CS6290 Privacy-enhancing Technologies
Tutorial 9: Differential Privacy

NOTE: This tutorial's content is based on lecture notes from Prof. Gautam Kamath's CS 860 course at UWaterloo (linked at http://www.gautamkamath.com/courses/CS860-fa2022.html). A link to these notes are also provided at the end of this file for your convenience.

1. Differential Privacy Basis

ϵ-Differential Privacy: : A randomized algorithm $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ is said to be **ϵ-Differential Private** if for any two adjacent datasets D and D', and for all possible outcomes $S \subseteq \text{Range}(\mathcal{M})$, the following inequality holds:

$$\mathbb{P}(\mathcal{M}(D) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(D') \in S)$$

where $\varepsilon \geq 0$ is the *privacy parameter*. This definition must hold for *all* possible adjacent datasets and *all* possible sets of outcomes.

(See the "Differential Privacy" section of the Lecture 3 Note from Gautam Kamath's course for more details.)

2. Properties of Differential Privacy

2.1. Post-processing (for ϵ-DP)

Theorem: Post-processing Invariance for Differential Privacy Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ be an ε-differentially private mechanism. Let $f : \mathcal{R} \to \mathcal{R}'$ be any function (possibly randomized). Then, the mechanism $\mathcal{M}'(D) = f(\mathcal{M}(D))$ is also ε-differentially private.

2.2. Composition (Sequential Composition for ϵ-DP)

Theorem: Sequential Composition for Differential Privacy Let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ be a sequence of mechanisms such that $\mathcal{M}_i$ is ε_i-differentially private. Let $\mathcal{M}_{comb}(D) = (\mathcal{M}_1(D), \mathcal{M}_2(D), \ldots, \mathcal{M}_k(D))$ be the mechanism that releases the outputs of all $\mathcal{M}_i$'s. Then, $\mathcal{M}_{comb}$ is $\left(\sum_{i=1}^{k} \varepsilon_i\right)$-differentially private.

(See the "Properties of Dierential Privacy" section of the Lecture 4 Note from Gautam Kamath's course for more details.)

3. Approximate Differential Privacy

(ϵ, δ)-Differential Privacy: Definition: Definition: (ϵ, δ)-Differential Privacy A randomized algorithm $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ is said to be **(ϵ, δ)-Differential Private** if for any two adjacent datasets D and D', and for all possible outcomes $S \subseteq \text{Range}(\mathcal{M})$, the following inequality holds:

$$\mathbb{P}(\mathcal{M}(D) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(D') \in S) + \delta$$

where $\varepsilon \geq 0$ is the privacy parameter, and $\delta \in [0, 1)$ is the failure probability.

(See the "Approximate Dierential Privacy" section of the Lecture 5 Note from Gautam Kamath's course for more details.)

4. Gaussian Mechanism

Gaussian Mechanism for (ϵ, δ)-DP: Definition: Gaussian Mechanism Let $f : \mathcal{X}^n \to \mathbb{R}^k$ be a function with l_2-sensitivity $\Delta_2 f$. The Gaussian Mechanism $\mathcal{M}_G$ is defined as:

$$\mathcal{M}_G(D) = f(D) + Y,$$

where $Y = (Y_1, \ldots, Y_k)$ and each Y_i is independently drawn from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, where σ is chosen to ensure (ϵ, δ)-DP. In vector notation, $Y \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_k)$.
Privacy Guarantee of Gaussian Mechanism for (ϵ, δ)-DP: Theorem: Privacy of Gaussian Mechanism The Gaussian Mechanism $\mathcal{M}_G(D) = f(D) + Y$, with $Y \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_k)$ and $\sigma \geq \frac{\Delta_2 f \sqrt{2 \ln(1.25/\delta)}}{\varepsilon}$, is (ϵ, δ)-differentially private.

(See the "Gaussian Mechanism" section of the Lecture 5 Note from Gautam Kamath's course for more details.)

CS 860: Algorithms for Private Data Analysis Fall 2020

Lecture 3 — Intro to Differential Privacy

Prof. Gautam Kamath *Scribe: Gautam Kamath*

In this lecture, we will introduce differential privacy. We start with perhaps the first differentially private algorithm, by Warner from 1965 [War65].

Randomized Response

We work in a very simple setting. Suppose you are the instructor of a large class which has an important exam. You suspect that many students in the class cheated, but you aren't sure. How can you figure out what fraction of students cheated? Naturally, students would not be likely to honestly admit that they cheated.

Being a bit more precise: there are n people, and individual i has a sensitive bit $X_i \in \{0, 1\}$. They would like to ensure that no one else learns the value of X_i. Each person sends the analyst a message Y_i, which may depend on X_i and some random numbers which the individual can generate. Based on these Y_i's, the analyst would like to get an accurate estimate of $p = \frac{1}{n}\sum_{i=1}^{n} X_i$.

