Chapter 2 — Differential Privacy Fundamentals

Draft version: September 13, 2021

Contents

Acknowledgement. This book chapter is an extension from the course material jointly developed by Jonathan Ullman and Adam Smith. Please feel free to provide feedback.

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 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

[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.

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.

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 biased random bit Y_i as follows. With probability $3/4$, we set $Y_i = x_i$, and with the remaining probability of $1/4$, we set Y_i to be the 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

Algorithm 1 Randomized Response, RR_{basic}

Input: Data set of n bits: $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$
Output: Bits $Y_1, \ldots, Y_n$
For $i = 1$ to n

$$Y_i = \begin{cases} x_i & \text{w.p. } 3/4 \\ 1 - x_i & \text{w.p. } 1/4 \end{cases}$$

Return $(Y_1, \ldots, Y_n)$

any particular output y_i, we have:

$$\frac{1}{3} \leq \frac{\mathbb{P}(Y_i = y_i \mid x_i = 1)}{\mathbb{P}(Y_i = y_i \mid x_i = 0)} \leq 3 \tag{1}$$

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 $Y_1, \ldots, Y_n$ from randomized response on input $x_1, \ldots, x_n$, 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 3 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: $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{U}^n$, predicate $\varphi : \mathcal{U} \to \{0,1\}$, and a privacy parameter $\varepsilon > 0$
Output: Bits $Y_1, \ldots, Y_n$
For $i = 1$ to n

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

Output: $(Y_1, \ldots, Y_n)$

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 satisfy φ.

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

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

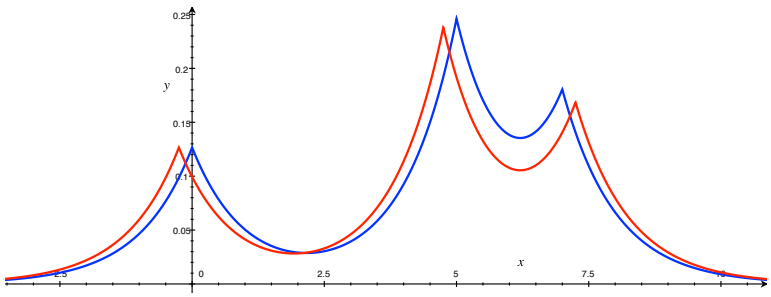

Figure 3: Two distributions P and Q that satisfy: for every event E, $P(E) \leq e^{1/4}Q(E)$ and $Q(E) \leq e^{1/4}P(E)$.

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 $\mathbf{y} = (y_1, \ldots, y_n)$. 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 $\frac{e^\varepsilon}{e^\varepsilon+1} \Big/ \frac{1}{e^\varepsilon+1} = e^\varepsilon$.

[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.

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

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

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 $\mathbb{P}\left(A(\mathbf{x}) = a\right) \leq e^\varepsilon \mathbb{P}\left(A(\mathbf{x}') = a\right)$ (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 $\mathbb{P}\left(A(\mathbf{x}) \geq 14\right) = 0.2$. What range of values for $\mathbb{P}\left(A(\mathbf{x}') \geq 14\right)$ is possible? (ii) How would the answer change if you knew instead that $\mathbb{P}\left(A(\mathbf{x}) \geq 14\right) = 0.5$? [*Hint:* Consider the constraints on $\mathbb{P}\left(A(\mathbf{x}) < 14\right) = 1 - \mathbb{P}\left(A(\mathbf{x}) \geq 14\right)$.]

4 Interpreting DP: Smoking, Cancer, and Correlations

What does it mean to decide if a concept like differential privacy is a good definition of "privacy"? There is no single answer, since it involves a connection between an unambiguous mathematical concept and a nebulous social one. "Privacy" covers lots of different concepts, many of which are more about control than confidentiality, and all of which are context-dependent.[4] Nevertheless, we can try to wrap our heads around the guarantee that a technical concept provides—perhaps we can chip off a piece of "privacy" which is accurately pinned down by DP.

How can we start? A good exercise is to write down an natural-language sentence that captures the type of guarantee we would like. A strong requirement, reminiscent of what is possible for encryption would be this:

> **A first attempt:** No matter what they know ahead of time, the attacker's beliefs about Alice are the same after they see the output as they were before.

