the US population, we would be asking a hstogram query. The global sensitivity of a histogram query is at most 2, regardless of how many bins there are.

Exercise 4.2. Suppose we want to compute the average of a set of numbers known to lie in the interval $[0, R]$ on a data set of size n. What is the sensivitiy of the average?

The Laplace Mechanism If f has sensitivity Δ, we can satisfy ε-DP by adding noise from a *Laplace distribution* with scale $\frac{\Delta}{\varepsilon}$, independently to each entry of the output. The Laplace distribution, also called the double exponential distribution, is sort of a pointy Gaussian (Fig.4). The mean-0, scale-1 Laplace has density $h(y) = \frac{1}{2}e^{-|y|}$ for $y \in \mathbb{R}$. This distribution has expected absolute value 1, and standard deviation $\sqrt{2}$. We can scale the distribution by a positive number $\lambda > 0$, to get the general form $\mathsf{Lap}(\lambda)$ with density

$$\mathsf{Lap}(\lambda) : \quad \text{a distribution on } \mathbb{R} \text{ with p.d.f. } h_\lambda(y) = \frac{1}{2\lambda}\exp(-|y|/\lambda)$$

Figure 4 illustrates the probability density function for a few values of λ. If we translate the distribution to have mean μ, then the density becomes $\frac{1}{2\lambda}\exp(-|y - \mu|/\lambda)$.

The resulting algorithm (Algorithm 3) is called the Laplace mechanism[4], and is a basic building block for the design of many other differentially private algorithms. If used to estimate a proportion, it produces far more accurate estimates than randomized response (for the same privacy budget)—see Figure 5 and Exercise 4.6.

Note that instead of the actual global sensitivity GS_f, we may use any upper bound $\Delta \geq GS_f$.

[4]Here and elsehwere in the course, the term "mechanism", inherited from literature on game theory, just means "algorithm".

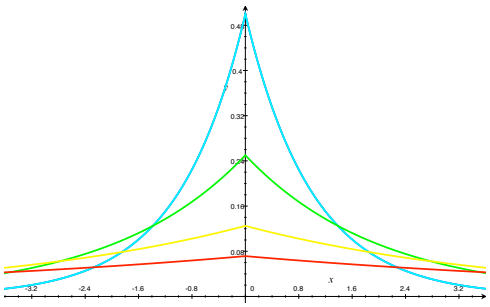


Figure 4: The probability density function of the Laplace distribution with $\lambda \in \{1, 2, 4, 7\}$.

Algorithm 3: Laplace mechanism $A_{\mathsf{Lap}}(\varepsilon, \mathbf{x})$

Input: Data set $\mathbf{x} = (x_1, ..., x_n) \in \mathcal{U}^n$ and parameter $\varepsilon > 0$.

1 Receive a query $f : \mathcal{U}^n \to \mathbb{R}^d$. Let GS_f denote its global sensitivity in the ℓ_1 norm;

2 **return** $f(\mathbf{x}) + (Z_1, ..., Z_d)$ where $Z_i \sim \mathsf{Lap}\left(\frac{GS_f}{\varepsilon}\right)$ are i.i.d.

Theorem 4.3. *The Laplace mechanism is ε-differentially private.*

Proof. Fix two neighboring data sets $\mathbf{x}$ and $\mathbf{x}'$ in $\mathcal{U}^n$, and a query $f : \mathcal{U}^n \to \mathbb{R}^d$. Let $\Delta = GS_f$ be the ℓ_1 global sensitivity of f. Let $\mu = f(\mathbf{x})$ and $\mu' = f(\mathbf{x}')$. We know that the ℓ_1 norm of $\mu - \mu'$ is at most Δ. Comparing the output distributions of A_{Lap} on $\mathbf{x}$ and $\mathbf{x}'$ thus means comparing two Laplace distributions that have been shifted relative to one another by $\mu - \mu'$.

