15 Differential Privacy

This section is based heavily on the course Privacy in Statistics and Machine Learning by Adam Smith and Jonathan Ullman, primarily Lectures 4 and 5 (Spring 2023) [Smi23a, Smi23b].

Now that we've seen membership inference attacks, we need to come up with a definition of privacy that prevents them. But of course we do not want a definition that merely prevents a specific attack, or a class of attacks, because the history of privacy and security is littered with examples of unanticipated attacks. What is remarkable is that we will see that preventing membership inference attacks, in a sufficiently strong way, actually gives a very general purpose privacy definition that gives protections against any possible attack in a strong sense, even ones we haven't come up with yet. The key intuition is:

> *If the attacker cannot even tell whether a particular individual is a member of the dataset, then they cannot learn anything more about that individual than they could have learned had that individual's data had never been collected.*

In the definition. As in our other models, we let $\mathcal{X}$ be a domain of possible elements of the dataset, and a dataset $S = \{X_1, \ldots, X_n\} \in \mathcal{X}^n$ is a collection of n elements from this domain. Two datasets S and S' are said to be *neighboring* if they differ on at most one element. That is, for some $i \in [n]$:

$$S = \{X_1, \ldots, X_i, \ldots, X_n\}, \qquad S' = \{X_1, \ldots, X_i', \ldots, X_n\}.$$

Definition 15.1 ([DMNS06]). A randomized algorithm $\mathcal{A} \colon \mathcal{X}^n \to \mathcal{R}$ is *ε-differentially private* if for every pair of neighboring samples S, S', and every event $R \subseteq \mathcal{R}$,

$$\mathbb{P}[\mathcal{A}(S) \in R] \leq e^{\varepsilon}\, \mathbb{P}[\mathcal{A}(S') \in R].$$

There are a few key aspects of this definition to focus on, and we will briefly explain the parts and what role they play:

- Differential privacy is a property of an *algorithm* not an *output*. There is no particular output that is or is not private, instead, privacy is a property of what information the algorithm's output reveals about its inputs. This is arguably the most important insight of differential privacy and there isn't really a competing approach.
- We quantify over *all* datasets (belonging to the domain), rather than assuming a distribution. One reason we do that is so privacy does not rest on unverifiable assumptions about the attacker's beliefs. The other reason is that it allows us to prove strong composiiton properties for differential privacy. People have studied many relaxations of this property.
- The dataset is a collection of n examples, and neighboring datasets differ in one example. This is not fundamental—what we are trying to capture is that the algorithm is insensitive to changing "one individual's data." But one individual's data can look very different in different contexts. People have studied differential privacy where the data is a graph and each node is an individual, where the data is a time series and each individaul contributes a segment, where data is a collection of training examples but each individaul contributes a large number of examples, and so on. Each case requires adjusting the notion of dataset and neighborin datasets but the core idea is the same.

- Differential privacy is *quantitative*—an algorithm is not simple private or not private, it has a parameter ε that captures how much it leaks. This is another fundamental insight of differential privacy, both because it allows us to reason about algorithms that are not perfectly private, which are all useful algorithms, and because it allows us to reason about the composition of many differentially private algorithms.
- Lastly, differential privacy uses a particular notion of *closeness of distributions*, applied to $\mathcal{A}(S)$ and $\mathcal{A}(S')$. We have introduced the most basic notion of pointwise multiplicative closeness, which has nice properties. People have considered many other more relaxed notions of closeness (in fact, the original definition is almost never used anymore), but some care is required because not all notions are equally good or even reasonable.

15.1 A First Example: The Randomized Response Mechanism

Let's start with a very simple algorithm called *randomized response*. Suppose each individual has a bit $X_i \in \{0,1\}$, indicating, for example, whether they have some medical condition. We would like to estimate the fraction of individuals with the condition, but we do not want to reveal any individual's answer. The idea is to ask each individual to give a noisy answer: with probability $2/3$ they report X_i, and with probability $1/3$ they report $1 - X_i$. The random choices are independent across individuals, and we only release the noisy answers $Y_1, \ldots, Y_n$.

Why might this be private? If we see $Y_i = 1$, that answer is twice as likely when $X_i = 1$ as when $X_i = 0$. The same is true with the roles of 0 and 1 reversed. So either answer gives some evidence about X_i, but it could have come from either value. We will show that even releasing all n answers satisfies differential privacy with $\varepsilon = \log 2$.