We can first start with the most obvious approach: individual i sends Y_i equal to the sensitive bit X_i. Foreshadowing, we write this in the following unconventional manner:

$$Y_i = \begin{cases} X_i & \text{with probability } 1 \\ 1 - X_i & \text{with probability } 0 \end{cases}$$

It is clear that the analyst can simply obtain $\tilde{p} = \frac{1}{n}\sum_{i=1}^{n} Y_i$, and that $\tilde{p} = p$ exactly. In other words, the result is perfectly accurate. However, the analyst sees Y_i, which is equal to X_i, and thus she learns the individual's private bit exactly: there is no privacy.

Consider an alternate strategy, as follows:

$$Y_i = \begin{cases} X_i & \text{with probability } 1/2 \\ 1 - X_i & \text{with probability } 1/2 \end{cases}$$

In this case, Y_i is perfectly private: in fact, it is a uniformly bit which does not depend on X_i at all, so the curator could not hope to infer anything about X_i from seeing it. But by the same token, this approach loses all sort of accuracy: $\tilde{Z} = \frac{1}{n}\sum_{i=1}^{n} Y_i$ is distributed as $\frac{1}{n}\text{Binomial}(n, 1/2)$, which is completely independent of the statistic Z.

At this point, we have two approaches: one which is perfectly accurate but not at all private, and one which is perfectly private but not at all accurate. The right approach will be to interpolate between these two extremes.

Consider the following strategy, which we will call *Randomized Response*, parameterized by some $\gamma \in [0, 1/2]$:

$$Y_i = \begin{cases} X_i & \text{with probability } 1/2 + \gamma \\ 1 - X_i & \text{with probability } 1/2 - \gamma \end{cases}$$

How private is this message Y_i, with respect to the true message X_i? We haven't built the tools to formally quantify this yet, so we'll be a bit informal for the time being. Note that $\gamma = 1/2$ corresponds to the first "honest" strategy, and $\gamma = 0$ is the second "uniformly random" strategy. What if we choose a γ in the middle, such as $\gamma = 1/4$? Then there will be a certain level of "plausible deniability" associated with the individual's disclosure: while $Y_i = X_i$ with probability $3/4$, it could be that their true bit was $1 - Y_1$, and this event happened with probability $1/4$. Informally speaking, how "deniable" their response is corresponds to the level of privacy they are afforded. In this way, they get a stronger privacy guarantee as γ approaches 0.

Let's put this aside for now, and focus on how accurate an estimate the analyst can obtain. Observe that

$$E[Y_i] = 2\gamma X_i + 1/2 - \gamma,$$

and thus

$$E\left[\frac{1}{2\gamma}\left(Y_i - 1/2 + \gamma\right)\right] = X_i.$$

This leads to the following natural estimator:

$$\tilde{p} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2\gamma}\left(Y_i - 1/2 + \gamma\right)\right].$$

The above calculation gives that $E[\tilde{p}] = p$. Next, we analyze the variance of $\tilde{p}$:

$$\mathbf{Var}[\tilde{p}] = \mathbf{Var}\left[\frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2\gamma}\left(Y_i - 1/2 + \gamma\right)\right]\right] = \frac{1}{4\gamma^2 n^2}\sum_{i=1}^{n}\mathbf{Var}\left[Y_i\right] \leq \frac{1}{16\gamma^2 n}.$$

The last inequality is due to the fact that the variance of a Bernoulli random variable is upper bounded by $1/4$. At this point, we can apply Chebyshev's inequality to obtain

$$|\tilde{p} - p| \leq O\left(\frac{1}{\gamma\sqrt{n}}\right).$$

This can also be obtained with high probability via a Chernoff bound.[1] As $n \to \infty$, this error goes to 0. An alternative way of wording this: if we wish to have additive error α, we require $n = O(1/\alpha^2\gamma^2)$ samples. Note that as γ gets closer to 0 (corresponding to stronger privacy), the error increases (or, with the second phrasing, the sample complexity). This is natural: the stronger the privacy guarantee we would like, the more data we require to achieve the same accuracy.

In order to proceed further in quantifying the level of privacy, we must (finally) introduce differential privacy. At its core, differential privacy is a broad formalization of this aforementioned notion of "plausible deniability."

Differential Privacy

In security and privacy, it is important to be precise about the precise setting in which we are working. We now define the setting for differential privacy, sometimes called *central differential*

[1] If you are not familiar with either the Chebyshev or Chernoff bound and the argument that we are applying here, it is important that you look it up and work out the details.

privacy or the *trusted curator* model. We imagine there are n individuals, X_1 through X_n, who each have their own datapoint. They send this point to a "trusted curator" – all individuals trust this curator with their raw datapoint, but no one else. Given their data, the curator runs an algorithm M, and publicly outputs the result of this computation. Differential privacy is a property of this algorithm M,[2] saying that no individual's data has a large impact on the output of the algorithm.