Unfortunately, such a strong guarantee is impossible to achieve if we actually want to release useful information. To see why, consider the example of a clinical study that explores the relationship between smoking and lung disease. A health insurance company with no *a priori* understanding of that relationship might, after seeing the results of the study, dramatically alter its estimates of different people's likelihood of disease. In turn, this would likely cause the company to raise premiums for smokers and lower them for nonsmokers. The conclusions drawn by the company about the riskiness of any one individual (say Alice) are strongly affected by the results of the study. Their beliefs about Alice have definitely changed.

However, the change can hardly be called a breach of Alice's privacy. It happens because the study reveals a feature of human biology—exactly what we want clinical studies to do!

So what can we hope to achieve? One important observation about the smoking and lung disease example is that the information about Alice would be learned by the insurance company regardless of whether Alice participated in the study. In other words, the conclusions the insurance company draws about Alice come from the totality of the data set, and don't depend strongly on her data. One way to understand differential privacy is that this is the only kind of inference about individuals that it allows.

> **The DP principle:** No matter what they know ahead of time, an attacker seeing the output of a differentially private algorithm would draw (almost) the same conclusions about Alice *whether or not her data were used.*

It is instructive to formalize this intuitive statement. What do we mean by "what the attacker knows" and "what they learn"? We'll adopt what statisticians call a Bayesian perspective, and encode knowledge via probability distributions. Specifically, let's think of the data set as a random variable $\mathbf{X}$ distributed over $\mathcal{U}^n$. For clarity, we'll use capital letters like $\mathbf{X}$ to refer to random variables, and lower case symbols like $\mathbf{x}$ to refer to specific realizations.

We can the adversary's background knowledge via a *prior distribution* $p(\mathbf{x}) = \mathbb{P}(\mathbf{X} = \mathbf{x})$. We should think of this as how likely a given data set is to occur given everything the attacker knows

[4] A number of writers have dissected the concept, trying to provide their own taxonomy of privacy's many facets. Brandeis [?], Solove [?], and Nissenbaum [?] provide good places to start.

ahead of time. [5] Because we don't know what other information the attacker has, we will want our analysis to work for *every* prior distribution p.

Given the output $a = A(\mathbf{X})$. We can model "what the adversary learns" by the *posterior distribution* of the data conditioned on the algorithm's output. That is,

$$p(\mathbf{x} \mid a) \stackrel{\text{def}}{=} \mathbb{P}\left(\mathbf{X} = \mathbf{x} \mid A(\mathbf{X}) = a\right) = \frac{\mathbb{P}\left(A(\mathbf{x}) = a\right) \cdot p(\mathbf{x})}{\sum_{\tilde{\mathbf{x}} \in \mathcal{U}^n} \mathbb{P}\left(A(\tilde{\mathbf{x}}) = a\right) \cdot p(\tilde{\mathbf{x}})}\,. \tag{6}$$

But how should we model "what the attacker would have learned had person i's data been removed"? Given a data set $\mathbf{x} \in \mathcal{U}^n$, let $\mathbf{x}_{-i}$ denote the data set in which person i's entry has been replaced by a default value. Consider a hypothetical world in which the data set $\mathbf{x}_{-i}$ is used instead of the real data set $\mathbf{x}$. Given an output a, we can now consider the conditional distribution $p_{-i}(\cdot \mid a)$ that the attacker would have constructed in the hypothetical world, namely:

$$p_{-i}(\mathbf{x} \mid a) \stackrel{\text{def}}{=} \mathbb{P}\left(\mathbf{X} = \mathbf{x} \mid A(\mathbf{X}_{-i}) = a\right) = \frac{\mathbb{P}\left(A(\mathbf{x}_{-i}) = a\right) \cdot p(\mathbf{x})}{\sum_{\tilde{\mathbf{x}} \in \mathcal{U}^n} \mathbb{P}\left(A(\tilde{\mathbf{x}}_{-i}) = a\right) \cdot p(\tilde{\mathbf{x}})}\,. \tag{7}$$

We can think of $p_{-i}(\cdot \mid a)$ as encoding what the attacker would have learned about person i had person i's data never been used.