Of course, we would like to choose the privacy parameter rather than fixing it to $\log 2$. We can do that by adjusting the probability of giving the true answer. We can also allow each individual to have an arbitrary record $X_i \in \mathcal{X}$ and ask a fixed yes-or-no question about that record, represented by a predicate $\varphi : \mathcal{X} \to \{0,1\}$.

Algorithm 7: Randomized response

Input: A dataset $S = \{X_1, \ldots, X_n\} \in \mathcal{X}^n$, a fixed predicate $\varphi : \mathcal{X} \to \{0,1\}$, and a parameter $\varepsilon > 0$.
For each $i \in [n]$, independently sample

$$Y_i = \begin{cases} \varphi(X_i) & \text{with probability } \dfrac{e^\varepsilon}{e^\varepsilon + 1}, \\ 1 - \varphi(X_i) & \text{with probability } \dfrac{1}{e^\varepsilon + 1}. \end{cases}$$

Output: $\mathcal{A}(S) = (Y_1, \ldots, Y_n)$.

When $\varepsilon = \log 2$, this is exactly the $2/3$ versus $1/3$ rule above. As ε approaches 0, each answer approaches a fair random bit, so we get more privacy but less information about the data.

Privacy.

Theorem 15.2. *For every fixed predicate φ and every $\varepsilon > 0$, Algorithm 7 is ε-differentially private.*

Proof. Fix neighboring datasets S, S' differing at index i, and fix any possible output $y = (y_1, \ldots, y_n) \in \{0,1\}^n$. By independence, the probability of this output is a product of the probabilities of the individual answers. For every $j \neq i$, the input record is the same in both datasets, so those factors cancel. Thus,

$$\frac{\mathbb{P}[\mathcal{A}(S) = y]}{\mathbb{P}[\mathcal{A}(S') = y]} = \frac{\mathbb{P}[Y_i = y_i \mid S]}{\mathbb{P}[Y_i = y_i \mid S']} \leq \frac{e^\varepsilon/(e^\varepsilon + 1)}{1/(e^\varepsilon + 1)} = e^\varepsilon.$$

Here the probabilities are over the random choices of the algorithm, with the input dataset fixed. Finally, for every event $R \subseteq \{0,1\}^n$, summing this inequality over $y \in R$ gives

$$\mathbb{P}[\mathcal{A}(S) \in R] \leq e^\varepsilon \, \mathbb{P}[\mathcal{A}(S') \in R],$$

as required. □

Notice that the privacy parameter is ε, even though we release n answers. The reason is that changing one individual's record only changes the distribution of that individual's answer. The other $n - 1$ answers do not make it any easier to distinguish these two fixed neighboring datasets.

Accuracy. Now what can we learn from these noisy answers? Our goal is to estimate the empirical mean

$$\mu_S := \frac{1}{n} \sum_{i=1}^{n} \varphi(X_i).$$

Simply averaging the Y_i's gives a biased estimate. To see how to fix the bias, let $\bar{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i$ and compute

$$\mathbb{E}[Y_i] = \frac{e^\varepsilon}{e^\varepsilon + 1} \varphi(X_i) + \frac{1}{e^\varepsilon + 1}(1 - \varphi(X_i)) = \frac{1 + (e^\varepsilon - 1)\varphi(X_i)}{e^\varepsilon + 1}.$$

Solving for $\varphi(X_i)$ suggests the debiased estimator

$$\hat{\mu} = \frac{(e^\varepsilon + 1)\bar{Y} - 1}{e^\varepsilon - 1}.$$

For the original choice $\varepsilon = \log 2$, this is just $\hat{\mu} = 3\bar{Y} - 1$.

Theorem 15.3. *For every fixed dataset S, the estimator $\hat{\mu}$ is unbiased for μ_S and satisfies*

$$\sqrt{\mathbb{E}[(\hat{\mu} - \mu_S)^2]} = \frac{e^{\varepsilon/2}}{(e^\varepsilon - 1)\sqrt{n}}.$$

In particular, for $0 < \varepsilon \leq 1$, the root mean squared error is $\Theta(1/\varepsilon\sqrt{n})$.

