CS839: Differential Privacy and Learning February 24, 2026

Lecture 11: Hypothesis Testing Interpretation of DP

Instructor: Gavin Brown *Scribe:* Zhuxiao Tang

Disclaimer: This document is intended as an informal supplement to in-class note-taking. It has not been given the level of scrutiny expected in polished lecture notes, let alone that reserved for peer-reviewed publications.

1 Motivation

Differential privacy formalizes "privacy" as: how well can someone guess if you were in the data? This can be interpreted as a *Hypothesis Testing Problem*: Suppose there is a randomized algorithm $A : \mathcal{X}^n \to \mathcal{Y}$. Fix $x, x \in \mathcal{X}^n$ such that $x \sim x'$. Run the algorithm on the input either x or x', an adversary sees the output Y. He has to test among the following two hypothesis:

- $H_0 : Y \sim A(x)$ (negative hypothesis).
- $H_1 : Y \sim A(x')$ (positive hypothesis).

This hypothesis testing problem, in a sense, completely describes the privacy of the algorithm A. If there are adjacent datasets where the hypotheses are easy to distinguish, then the adversary can easily guess if you were in the data or not, which means the algorithm has low privacy.

This perspective is called the "hypothesis testing interpretation of differential privacy." Since hypothesis testing is so well-studied, it is an appealing and natural perspective. It is also an extremely powerful theoretical tool: it underlies the optimal composition theorem of Kairouz et al. [2015] and the definitions of "f-DP" and "Gaussian DP" [Dong et al., 2022]. While we won't discuss these results in detail, these notes should get you in the right frame of mind to understand them.

2 ROC Curve

The difficulty of the above hypothesis testing problem can be expressed using the *receiver operating characteristic curve*, or ROC curve.[1] The adversary sees output Y and is trying to determine if it came from $Y \sim A(x)$ (the "negative" hypothesis) or $Y \sim A(x')$ (the "positive" hypothesis). The adversary observes y and outputs 0 or 1 following some strategy Adv, possibly a randomized starategy. Using Adv, the true positive rate (TPR) and the false positive rate (FPR) are defined as follows:

$$\text{TPR} = \Pr[Adv(Y) = 1 \mid Y \sim A(x')];$$
$$\text{FPR} = \Pr[Adv(Y) = 1 \mid Y \sim A(x)].$$

For this fixed adversary strategy Adv, we can plot the point (FPR, TPR) on the 2-D plane. For example, the point $(0, 0)$ corresponds to the strategy Adv_0: always output 0, and the point $(1, 1)$ corresponds to the strategy Adv_1: always output 1. Note that the point $(0, 1)$ corresponds to perfect

[1] Standard presentations of this topic in DP present things in an equivalent but slightly different language: essentially they present the ROC curve on different axes. Since ROC curves are more familiar to a CS/ML audience, we discuss them this way.

guessing (which may or may not be achieveable). For a given mechanism A, datasets x and x', the ROC curve is the "upper closure" of all possible Adv strategies.

Claim 2.1 (Mixing Adversaries)**.** *The ROC curve is concave.*

Proof. Here's a definition of concavity: a function f is concave if for all x, y and $\alpha \in [0, 1]$ we have $f(\alpha x + (1-\alpha)y) \geq \alpha f(x) + (1-\alpha)f(y)$.

For any two adversaries Adv_a and Adv_b and any coefficient α, we can obtain a new adversary by running Adv_a with probability α and otherwise running Adv_b. The two FPR and TPRs combine linearly, so best possible TPR at that (mixed) FPR can only be higher than this. □

Intuitively, the "closer" the ROC curve is to the top-left corner $(0, 1)$, the easier it is to distinguish between datasets, and the lower the privacy of algorithm A becomes (see some examples in Figure 1).