Because we add noise independently to each entry of the output, the density of the output at vector $\mathbf{y}$ on input $\mathbf{x}$ can be written as a product:

$$h_{\mathbf{x}}(\mathbf{y}) = \tfrac{\varepsilon}{2\Delta} e^{-\frac{\varepsilon}{\Delta}|y_1 - \mu_1|} \times \tfrac{\varepsilon}{2\Delta} e^{-\frac{\varepsilon}{\Delta}|y_2 - \mu_2|} \times \cdots \times \tfrac{\varepsilon}{2\Delta} e^{-\frac{\varepsilon}{\Delta}|y_d - \mu_d|}, \tag{8}$$

which can be simplified to $h_{\mathbf{x}}(\mathbf{y}) = \left(\frac{\varepsilon}{2\Delta}\right)^d e^{-\frac{\varepsilon}{\Delta}\|\mathbf{y} - \mu\|_1}$. If we look at the ratio of the densities for $\mathbf{x}$ and $\mathbf{x}'$ at the same output $\mathbf{y}$, we get

$$\frac{h_{\mathbf{x}'}(\mathbf{y})}{h_{\mathbf{x}}(\mathbf{y})} = e^{-\frac{\varepsilon}{\Delta}(\|\mathbf{y} - \mu'\|_1 - \|\mathbf{y} - \mu\|_1)}. \tag{9}$$

By the triangle inequality, the difference $\|\mathbf{y} - \mu\|_1 - \|\mathbf{y} - \mu'\|_1$ is at most $\|\mu - \mu'\|_1$, which is at most Δ. We thus get:

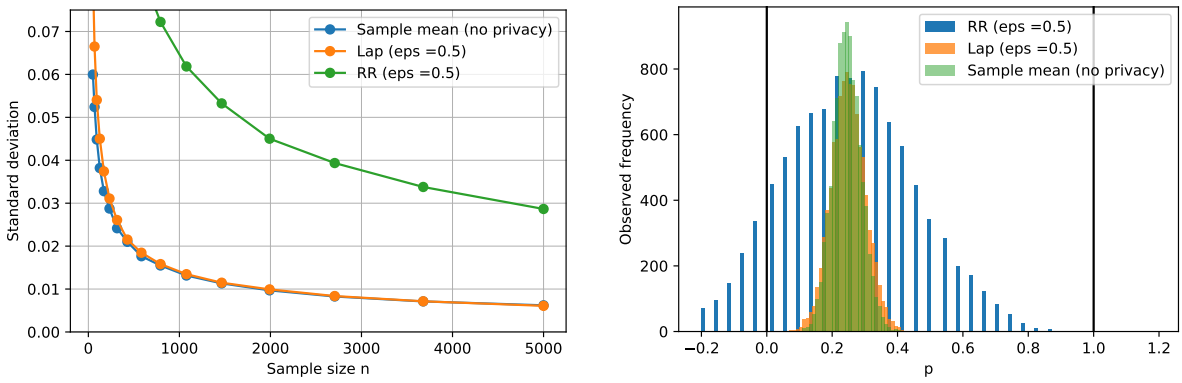
$$\frac{h_{\mathbf{x}'}(\mathbf{y})}{h_{\mathbf{x}}(\mathbf{y})} \leq e^{\frac{\varepsilon}{\Delta}\|\mu - \mu'\|_1} \leq e^{\frac{\varepsilon}{\Delta} \cdot \Delta} = e^{\varepsilon}. \tag{10}$$

And that is enough to conclude the mechanism is ε-DP: For any measurable set E, we have $\Pr(A_{\mathsf{Lap}}(\varepsilon, \mathbf{x}) \in E = \int_{\mathbf{y} \in E} h_{\mathbf{x}}(\mathbf{y})$. So if the ratio of the densities is bounded everywhere by e^{ε}, then so is the ratio of the probabilities of any given event E. □

Thus, we can guarantee differential privacy by adding noise to the output of a function that scales with the function's sensitivity. Histograms, for example, fit the framework well: we can get away with adding error whose expected magnitude is $2/\varepsilon$ to each of the bin counts, regardless of the number of bins.