To formalize our claim about differential privacy, we'll use the following shorthand. For two distributions p and q on the same set $\mathcal{Y}$ (technically, over the same σ-algebra of events), we'll write

$$p \approx_\varepsilon q \qquad \Leftrightarrow \qquad (\forall \text{ events } E \subseteq \mathcal{Y} \;:\; p(E) \le e^\varepsilon q(E) \text{ and } q(E) \le e^\varepsilon p(E))\,. \tag{8}$$

Given two random variables A and B distributed over the same set, we'll sometimes abuse notation and write $A \approx_\varepsilon B$ to mean that the relation in (8) is satisfied by their distributions. With this notation, an algorithm A is ε-DP if and only if, for every pair of neighboring data sets $\mathbf{x}$ and $\mathbf{x}'$, we have $A(\mathbf{x}) \approx_\varepsilon A(\mathbf{x}')$.

Theorem 4.1. *Let $A : \mathcal{U}^n \to \mathcal{Y}$ be ε-differentially private. For every distribution on $\mathbf{X}$ (possibly with dependencies among the entries), for every output $a \in \mathcal{Y}$, for every index i, we have*

$$p_{-i}(\cdot \mid a) \approx_{2\varepsilon} p(\cdot \mid a)\,. \tag{9}$$

Exercise 4.2. Prove Theorem 4.1. *Hint:* Fix an output $a \in \mathcal{Y}$. Given a data set $\mathbf{x} \in \mathcal{U}^n$, how can you write the ratio $\frac{p_{-i}(\mathbf{x}|a)}{p(\mathbf{x}|a)}$ in terms of the ratios of the form $\frac{\mathbb{P}(A(\mathbf{x}_{-i})=a)}{\mathbb{P}(A(\mathbf{x})=a)}$?

Something here might seem weird: how can the attacker learn about i's data from a if x_i was not used to compute a? The answer is in the dependencies among the data records—the attacker can learn about $\mathbf{x}_{-i}$, which itself reveals information about x_i.

Returning to the smoking and lung disease example: Suppose the records in $\mathbf{X}$ are drawn i.i.d. from one of several possible distributions. For simplicity, imagine there are two possible distributions, one where the features are independent, and one where they are strongly correlated,

[5] Our use of probability to model knowledge this way corresponds to the *subjective interpretation* of probability (see, e.g., [Háj19]). It's pretty different from the way we use probability in the definition of a randomized algorithm, or in the definition of differential privacy. In those contexts, the probabilities reflect a process we control, and it's reasonable to think of them as known exactly. In contrast, we cannot expect to know an attacker's prior. Here, we posit only that it exists. Even this postulate is delicate, especially since real attackers are computationally bounded. We ignore computational restrictions here for simplicity.

so that the prior on $\mathbf{X}$ is a mixture of the two. Seeing the clinical study's results basically causes the insurance company's posterior to collapse to the i.i.d. distribution in which the features are correlated. Whether the study used Alice's data or not, the insurance company's posterior distribution on Alice's record would have the features correlated.

What have we learned? We can model knowledge via probabilities, and learning via the change from prior to posterior distributions. When we do that, we can make our intuition precise—-that differentially private mechanisms reveal only information that could be learned without any particular person's data.

We've also found a useful natural language formulation of our goal when thinking about confidentiality of individuals' data when releasing aggregate statistics. That type of formulation is particularly useful since it can guide our intuition for the technical concepts. It can also help us articulate goals in legal and policy discussions.

4.1 A not-so-great variation on differential privacy

The formulation of Theorem 4.1 also helps us distinguish among similar definitions of privacy.

Suppose we were to require that probabilities differ by an additve error term rather than a multiplicative one. We might say that a randomized algorithm satisfies "δ-additive secrecy" if

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{U}^n \text{ neighbors }, \ \forall \text{ events } E \ : \ \mathbb{P}\left(A(\mathbf{x}) \in E\right) \leq \mathbb{P}\left(A(\mathbf{x}') \in E\right) + \delta \, . \tag{10}$$

How different is this from differential privacy? It certainly has things in common: for example, it is closed under composition and postprocessing, and satisfies a similar version of group privacy. In particular, we must have $\delta > 1/n$ to get useful information out of such an algorithm. However, it does *not* satisfy a reasonable analogue to Theorem 3.1, and it *does* allow some algorithms that are pretty obviously disclosive.