More formally, suppose we have an algorithm $M : \mathcal{X}^n \to \mathcal{Y}$. Consider any two datasets $X, X' \in \mathcal{X}^n$ which differ in exactly one entry. We call these *neighbouring datasets*, and sometimes denote this by $X \sim X'$. We say that M is *ε-(pure) differentially private* (ε-(pure) DP) if, for all neighbouring X, X', and all $T \subseteq \mathcal{Y}$, we have

$$\Pr[M(X) \in T] \leq e^{\varepsilon} \Pr[M(X') \in T],$$

where the randomness is over the choices made by M.

This definition was given by Dwork, McSherry, Nissim, and Smith in their seminal paper in 2006 [DMNS06]. It is now widely accepted as a strong and rigorous notion of data privacy. It has received acclaim in theory, winning the 2017 Gödel Prize, and the 2016 TCC Test-of-Time Award. At the same time, it has now seen adoption in practice at many organizations, including Apple [Dif17], Google [EPK14], Microsoft [DKY17], the US Census Bureau for the 2020 US Census [DLS+17], and much more.

Differential Privacy is an unusual sounding definition the first time you see it, so some discussion is in order.

- Differential privacy is quantitative in nature. A small ε corresponds to strong privacy, degrading as ε increases.
- ε should be thought of as a small-ish constant. Anything between (say) 0.1 and 5 might be a reasonable level privacy guarantee (smaller corresponds to stronger privacy), and you should be slightly skeptical of claims significantly outside this range.
- This is a worst-case guarantee, over all neighbouring datasets X and X'. Even if we expect our data to be randomly generated (and some realizations are incredibly unlikely), we still require privacy for all possible datasets nonetheless. While there do exist some notions of average-case privacy, these should be approached with caution – Steinke and Ullman write a series of posts which warn about the pitfalls of average-case notions of differential privacy [SU20a, SU20b].
- In words, the definition bounds the multiplicative increase (incurred by changing a single point in the dataset) in the probability of M's output satisfying any event.
- The use of a multiplicative e^{ε} in the probability might seem unnatural. For small ε, a Taylor expansion allows us to treat this as $\approx (1 + \varepsilon)$. The given definition is convenient because of the fact that $e^{\varepsilon_1} \cdot e^{\varepsilon_2} = e^{\varepsilon_1 + \varepsilon_2}$, which is useful when we examine the property of "group privacy" later.
- While the definition may look asymmetric, it is not: one can simply swap the role of X and X'.

[2] In differential privacy lingo, an algorithm is sometimes (confusingly) called a "mechanism."

- Convince yourself that any non-trivial (i.e., one that is not independent of the dataset) differentially private algorithm must be randomized.
- One might consider other notions of "closeness" of the distributions of $M(X)$ and $M(X')$. The given definition says the probability of any event is multiplicatively close. But at a glance, the statistical or total variation distance might also seem reasonable – essentially converting the multiplicative guarantee to an additive one. But this alternative notion would not give meaningful guarantees; we don't get into this here, but see Section 1.6 of [Vad17] for more discussion.
- Finally, we will generally use the notion "neighbouring datasets" where one point in X is changed arbitrarily to obtain X'. This is sometimes called "bounded" differential privacy, in contrast to "unbounded" differential privacy, where a point is either added or removed. In theory, these notions are equivalent up to a factor of 2, as an arbitrary change can be performed by removing one point and adding another. This can be formalized later, once we study the notion of group privacy. The former definition is usually more convenient mathematically.

That's it for technical comments on the definition.

As a brief interlude, let's discuss an alternative formulation of differential privacy in terms of hypothesis testing, due to Wasserman and Zhou [WZ10], and also explored by [KOV15, BBG+20].

This phrasing is slightly more "operational" in nature, viewing things from the perspective of an adversary. Specifically, suppose the adversary is trying to decide between the following two scenarios, where X and X' are neighbouring datasets, and one of the two is guaranteed to hold:

$$H_0 : \text{the underlying dataset is } X$$

$$H_1 : \text{the underlying dataset is } X'$$

Using statistics terminology, these are called the null and the alternate hypothesis, respectively. Based on the output of some algorithm M which is run on the dataset, the adversary is trying to determine whether H_0 or H_1 is true. Intuitively, differential privacy says that the adversary shouldn't be to get significant advantage over randomly guessing. The actual guarantee is slightly more refined – for example, they could simply guess H_0 every time, and they would always be right when H_0 is true (compared to probability $1/2$ by random guessing). Specifically, let p be the probability that the adversary predicts H_1 when H_0 is true (a "false positive") and q be the probability that the adversary predicts H_0 when H_1 is true (a "false negative"). ε-differential privacy implies that, simultaneously:

$$p + e^{\varepsilon} q \geq 1$$

$$e^{\varepsilon} p + q \geq 1$$

One can see that, when $\varepsilon = 0$, the adversary is essentially restricted to strategies that ignore the data and guess randomly (potentially in a biased way). As ε is increased, it allows the adversary some possibility of getting some advantage over blind guessing.