Proof. The expression for $\mathbb{E}[Y_i]$ gives $\mathbb{E}[\hat{\mu}] = \mu_S$. Each Y_i is a Bernoulli random variable whose probability of being 1 is either $e^\varepsilon/(e^\varepsilon + 1)$ or $1/(e^\varepsilon + 1)$. Either way,

$$\mathrm{Var}(Y_i) = \frac{e^\varepsilon}{(e^\varepsilon + 1)^2}.$$

Since the answers are independent and the estimator is unbiased,

$$\mathbb{E}[(\hat{\mu} - \mu_S)^2] = \operatorname{Var}(\hat{\mu}) = \left(\frac{e^\varepsilon + 1}{e^\varepsilon - 1}\right)^2 \frac{1}{n^2} \sum_{i=1}^{n} \operatorname{Var}(Y_i) = \frac{e^\varepsilon}{n(e^\varepsilon - 1)^2}.$$

Taking square roots proves the formula. For $0 < \varepsilon \le 1$, we have $e^\varepsilon - 1 = \Theta(\varepsilon)$ and $e^{\varepsilon/2} = \Theta(1)$, which gives the stated dependence on ε and n. □

We can also get a guarantee that holds with high probability. Applying Hoeffding's inequality to $\bar{Y}$, for every $\beta \in (0, 1)$, with probability at least $1 - \beta$,

$$|\hat{\mu} - \mu_S| \le \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \sqrt{\frac{\log(2/\beta)}{2n}}.$$

Thus, for $0 < \varepsilon \le 1$, a sample size $n = O(\log(2/\beta)/\varepsilon^2\alpha^2)$ suffices for error at most α with probability at least $1 - \beta$. At a fixed confidence level, halving either the error tolerance α or the privacy parameter ε requires about four times as much data. These guarantees hold for every fixed dataset; we have not assumed that the records come from a distribution.

This example shows how we can learn an accurate aggregate statistic while protecting each individual's contribution. It also has a useful practical feature: each individual can generate their own noisy answer, so the curator never needs to see the original records. This stronger requirement is called *local differential privacy*. The Laplace mechanism will instead allow the curator to see the dataset and add noise to the final statistic, which can give much better accuracy.

15.2 Basic Properties of Differential Privacy

We would like to build algorithms out of smaller private pieces, and understand what happens when the same data is used more than once. Three basic properties make this possible.

Postprocessing. Once we have released a private output, further computations on that output do not cost any additional privacy.

Lemma 15.4 (Postprocessing). *If $\mathcal{A} \colon \mathcal{X}^n \to \mathcal{R}$ is ε-differentially private and $\mathcal{B} \colon \mathcal{R} \to \mathcal{Y}$ is any fixed, possibly randomized algorithm, then $\mathcal{B}(\mathcal{A}(S))$ is ε-differentially private. Here $\mathcal{B}$ only accesses the output of $\mathcal{A}$, and its random choices are independent of the data and the random choices of $\mathcal{A}$.*

Proof. If $\mathcal{B}$ is deterministic, apply the privacy guarantee for $\mathcal{A}$ to the event $\mathcal{B}^{-1}(R)$:

$$\mathbb{P}[\mathcal{B}(\mathcal{A}(S)) \in R] = \mathbb{P}[\mathcal{A}(S) \in \mathcal{B}^{-1}(R)] \le e^\varepsilon \, \mathbb{P}[\mathcal{A}(S') \in \mathcal{B}^{-1}(R)].$$

For randomized $\mathcal{B}$, fix its random choices, apply this inequality, and average over those choices. □

For example, the debiased estimate from randomized response is private because it is a function of the noisy answers. We can also clip the estimate to $[0, 1]$ without any additional privacy cost.

Composition. What if we go back to the original dataset to answer another question? Then the privacy parameters add, even if we choose the next question using the previous answers.

Lemma 15.5 (Basic composition). *Suppose an algorithm produces $Y_1, \dots, Y_k$ sequentially, using fresh randomness at each step, where*

$$Y_t = \mathcal{A}_t(S; Y_1, \dots, Y_{t-1}).$$

If, for every fixed history $y_1, \dots, y_{t-1}$, the algorithm $\mathcal{A}_t(\cdot\,; y_1, \dots, y_{t-1})$ is ε_t-differentially private, then releasing $(Y_1, \dots, Y_k)$ is $(\sum_{t=1}^{k} \varepsilon_t)$-differentially private.