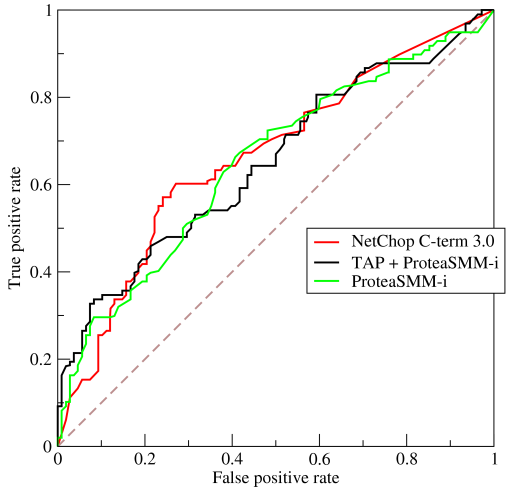


Figure 1: ROC curve. These curves are not concave; perhaps the creators don't know about Claim 2.1. Source: Wikipedia.

3 ROC Curves of Four DP Algorithms

In this section, we are going to analyze the ROC curves of four DP algorithms: Randomized Response, Leaky Input, Leaky Randomized Response, Laplace Mechanism. Each of these will come with ε and/or δ privacy parameters; we'll draw the curves with these values fixed.

3.1 Randomized Response

As defined in Table 1.

In addition to the "always output 0" and "always output 1" adversaries, the obvious adversary strategy is Adv_Y: Output Y. Note that this strategy is quite accurate if ε is large. We can calculate the TPR_Y and FPR_Y for Adv_Y:

$$\text{TPR}_Y = \Pr[Y = 1 \mid x = 1] = \frac{e^\varepsilon}{1 + e^\varepsilon};$$
$$\text{FPR}_Y = \Pr[Y = 1 \mid x = 0] = \frac{1}{1 + e^\varepsilon}.$$

Input \ Output	0	1
0	$\frac{e^\epsilon}{1+e^\epsilon}$	$\frac{1}{1+e^\epsilon}$
1	$\frac{1}{1+e^\epsilon}$	$\frac{e^\epsilon}{1+e^\epsilon}$

Table 1: RR_ε

By mixing these adversaries, we get an ROC curve with two linear segments, as in Fig. 2. Unsurprisingly, there are no better adversaries; it's impossible to sit above this curve for Randomized Response with ε-DP. We will discuss this more in Section 5.

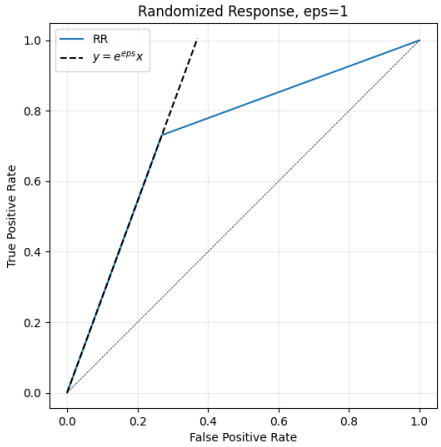


Figure 2: The ROC curve for ε-DP Randomized Response. I also plotted the line $y = e^\varepsilon x$ for comparison.

3.2 Leaky Input

In Table 2 we have a mechanism which satisfies $(0, \delta)$-DP. On any input, with probability $1 - \delta$, it outputs the "bot" symbol and leaks no information. Otherwise it leaks its input bit entirely. We've written this like "I am 0" to emphasize that this is a special output which means the algorithm isn't lying.

What do the adversaries look like? When you see "I am b" you should guess b. Otherwise, you can always guess 0 or always guess 1; this leads to two basic adversaries. And then we can mix them to get the full curve. We've plotted the ROC curve in Fig. 3, with an unreasonably large δ so that it's easier to see the line.

3.3 Leaky Randomized Response

Table 3 defines the Leaky Randomized Response, which is (ε, δ)-DP and should be interpreted as follows: with probability $1 - \delta$ run standard Randomized Response; otherwise leak your input in

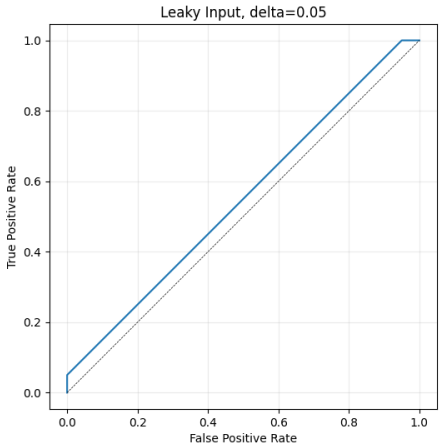


Figure 3: The ROC curve for $(0, \delta)$-DP "Leaky Input" mechanism.