Why use this formulation of differential privacy? One reason is that it is more "operational" in nature, and gives one an alternative quantitative understanding of how well an adversary can detect the contribution of an individual. It is also used in understanding the privacy guarantees

we get when we run multiple private algorithms on the same dataset [KOV15]. The recent notion of Gaussian differential privacy [DRS19] also embraces this interpretation, rephrasing the privacy guarantee in terms of hypothesis testing between two Gaussian distributions.

Let's take a step back: what does differential privacy *mean*? Simply repeating the definition: differential privacy says that, the probability of any event is comparable in the cases when an individual does or does not include their data in the dataset. This has a number of implications of what differential privacy does and does not ensure.

First, it prevents many of the types of attacks we have seen before. The linkage-style attacks that we have observed are essentially ruled out – if such an attack were effective with your data in the dataset, it would be almost as effective without. This holds true for existing auxiliary datasets, as well as any *future* data releases as well. It also prevents reconstruction attacks, in some sense "matching" the bounds shown in the Dinur-Nissim attacks [DN03], as we will quantify in a later lecture. In fact, it protects against *arbitrary* risks, which can be reasoned about by simply revisiting the fact that any outcome is comparably likely whether or not the individual's data was actually included.

Differential privacy does *not* prevent you from making inferences about individuals. Stated alternatively: differential privacy does not prevent statistics and machine learning. Consider the classic "Smoking Causes Cancer" example [DR14]. Suppose an individual who smokes cigarettes is weighing their options in choosing to participate in a medical study, which examines whether smoking causes cancer. They know that a positive result to this study would be detrimental to them, as it would cause their insurance premiums to rise. They also know that the study is being performed using differentially privately, so they choose to participate, and they know their privacy will be respected. Unfortunately for them, the study reveals that smoking does cause cancer! This is a privacy violation, right? No: differential privacy ensures that the outcome of the study would not be significantly impacted by their participation. In other words, whether they participated or not, the result was going to come out anyway. For more discussion of the compatibility of privacy and learning, see [McS16].

Differential privacy is also not suitable for the case where the goal is to identify a specific individual, and this is antithetical to the definition. As a timely example, despite the clamoring for privacy-preserving solutions for tracking the spread of COVID-19, it is not immediately clear how one could use differential privacy to facilitate *individual-level* contact tracing. This would seem to require information about where a specific individual has been, and which particular individuals they have interacted with. On the other hand, it might be possible to facilitate aggregate-level tracking, say if many people who tested positive all attended the same event. In this vein, there is some interesting work done by Google on DP analysis of location traces, to see which types of locations people spend more and less time at since COVID-19 struck [ABC+20].

The definition of differential privacy is information theoretic in nature. That is, an adversary with unlimited amounts of computational power and auxiliary information is still unable to get an advantage. This is in contrast to cryptography, which typically focuses on computationally bounded adversaries. There has been some work on models of differential privacy where the adversary is computational bounded, see, e.g., [BNO08].

Randomized Response, Revisited

Design of differentially private algorithms is usually built around a few core primitives. One of these is randomized response, which we are now equipped to analyze the privacy of.

Now that we have the definition in hand, let's analyze the differential privacy guarantee when our algorithm M is randomized response. In fact, we will actually show that the bit-string $M(X_1, \ldots, X_n) = (Y_1, \ldots, Y_n)$ is differentially private – privacy of our estimate $\tilde{p}$ will follow by the post-processing property of differential privacy (essentially saying that a function of a differentially private object is also private), which we will discuss next lecture. We consider any particular realization $a \in \{0,1\}^n$ of $(Y_1, \ldots, Y_n)$. We have that $\Pr[M(X) = a] = \prod_{i=1}^n \Pr[Y_i = a_i]$. Suppose that X and X' differ only in coordinate j. Then we have that

$$\frac{\Pr[M(X) = a]}{\Pr[M(X') = a]} = \frac{\prod_{i=1}^n \Pr[Y_i = a_i]}{\prod_{i=1}^n \Pr[Y_i' = a_i]} = \frac{\Pr[Y_j = a_j]}{\Pr[Y_j' = a_j]} \leq \frac{1/2 + \gamma}{1/2 - \gamma} \leq e^{O(\gamma)},$$

where the last inequality holds for γ (say) smaller than $1/4$. Therefore, we have that ε-randomized response is $O(\varepsilon)$-differentially private, and achieves accuracy $O\left(\frac{1}{\varepsilon\sqrt{n}}\right)$. Actually, randomized response provides a stronger privacy guarantee than (central) differential privacy, it provides *local* differential privacy, in which individuals trust no one but themselves. This will be the topic of later lectures.