The following lemma provides useful bounds on the magnitude of the Laplace mechanism's error.

Lemma 4.4.



(*a*) (*b*)

Figure 5: (*a*) The standard deviation of several different algorithms for estimating the bias of a coin from n independent flips of the coin. The non-private algorithm simply reports the sample mean (i.e., the proportion of 1's in the data). The other two curves are generated by applying randomized response and the Laplace mechanism with $\varepsilon = 0.5$. The data are generated from a coin with bias 0.25. The standard deviations were computed from 10,000 independent runs of each algorithm, at sample sizes ranging from 50 to 5,000. (*b*) The empirical distribution from 10,000 executions of the three algorithms with a sample size of $n = 100$, using $\varepsilon = 0.5$. Note that randomized response (using the simplest post-processing) generates estimates outside of $[0, 1]$.

1. *If* $Z \sim \text{Lap}(\lambda)$ *is a Laplace-distributed random variable, we have*
   (*a*) $\mathbb{E}(|Z|) = \lambda$
   (*b*) $\sqrt{\mathbb{E}(Z^2)} = \sqrt{2}\lambda$
   (*c*) *For every* $t > 0$: $\mathbb{P}(|Z| > t\lambda) \leq \exp(-t)$.
2. *Let* $Z_1, ..., Z_d$ *are i.i.d.* $\text{Lap}(\lambda)$ *random variables, and let* $M = \max(|Z_1|, ..., |Z_d|)$.
   (*a*) $\mathbb{E}(\|(Z_1, ..., Z_d)\|_1) = d\lambda$
   (*b*) *For every* $t > 0$: $\mathbb{P}(M > \lambda(\ln(d) + t)) \leq \exp(-t)$.
   (*c*) $\mathbb{E}(M) \leq \lambda(\ln(d) + 1)$.

Exercise 4.5. Suppose we use the Laplace mechanism to estimate the number of individuals in a data set who reside in each of the 3,143 counties[5] in the US, using parameter $\varepsilon = 0.1$. What does Lemma 4.4 imply about the expected error of the count for Suffolk County, Mass.? What does it imply about the expectation of the largest error in the estimate of any county population?

Exercise 4.6. Suppose we use the Laplace mechanism to estimate the fraction $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\varphi(x_i)$ where $\varphi : \mathcal{U} \to [0, 1]$ is a predicate. Give an expression for the data set size $n(\alpha, \varepsilon)$ at which the Laplace mechanism's root mean square error $\sqrt{\mathbb{E}\left((A(\mathbf{x}) - f(\mathbf{x}))^2\right)}$ drops below α (when run with privacy parameter ε). How does this compare to the analogous calculation for Randomized Response (Exercise 2.6)? What happens to $n(\alpha, \varepsilon)$ when α is halved? When ε is halved?

Notice that the accuracy of the Laplace mechanism is pretty bad when ε is very small. Suppose we want to estimate an fraction (as in the Exercise 4.6). If we set $\varepsilon = 1/n$, then the standard deviation of the Laplace mechanism is $\sqrt{2} \cdot \frac{\Delta}{1/n} = \sqrt{2}$ (since $\Delta = 1/n$ in this case). But the fraction can only take values

[5]This number includes county equivalents, and was drawn from the Wikipedia article "List of United States counties and county equivalents" in February 2021.

between 0 and 1—the noise is thus of larger magnitude than the "'signal" one is trying to release. This feature is inherent. As we will see next lecture, differentially private algorithms cannot yield any useful information when $\varepsilon < 1/n$.

Exercise 4.7. Prove Lemma 4.4. To bound $\mathbb{E}(M)$ in the final statement, it may be useful to recall that for any nonnegative random variable M, we have $\mathbb{E}(M) = \int_{x=0}^{\infty} \Pr(M > x)$. This inequality allows us to compute expectations in terms of "tail bounds".