Input \ Output	$\perp$	"I am 0"	"I am 1"
0	$1-\delta$	δ	0
1	$1-\delta$	0	δ

Table 2: Leaky Input

the clear. The ROC curve is in Fig. 4, again with extremely large $\delta = 0.05$ for illustration. The adversaries here combine the previous adversaries.

Input \ Output	0	1	"I am 0"	"I am 1"
0	$\frac{e^\varepsilon}{1+e^\varepsilon}(1-\delta)$	$\frac{1}{1+e^\varepsilon}(1-\delta)$	δ	0
1	$\frac{1}{1+e^\varepsilon}(1-\delta)$	$\frac{e^\varepsilon}{1+e^\varepsilon}(1-\delta)$	0	δ

Table 3: Leaky RR

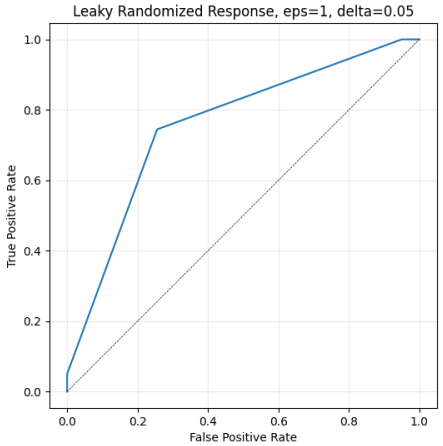


Figure 4: The ROC curve for (ε, δ)-DP "Leaky Randomized Response."

3.4 Laplace Mechanism

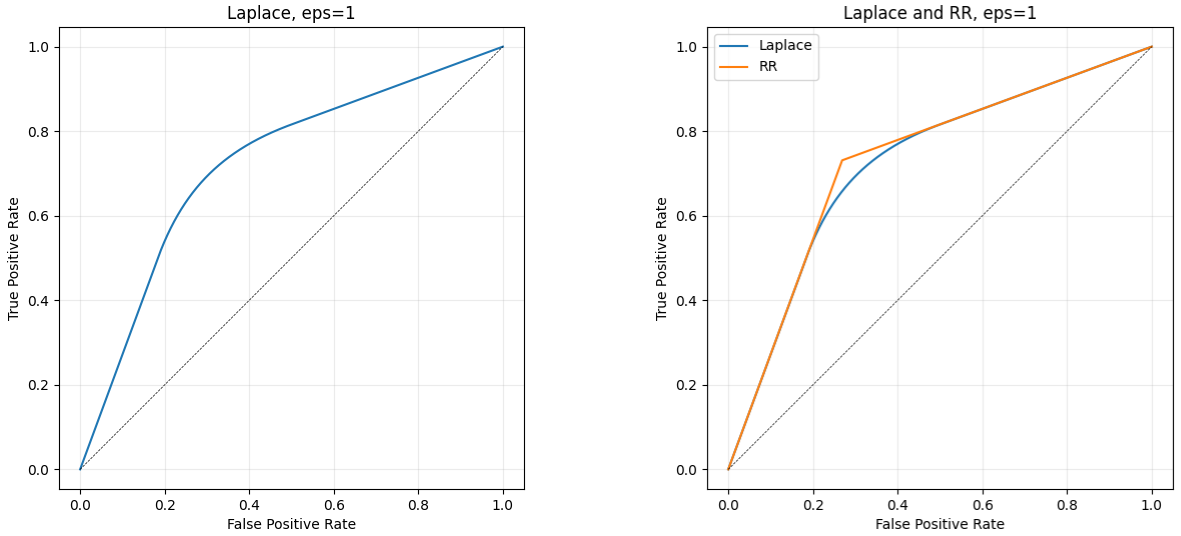
Here is the hypothesis-testing setup for the Laplace mechanism (assuming global sensitivity 1).

- $H_0 = 0 + \text{Lap}(1/\varepsilon) = \text{Lap}(0, 1/\varepsilon)$.
- $H_1 = 1 + \text{Lap}(1/\varepsilon) = \text{Lap}(1, 1/\varepsilon)$.