We'll end here, but next time we will start with the Laplace Mechanism. This is a very flexible algorithm which applies in more general settings, also achieves ε-differential privacy, and much better accuracy for this task than randomized response.

References

[ABC+20] Ahmet Aktay, Shailesh Bavadekar, Gwen Cossoul, John Davis, Damien Desfontaines, Alex Fabrikant, Evgeniy Gabrilovich, Krishna Gadepalli, Bryant Gipson, Miguel Guevara, Chaitanya Kamath, Mansi Kansal, Ali Lange, Chinmoy Mandayam, Andrew Oplinger, Christopher Pluntke, Thomas Roessler, Arran Schlosberg, Tomer Shekel, Swapnil Vispute, Mia Vu, Gregory Wellenius, Brian Williams, and Royce J. Wilson. Google covid-19 community mobility reports: Anonymization process description (version 1.0). *arXiv preprint arXiv:2004.04145*, 2020.

[BBG+20] Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and rényi differential privacy. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, AISTATS '20, pages 2496–2506. JMLR, Inc., 2020.

[BNO08] Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In *Proceedings of the 28th Annual International Cryptology Conference*, CRYPTO '08, pages 451–468, Berlin, Heidelberg, 2008. Springer.

[Dif17] Differential Privacy Team, Apple. Learning with privacy at scale. https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appledifferentialprivacysystem.pdf, December 2017.

[DKY17] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems 30*, NIPS '17, pages 3571–3580. Curran Associates, Inc., 2017.

[DLS+17] Aref N. Dajani, Amy D. Lauger, Phyllis E. Singer, Daniel Kifer, Jerome P. Reiter, Ashwin Machanavajjhala, Simson L. Garfinkel, Scot A. Dahl, Matthew Graham, Vishesh Karwa, Hang Kim, Philip Lelerc, Ian M. Schmutte, William N. Sexton, Lars Vilhuber, and John M. Abowd. The modernization of statistical disclosure limitation at the U.S. census bureau, 2017. Presented at the September 2017 meeting of the Census Scientific Advisory Committee.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006. Springer.

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, pages 202–210, New York, NY, USA, 2003. ACM.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[DRS19] Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019.

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM Conference on Computer and Communications Security*, CCS '14, pages 1054–1067, New York, NY, USA, 2014. ACM.

[KOV15] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML '15, pages 1376–1385. JMLR, Inc., 2015.

[McS16] Frank McSherry. Statistical inference considered harmful. https://github.com/frankmcsherry/blog/blob/master/posts/2016-06-14.md, June 2016.

[SU20a] Thomas Steinke and Jonathan Ullman. The pitfalls of average-case differential privacy. https://differentialprivacy.org/average-case-dp/, July 2020.

[SU20b] Thomas Steinke and Jonathan Ullman. Why privacy needs composition. https://differentialprivacy.org/privacy-composition/, August 2020.

[Vad17] Salil Vadhan. The complexity of differential privacy. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, chapter 7, pages 347–450. Springer International Publishing AG, Cham, Switzerland, 2017.

[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.

[WZ10] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

CS 860: Algorithms for Private Data Analysis Fall 2020

Lecture 4 — Intro to Differential Privacy, Part 2

Prof. Gautam Kamath *Scribe: Gautam Kamath*

Today, we continue with some of the core fundamentals of differential privacy. We start by presenting arguably the most important algorithm in differential privacy: the Laplace mechanism.

Laplace Mechanism

Content in this section is based heavily off of Section 3.3 of [DR14].

Last time, we saw our first differentially private algorithm: randomized response. At its core, this is useful for privatizing the value of a single bit: whether an individual's private data is 0 or 1 (though it can be generalized to categorical data). While the privatized result can be used for whatever other query we wish to answer, this is indirect and often lossy. Our first focus today, the *Laplace mechanism*, will directly address any sort of numeric query. Before we introduce the algorithm itself, we will require the important concept of the *sensitivity* of a function (in particular, the ℓ_1 sensitivity).

Definition 1. *Let $f : \mathcal{X}^n \to \mathbb{R}^k$. The ℓ_1-sensitivity of f is*

$$\Delta^{(f)} = \max_{X,X'} \|f(X) - f(X')\|_1,$$

where X and X' are neighbouring databases.

When the function we are discussing is clear from context, we will drop f and just use Δ for the ℓ_1-sensitivity.