Summary

4.1 Key Points

- To reason about information leakage and confidentiality we must look at the algorithms that process the data, not only the form of the output.
- Differential privacy is one way to quantify how much an algorithm leaks about individual inputs. It is parametrized by a positive real number ε, which bounds the amount of leakage.
- Randomized response and the Laplace mechanism satisfy ε-DP. For the same privacy budget, the Laplace mechanism adds far less error.

Additional Reading and Watching

- The paper that defined differential privacy [DMNS06, DMNS16]
- A nontechnical introduction to DP [WAB+18]
- Dwork and Roth book, Chapter 2 [DR14]
- Videos from the MinutePhysics Youtube channel: "Protecting Privacy with MATH" and "When It's OK to Violate Privacy", 2019.
- Tutorial talks by K. Ligett (NeurIPS 2016 Tutorial) and A. Smith (NASIT 2019 Tutorial).
- For further discussion of composition attacks, see [GKS08]
- For an attack on a specific k-anonymized data set, see [Coh22]

References

[Coh22] Aloni Cohen. Attacks on deidentification's defenses. In *31st USENIX Security Symposium (USENIX Security 22)*, August 2022. USENIX Association.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Conference on Theory of Cryptography*, TCC '06, 2006.

[DMNS16] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3), 2016.

[DR14] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. NOW Publishers, 2014.

[GKS08] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, 2008. ACM.

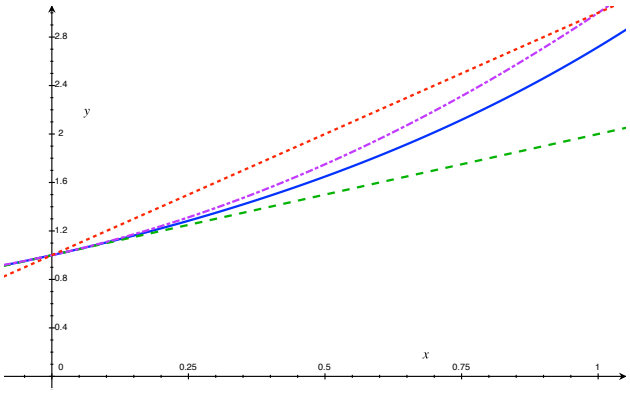


Figure 6: The function e^x (solid blue), seen here bounded below by $1 + x$ (green dashed) and bounded above on $[0, 1]$ by $1 + x + x^2$ (purple dashed) and $1 + 2x$ (red dashed).

[NS08] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, 2008.

[Swe02] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[WAB+18] Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, James Honaker, Kobbi Nissim, David O'Brien, Thomas Steinke, and Salil Vadhan. Differential privacy: A primer for a non-technical audience. *Vanderbilt Journal of Entertainment & Technology Law*, 21(1):209, 2018.

[War65] Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

Appendix

A The function e^x

We'll be working with the quantity e^x (often e^ε for DP algorithms) a lot. Here a few useful inequalities:

- For all $x \in \mathbb{R}$, we have $e^x > 1 + x$ (and thus $e^{-x} \geq 1 - x$).
- As $x \to 0$ (either positive or negative), we have $e^x = 1 + x + \Theta(x^2)$. As a consequence, we have:
  - $x \geq 1 - e^{-x} \geq x - O(x^2)$,
  - $\frac{1}{x} \geq \frac{1}{e^x - 1} \geq \frac{1}{x} - O(x^2)$, and
  - $\frac{1}{x} \leq \frac{1}{1 - e^{-x}} \leq \frac{1}{x} + O(x^2)$.

You can double check the direction of inequalities and a sense of specific constants by using a graphing app. For example, Figure 6 shows that $1 + x \leq e^x \leq 1 + x + x^2 \leq 1 + 2x$ for $x \in [0, 1]$.