The ROC curve, Fig. 5a, is much harder to sketch. To understand it we need to refer to the Neyman–Pearson lemma, which we discuss in more detail in Section 5. It implies that optimal adversaries only need to look at *likelihood ratios*; in our setup, this is exactly equivalent to looking at the *privacy loss*! For this hypothesis testing problem, we have basically three cases:

- When the adversary sees $y \leq 0$, we abuse notation and write privacy loss $\log\left(\frac{\Pr[\text{Lap}(0,1/\varepsilon)=y]}{\Pr[\text{Lap}(1,1/\varepsilon)=y]}\right) = \varepsilon$. This is exactly what we see under Randomized Response when the output matches the input, so like the ROC curve for Randomized Response we have a straight line for part of this.
- When the adversary sees $y \geq 1$, the privacy loss is $-\varepsilon$, which accounts for the other part of our ROC curve which is straight.
- For outcomes $y \in (0, 1)$, the privacy loss lies in $(-\varepsilon, \varepsilon)$. This is a lower magnitude for privacy loss, which translates to a harder task for our adversary. Thus, as illustrated in Fig. 5b, the

ROC curve here lies within the curve for Randomized Response, meaning that there is a TPR/FPR regime which is feasible for Randomized Response but not Laplace noise.



(a) ROC curve for the Laplace mechanism, i.e. distinguishing $\mathrm{Lap}(0, 1/\varepsilon)$ from $\mathrm{Lap}(1, 1/\varepsilon)$

(b) ROCs for Laplace mechanism and Randomized Response, both with $\varepsilon = 1$.

Figure 5: ROC curves comparison.

4 Connecting the Figures to Definitions

Above, we saw that the ROC curve corresponding to the ε-DP Laplace mechanism lies underneath the ROC curve for ε-DP Randomized Response. This turns out to be a general phenomenon:

Theorem 4.1. *An algorithm A is (ε, δ)-DP if and only if for all adjacent datasets x and x' the ROC curve for distinguishing $A(x)$ from $A(x')$ lies on or below the ROC curve for (ε, δ)-Leaky RR (Table 3).*

Since Leaky RR with $\delta = 0$ is simply ε-Randomized Response, this theorem also characterizes pure DP.

The key application (and origin) of this claim is Kairouz et al. [2015], who use it to analyze composition. Essentially, to analyze the composition of arbitrary (ε, δ)-DP mechanisms, it suffices to analyze the composition of Leaky RR, which is much simpler. (For exactly this reason, your analysis of composition for Randomized Response on Homework 1 is quite a general result!) See Smith and Ullman [2025] for lecture notes on this.

Building on the hypothesis testing interpretation we have developed in these notes, Dong et al. [2022] introduced a notion called "f-DP," which is basically: any valid ROC curve $f : [0, 1] \to [0, 1]$ restricts the possible success of adversaries and thus gives us a privacy notion. We say that an algorithm satisfies f-DP if for all adjacent datasets the corresponding ROC curve lies below f. This is really quite a shift: the other definitions we'll see and use ((ε, δ)-DP, ρ-zCDP, (α, ε)-Rényi DP) all summarize the privacy of an algorithm with one or two numbers. In contrast, f-DP allows one to describe the privacy guarantees with a potentially very complex function.

The dominant instantiation of f-DP, however, still only uses one parameter. Dong et al. [2022] define *μ-Gaussian differential privacy* relative to the ROC curve that comes from distinguishing $\mathcal{N}(0,1)$ from $\mathcal{N}(\mu,1)$. See an example in Fig. 6. Like concentrated DP, this is a way we can formalize the concept of "algorithms with privacy loss that look Gaussian."

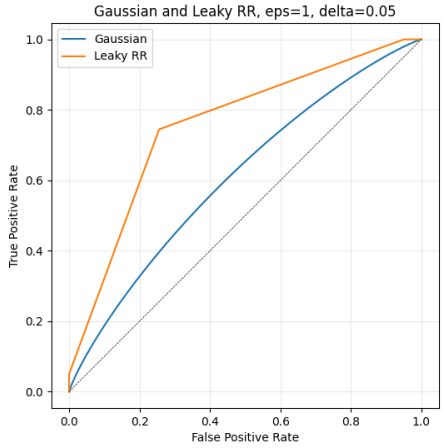


Figure 6: The ROC curves for the Gaussian mechanism and (ε, δ)-DP Leaky RR.