The sensitivity is a rather natural quantity to consider in the context of differential privacy. Indeed, recall that differential privacy attemps to mask the contributions of any one individual. Upper bounding "how much" the function can change by modifying a single datum is thus well motivated intuitively, and we will see how we exploit it technically. I put "how much" in quotes, since it may seem mysterious why we consider the ℓ_1-sensitivity of the function, and not the ℓ_2-sensitivity or some other notion. The answer is that we use it for technical reasons, though ℓ_2-sensitivity is the right notion in other settings (say, for the *Gaussian mechanism*, rather than the Laplace mechanism). Note that these are identical in the univariate setting (i.e., when $k = 1$), but may vary in the multivariate setting (up to a factor of $\sqrt{k}$).

As a simple running example, we will consider the function $f = \frac{1}{n}\sum_{i=1}^n X_i$, where $X_i \in \{0, 1\}$. It is not hard to verify that the sensitivity of this function is $1/n$, realized when any bit is flipped.

As the name of the mechanism suggests, the *Laplace distribution* will be a key component of the Laplace mechanism.

Definition 2. *The* Laplace distribution *with location and scale parameters* 0 *and b, respectively, has the following density:*

$$p(x) = \frac{1}{2b}\exp\left(-\frac{|x|}{b}\right).$$

Note that the variance of this distribution is $2b^2$. Some visualizations of the density of the Laplace distribution are provided in Figure 1. It can be seen as a symmetrization of the exponential distribution, which is only supported on $x \in [0, \infty)$ and has density $\propto \exp(-cx)$, versus the Laplace distribution which is supported on $x \in \mathbb{R}$ and has density $\propto \exp(-c|x|)$. As another potentially familiar point of comparison, the Gaussian distribution is also supported on $\mathbb{R}$, and has density $\propto \exp(-cx^2)$. We can see the Gaussian distribution has lighter tails than the Laplace distribution, meaning that it enjoys somewhat stronger concentration (though both tails decay at least exponentially).

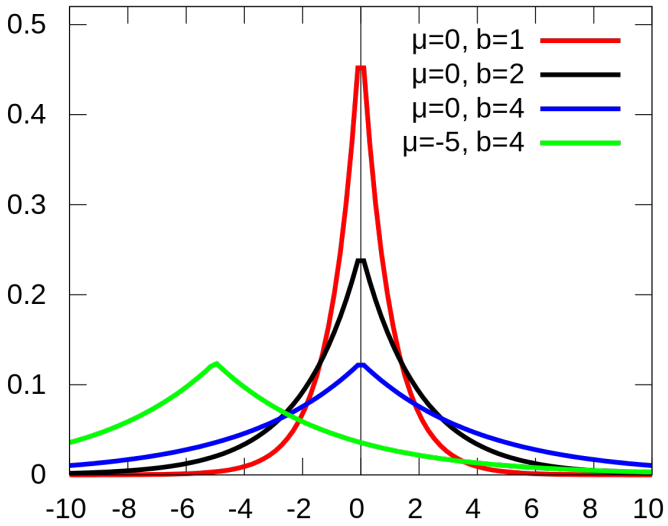

Figure 1: Figure from Wikipedia. Laplace distributions with various parameters.

With the Laplace distribution in hand, we are ready to introduce the Laplace mechanism. It is very simple to state: add noise to the statistic of magnitude proportional to its sensitivity.

Definition 3. *Let* $f : \mathcal{X}^n \to \mathbb{R}^k$. *The* Laplace mechanism *is defined as*

$$M(X) = f(X) + (Y_1, \ldots, Y_k),$$

where the Y_i *are independent* Laplace(Δ/ε) *random variables.*

Let us apply this to our running example of $f = \frac{1}{n}\sum_{i=1}^n X_i$. This is a simple application of Definition 3, where $k = 1$. As we previously established, $\Delta = 1/n$. Therefore, the Laplace mechanism run on a dataset is $\tilde{p} = f(X) + Y$, where Y is Laplace$(1/\varepsilon n)$. Recalling that we previously defined $p = f(X)$, we have that $\mathbf{E}[\tilde{p}] = p$, by linearity of expectations and since $\mathbf{E}[Y] = 0$. Computing the variance, we have $\mathbf{Var}[\tilde{p}] = \mathbf{Var}[Y] = O(1/\varepsilon^2 n^2)$, and using Chebyshev's inequality gives that $|\tilde{p} - p| \leq O(1/\varepsilon n)$ with reasonable probability.[1] We can compare this with the accuracy of ε-randomized response, which was $O(1/\varepsilon\sqrt{n})$ – the Laplace mechanism's error is quadratically smaller in n.

It remains to show that the Laplace mechanism is differentially private.

[1]Note that one can easily get a high probability bound by examining the tails of the distribution – the error will exceed $O(\log(1/\beta)/\varepsilon n)$ with probability $\leq \beta$.