Proof. For discrete outputs, fix a transcript $y = (y_1, \dots, y_k)$ and write its probability as a product of conditional probabilities. For neighboring S, S', each factor satisfies

$$\mathbb{P}[\mathcal{A}_t(S; y_1, \dots, y_{t-1}) = y_t] \le e^{\varepsilon_t}\, \mathbb{P}[\mathcal{A}_t(S'; y_1, \dots, y_{t-1}) = y_t].$$

Multiplying gives a factor $e^{\sum_t \varepsilon_t}$ for the transcript, and summing over transcripts gives the claim for every event. The same argument uses conditional distributions for general output spaces. □

This lets us think of ε as a *privacy budget*: to run k private computations, we can allocate parameters whose sum is at most ε. By postprocessing, releasing only the final answer also satisfies this guarantee.

Group privacy. The definition protects a change to one record. The guarantee weakens gracefully when we change several records.

Lemma 15.6 (Group privacy). *If $\mathcal{A}$ is ε-differentially private and S, S' differ in at most k records, then for every event R,*

$$\mathbb{P}[\mathcal{A}(S) \in R] \le e^{k\varepsilon}\, \mathbb{P}[\mathcal{A}(S') \in R].$$

Proof. Connect S to S' by a sequence $S = S^{(0)}, S^{(1)}, \dots, S^{(k)} = S'$ in which successive datasets are neighboring. Applying differential privacy at each step gives a factor of e^{ε} per step, hence $e^{k\varepsilon}$ overall. □

In particular, if $n\varepsilon \ll 1$, the output distributions are nearly identical even for datasets differing in every record. There is then little room for the algorithm to learn useful information about the data.

15.3 Differential Privacy and Membership Inference

Let's connect the definition back to membership inference. Fix a target record $Z = z$ and neighboring datasets S, S' that differ at index i, where S contains z and S' has a replacement record z'. A tester $\mathcal{T}$ sees the released output and the target, and tries to distinguish these two hypotheses. Write

$$\mathrm{TPR} = \mathbb{P}[\mathcal{T}(\mathcal{A}(S), z) = \mathsf{IN}], \qquad \mathrm{FPR} = \mathbb{P}[\mathcal{T}(\mathcal{A}(S'), z) = \mathsf{IN}].$$

The probabilities include the random choices of both the algorithm and the tester. We can bound the tester's performance directly from differential privacy.

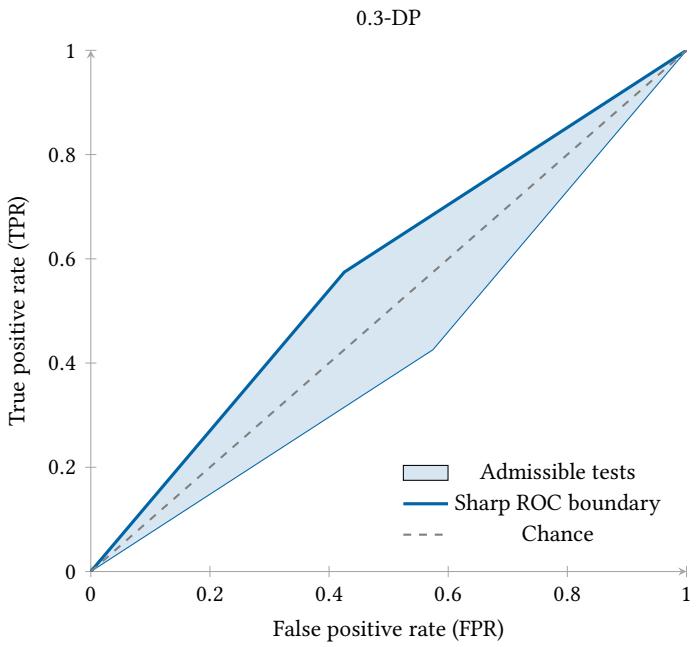


Figure 11: The hypothesis testing region for 0.3-DP. Every test of two neighboring datasets has its (FPR, TPR) pair in the shaded region. The blue upper boundary gives the largest allowable TPR at each FPR and is attainable by testing the output of binary randomized response, allowing randomized tests. The dashed diagonal is chance performance.

Theorem 15.7 (The hypothesis testing view of differential privacy). *If $\mathcal{A}$ is ε-differentially private, every tester and every neighboring pair satisfy*

$$\text{TPR} \le \min\{e^{\varepsilon}\text{FPR},\ 1 - e^{-\varepsilon}(1 - \text{FPR})\},$$
$$\text{TPR} \ge \max\{e^{-\varepsilon}\text{FPR},\ 1 - e^{\varepsilon}(1 - \text{FPR})\}.$$

Conversely, if these bounds hold for every tester and every neighboring pair, then $\mathcal{A}$ is ε-differentially private.