5 Optimality & the Neyman–Pearson Lemma

Above we construct some natural adversary strategies and used those to draw ROC curves. However, are these strategies optimal? Can there be other adversaries which do better? The answer is no. Fix an FPR $= \alpha$ (the probability that rejects H_0 when H_0 is true), we consider how large the TPR (the probability that accepts H_1 when H_1 is true) can be. The test (adversary strategy) that maximizes TPR with given FPR $= \alpha$ is called the uniformly most powerful (UMP) test (Wikipedia). The Neyman–Pearson lemma [Lehmann and Romano, 2005] says that: there exists a test ϕ and a constant k such that

1. $\mathbb{E}[\phi(Y) \mid x = 0] = \alpha$,
2. $$\phi(y) = \begin{cases} 1 & \text{when } \Pr[Y = y \mid x = 1] > k \Pr[Y = y \mid x = 0], \\ 0 & \text{when } \Pr[Y = y \mid x = 1] < k \Pr[Y = y \mid x = 0], \end{cases}$$

and ϕ (reject H_0 with probability $\phi(y)$) is the unique UMP-test at level α. Suppose $0 < \alpha < \frac{1}{1+e^\varepsilon}$, we may construct $k = e^\varepsilon$, $\phi(0) = 0$, $\phi(1) = (1 + e^\varepsilon)\alpha$. It is easy to check ϕ and k satisfies 1,2. Also,

$$\text{TPR} = \mathbb{E}[\phi(Y) \mid x = 1] = \phi(1) \cdot \frac{e^\varepsilon}{1 + e^\varepsilon} = e^\varepsilon \alpha.$$

This shows that the ROC curve of the UMP-test for $0 < \alpha < \frac{1}{1+e^\varepsilon}$ is the line connecting $(0,0)$ and $(\frac{1}{1+e^\varepsilon}, \frac{e^\varepsilon}{1+e^\varepsilon})$. For $\frac{1}{1+e^\varepsilon} < \alpha < 1$, it is similar.

Remark 5.1. The classical Neyman-Pearson lemma only applies to continuous random variables, and the UMP test has the form

$$\text{if } \frac{\Pr[Y = y \mid H_1]}{\Pr[Y = y \mid H_0]} > k \text{ then reject } H_0, \text{otherwise accept } H_0.$$

However, Theorem 3.2.1 in Lehmann and Romano [2005] allows the Neyman-Pearson lemma apply to discrete random variables, with $\phi(y)$ set to some real number at boundary case $\Pr[Y = y \mid H_1] = k \Pr[Y = y \mid H_0]$.

Here is another formal statement, also adapted from Lehmann and Romano [2005] but taken verbatim from Dong et al. [2022]. What we call an adversary above they call the test ϕ, and instead of thinking of it as being explicitly randomized we write it as outputting a number in $[0, 1]$ (which can be interpreted as a probability of guesing "positive").

Theorem 5.2 (Neyman–Pearson lemma)**.** *Let P and Q be probability distributions on Ω with densities p and q, respectively. For the hypothesis testing problem $H_0 : P$ versus $H_1 : Q$, a test $\phi : \Omega \to [0, 1]$ is the most powerful test at level α if and only if there are two constants $h \in [0, +\infty]$ and $c \in [0, 1]$ such that ϕ has the form*

$$\phi(\omega) = \begin{cases} 1, & \text{if } p(\omega) > hq(\omega) \\ c, & \text{if } p(\omega) = hq(\omega) \\ 0, & \text{if } p(\omega) < hq(\omega) \end{cases}$$

and $\mathbb{E}_P[\phi] = \alpha$.

References

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):3–37, 2022.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

Erich Leo Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer, 2005.

Adam Smith and Jonathan Ullman. Privacy in statistics and machine learning, 2025. URL `https://dpcourse.github.io/2025-spring/lecnotes-web/DP-S25-notes-lec-10-adv-composition.pdf`.