Theorem 4. *The Laplace mechanism is ε-differentially private.*

Proof. Let X and Y be any neighbouring databases, differing in any one entry. We let $p_X(z)$ and $p_Y(z)$ be the probability density functions of $M(X)$ and $M(Y)$ evaluated at a point $z \in \mathbb{R}^k$. To prove differential privacy, we will show that that their ratio is bounded above by $\exp(\varepsilon)$, for an arbitrary choice of z and neighboring X and Y.

$$\begin{aligned}
\frac{p_X(z)}{p_Y(z)} &= \frac{\prod_{i=1}^k \exp\left(-\frac{\varepsilon|f(X)_i - z_i|}{\Delta}\right)}{\prod_{i=1}^k \exp\left(-\frac{\varepsilon|f(Y)_i - z_i|}{\Delta}\right)} \\
&= \prod_{i=1}^k \exp\left(-\frac{\varepsilon(|f(X)_i - z_i| - |f(Y)_i - z_i|)}{\Delta}\right) \\
&\leq \prod_{i=1}^k \exp\left(-\frac{\varepsilon|f(Y)_i - f(X)_i|}{\Delta}\right) \\
&= \exp\left(\frac{\varepsilon \sum_{i=1}^k |f(X)_i - f(Y)_i|}{\Delta}\right) \\
&= \exp\left(\frac{\varepsilon\|f(X) - f(Y)\|_1}{\Delta}\right) \\
&\leq \exp(\varepsilon).
\end{aligned}$$

The first inequality is the triangle inequality, and the last uses the definition of ℓ_1-sensitivity. □

Counting Queries

We'll apply this in a few different scenarios. First, let's look at *counting queries*. This is essentially the non-normalized version of our running example we have used so far (though the term counting query is sometimes used interchangeably for both). Specifically, we can ask the question "How many people in the dataset have property P?" If we just ask one question like this, the analysis follows very similarly to before. Each individual will have a bit $X_i \in \{0, 1\}$ indicating whether or not this is true about them, and the function f we consider is their sum. The sensitivity is 1, and thus an ε-differential privatization of this statistic would be $f(X) + \text{Laplace}(1/\varepsilon)$. This introduces error to this query on the order of $O(1/\varepsilon)$, independent of the size of the database.

What if we wanted to answer many queries? The way we defined the Laplace mechanism this makes this easy to reason about. Suppose we had k counting queries $f = (f_1, \ldots, f_k)$, which are all specified in advance. We would simply output the vector $f(X) + Y$, where the Y_i's are i.i.d. Laplace random variables. But what scale parameter should we use for the Y_i's? Each individual counting query f_j has sensitivity 1, but we are using the same dataset to answer all queries, so changing a single individual may affect the result of many queries at once. Consider, for example, the swapping of two individuals: one who satisfies no properties, and one who satisfies every property. This swap would change the result of every query by 1, and therefore the overall ℓ_1 sensitivity is k. Let's analyze this slightly more mathematically. Since $f(X) = \sum(f_1(X_i), \ldots, f_k(X_i))$, if neighbouring datasets X and Y differ in that one contains x and the other contains y, the ℓ_1 difference can be

written as $\sum_j |f_j(x) - f_j(y)|$, as the common terms cancel. This can be upper bounded as follows: $\sum_j |f_j(x) - f_j(y)| \leq \sum_j 1 = k$.

With this sensitivity bound $\Delta = k$ in hand, we can add $Y_i \sim \text{Laplace}(k/\varepsilon)$ noise to each coordinate, answering each counting query with error of magnitude $O(k/\varepsilon)$.

Some discussion is in order. First, this method of answering k counting queries required us to specify all the queries in advance – in other words, a *non-adaptive* setting. We will later see that similar guarantees are achieveable in the adaptive setting, where the choice of a query may depend on previous ones. Secondly, let's compare this with the Dinur-Nissim attacks [DN03] discussed in previous lectures. That showed that if the analyst asks $\Omega(n)$ counting queries, defended by the curator using noise of magnitude $O(\sqrt{n})$, the analyst can reconstruct the database and cause blatant non-privacy. On the other hand, the above strategy shows that, if the analyst asks $O(n)$ counting queries and the curator adds noise of magnitude $O(n/\varepsilon)$, then privacy is preserved. This seems to be a huge gap in the two results: are there stronger attacks, which allow the adversary to succeed even with more noise? Or can we add less noise and still preserve privacy? Fortunately, the latter is true, and it is possible to add less noise via better analysis (as well as a slight relaxation of the definition of differential privacy), using something called *advanced composition*.