Proof. By postprocessing, the tester's output is private. Applying the definition to the output IN gives $\text{TPR} \le e^{\varepsilon}\text{FPR}$. Swapping S, S' gives $\text{FPR} \le e^{\varepsilon}\text{TPR}$. Applying the same two inequalities to the output OUT gives the remaining bounds. For the converse, take the tester that outputs IN exactly when $\mathcal{A}(S)$ belongs to an arbitrary event R. □

Thus, differential privacy constrains the entire ROC curve of every possible attack (Figure 11). For example, under 0.3-DP, an attack with FPR = 0.01 must have $\text{TPR} \le e^{0.3} \cdot 0.01 \approx 0.0135$. The guarantee holds even if the attacker knows all the other records.

These bounds also apply to the i.i.d. membership game we started with, where Z is either a uniformly chosen training record or an independent draw from $\mathcal{D}$. To see this, replace the selected

training record with a fresh independent draw while keeping the target Z fixed. The original and replacement datasets are neighboring, and the replacement dataset is independent of Z. Averaging the testing inequalities over these choices gives the same bounds on FPR and TPR in that game. This is why differential privacy prevents the membership inference attacks we saw earlier, regardless of the population distribution $\mathcal{D}$.

15.4 Interpreting Differential Privacy

What does this definition actually promise an individual? A tempting goal is that seeing the output should not change the attacker's beliefs about that individual. But this would rule out useful statistical studies! A study might reveal a connection between smoking and lung disease, changing what an insurer believes about a smoker's health even if that person never participated in the study.

Differential privacy instead compares what the attacker learns when an individual's data is used to what they would learn without that data. The study can reveal a population-level pattern, but a particular individual's contribution should make little difference. Postprocessing makes this meaningful for any decision the attacker makes using the released output.

We can make this comparison precise using Bayesian beliefs. For simplicity, take a discrete domain and output space. Let S be random under an arbitrary prior π on datasets, allowing any dependencies between the records. Fix an individual i and a default record $x_0 \in \mathcal{X}$, and let $S^{(-i)}$ replace X_i with x_0. For an output y, compare the two posterior distributions on the *original* dataset:

$$\pi(s \mid y) := \mathbb{P}[S = s \mid \mathcal{A}(S) = y], \qquad \pi_{-i}(s \mid y) := \mathbb{P}[S = s \mid \mathcal{A}(S^{(-i)}) = y].$$

The second posterior describes what the attacker would believe if individual i's record had not been used.

Theorem 15.8 (A Bayesian interpretation)**.** *If $\mathcal{A}$ is ε-differentially private, then for every prior π, every individual i, every output y with positive probability, and every event $E \subseteq \mathcal{X}^n$,*

$$e^{-2\varepsilon}\pi_{-i}(E \mid y) \le \pi(E \mid y) \le e^{2\varepsilon}\pi_{-i}(E \mid y).$$

Proof. Write $L(s) = \mathbb{P}[\mathcal{A}(s) = y]$ and $L_{-i}(s) = \mathbb{P}[\mathcal{A}(s^{(-i)}) = y]$. Differential privacy gives $e^{-\varepsilon}L_{-i}(s) \le L(s) \le e^{\varepsilon}L_{-i}(s)$ for every s. Averaging over the prior gives the same bounds for the normalizing constants in Bayes' rule. Thus the normalized posterior probabilities differ by at most $e^{2\varepsilon}$ in either direction; summing over $s \in E$ proves the claim. □

Notice that neither posterior has to be close to the prior. If records are correlated, an attacker can learn about X_i from other people's data, even when X_i is never used. Differential privacy controls the additional information from using that individual's record, rather than promising to hide everything that could be inferred about them.

References

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. "Calibrating Noise to Sensitivity in Private Data Analysis." *Conference on Theory of Cryptography*. TCC 2006. (Link).

[Smi23a] Adam Smith. *Lecture 4: Differential Privacy I*. Privacy in Statistics and Machine Learning, Spring 2023. 2023. (Link).

[Smi23b] Adam Smith. *Lecture 5: Differential Privacy II*. Privacy in Statistics and Machine Learning, Spring 2023. Based on materials developed with Jonathan Ullman. 2023. (Link).