Exercise 4.3 (Name and Shame Mechanism)**.** Consider the following mechanism NS_δ. On input $\mathbf{x} = (x_1, \ldots, x_n)$, for each i from 1 to n, it generates

$$Y_i = \begin{cases} (i, x_i) & \text{w. prob. } \delta \, , \\ \perp & \text{w. prob. } 1 - \delta \, . \end{cases} \tag{11}$$

Here $\perp$ is just a special symbol meaning "no information".

(i) Show that NS_δ satisfies "δ-additive secrecy". (ii) Show that for $\delta \gg 1/n$, the mechanism publishes some individuals' data in the clear with high probability, and that for such outputs, Eq. (9) in Theorem 4.1 does not hold.

Summary

Key Points

- Differentially private algorithms can be assembled modularly, or run independently by different organizations. The privacy parameter accumulates at most additively across all executions that use the same person's record.
- We can view the privacy parameter as a budget to be divided among different efforts.

- For some algorithms, one gets a much better analysis by considering the steps jointly, rather than using composition. (Exercise **??**)
- Algorithms that access their data using summation queries can often be made differentially private without too much loss of accuracy. We saw the example of Lloyd's algorithm.
- Useful statistical summaries may have to reveal information about an individual to an attacker. However, we can make a more subtle claim: No matter what they know ahead of time, an attacker seeing the output of a differentially private algorithm would draw (almost) the same conclusions about Alice *whether or not her data were used*.

Additional Reading and Watching

- More on the formulation of Theorem 4.1: [KS14]
  - MinutePhysics' Youtube video "When It's OK to Violate Privacy", 2019.
- A thorough proof of the impossibility of the "first attempt" privacy guarantee: [DN10] (see also [KM11]).
- Why noisy sums can be used to find useful approximations to many natural procedures: [Kea93, BDMN05, DMNS16].

References

[BDMN05] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In *Proceedings of the 24th Annual ACM Symposium on Principles of Database Systems*, PODS '05, 2005. ACM.

[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.

[DN10] Cynthia Dwork and Moni Naor. On the difficulties of disclosure prevention, or the case for differential privacy. *Journal of Privacy and Confidentiality*, 2(1), 2010.

[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.

[Háj19] Alan Hájek. Interpretations of Probability. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2019 edition, 2019.

[Kea93] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. In *ACM Symposium on Theory of Computing*. ACM, 1993.

[KM11] Daniel Kifer and Ashwin Machanavajjhala. No Free Lunch in Data Privacy. In *SIGMOD*, 2011.

[KS14] Shiva Prasad Kasiviswanathan and Adam D. Smith. On the 'semantics' of differential privacy: A bayesian formulation. *Journal of Privacy and Confidentiality*, 2014.

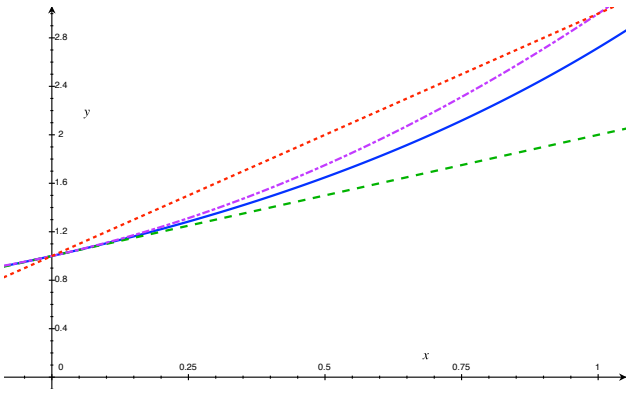

Figure 4: The function e^x (solid blue), seen here bounded below by $1+x$ (green dashed) and bounded above on $[0,1]$ by $1+x+x^2$ (purple dashed) and $1+2x$ (red dashed).

[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.

[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 4 shows that $1 + x \leq e^x \leq 1 + x + x^2 \leq 1 + 2x$ for $x \in [0, 1]$.