Histograms

Another natural type of query is a *histogram query*. With counting queries, we had to be pessimistic – changing a single individual could affect the results of every query at once. But certain *structures* of queries might allow us to perform better sensitivity analysis. Suppose each individual in the dataset has some categorical feature: for example, let's say the person's age (rounded down to the nearest whole number). We would like to answer questions like "How many people in the dataset are X years old?" While this is similar to the counting queries example, an individual here can not have more than one age. Our function f will be $(f_0, f_1, \ldots, f_{k-1})$, where f_i asks how many people are i years old. It is not hard to argue that the ℓ_1-sensitivity of this function is 2: changing any individual's age would result in one count decrementing and another count incremening. More formally, similar to before we consider neighbouring datasets X and Y, where the difference is that one dataset has x and the other has it replaced by y. Then the ℓ_1 sensitivity is equal to $\|e_a - e_b\|_1 = 2$, where e_j is the j-th standard basis vector (having a 1 in the jth position and 0 elsewhere), and f_a and f_b are the functions which evaluate to 1 on x and y. As such, the Laplace mechanism prescribes outputting $f(X) + Y$, where $Y_i \sim \text{Laplace}(2/\varepsilon)$, where the magnitude is independent of the number of "bins" k.

How much error does this incur? As before, we observe that any individual count will have error on the order of $O(1/\varepsilon)$. However, we can also reason about the error incurred in all counts simultaneously! We can use the following basic fact about the Laplace distribution:

Fact 5. *If* $Y \sim \text{Laplace}(b)$*, then*

$$\Pr[|Y| \geq tb] = \exp(-t).$$

This can be verified simply by integrating the PDF of the Laplace distribution. Now, for the ith bin, the error in the count is exactly Y_i, and we have that $\Pr[|Y_i| \geq 2\log(k/\beta)/\varepsilon] \leq \beta/k$. Taking a union bound over all bins, it means that the probability that *any* bin has error $\geq 2\log(k/\beta)/\varepsilon$ is at most β. Stated differently: the magnitude of the error scales only logarithmically with the number of bins, in contrast to the linear relationship when our counting queries were arbitrary.

Properties of Differential Privacy

One of the reasons for the success of differential privacy is how "user friendly" it is. Specifically, it possesses a number of convenient properties that make it possible to think about differential privacy in a very modular fashion. We will discuss some of the most fundamental properties: closure under post-processing, group privacy, and basic composition.

Post-Processing

One convenient fact about differentially private algorithms is that once a quantity is privatized, it can't be "un-privatized," if the data is not used again. We used this already when we were analyzing randomized response.

Theorem 6. *Let $M : \mathcal{X}^n \to \mathcal{Y}$ be ε-differentially private, and let $F : \mathcal{Y} \to \mathcal{Z}$ be an arbitrary randomized mapping. Then $F \circ M$ is ε-differentially private.*

Proof. Since F is a randomized function, we can consider it to be a distribution over deterministic functions f. The privacy proof follows for every neighbouring dataset X, X' and $T \subseteq \mathcal{Y}$:

$$\begin{aligned}\Pr[F(M(X)) \in T] &= \mathbf{E}_{f\sim F}[\Pr[M(X) \in f^{-1}(T)]] \\ &\leq \mathbf{E}_{f\sim F}[e^{\varepsilon} \Pr[M(X') \in f^{-1}(T)]] \\ &= e^{\varepsilon} \Pr[F(M(X')) \in T].\end{aligned}$$

□

Group Privacy

So far, we've discussed differential privacy with respect to neighbouring datasets – ones which differ in exactly one entry. But one might wonder about datasets which differ in multiple entries. The definition of differential privacy allows for the guarantee to decay gracefully as the distance is increased.

Theorem 7. *Let $M : \mathcal{X}^n \to \mathcal{Y}$ be an ε-differentially private algorithm. Suppose X and X' are two datasets which differ in exactly k positions. Then for all $T \subseteq \mathcal{Y}$, we have*

$$\Pr[M(X) \in T] \leq \exp(k\varepsilon) \Pr[M(X') \in T].$$

Proof. The proof follows by what is known in the business as a "hybrid" argument. Let $X^{(0)} = X$, $X^{(k)} = X'$ – since they differ in k positions, there exists a sequence $X^{(0)}$ through $X^{(k)}$ such that each consecutive pair of datasets is neighbouring. Then, for all $T \subseteq \mathcal{Y}$:

$$\begin{aligned}\Pr[M(X^{(0)}) \in T] &\leq e^{\varepsilon} \Pr[M(X^{(1)}) \in T] \\ &\leq e^{2\varepsilon} \Pr[M(X^{(2)}) \in T] \\ &\cdots \\ &\leq e^{k\varepsilon} \Pr[M(X^{(k)}) \in T].\end{aligned}$$

□

(Basic) Composition

As a final but important property, we discuss *composition* of differentially private algorithms. Suppose you ran k differentially private algorithms on the same dataset, and released all of their results – how private is this as a whole? Essentially, the overall privacy guarantee decays by a factor of k. We already saw this when we considered the Laplace mechanism when the queries were chose in advance, but the following result holds for general differentially private algorithms, even when the queries are chosen adaptively!

Theorem 8. *Suppose $M = (M_1, \ldots, M_k)$ is a sequence of ε-differentially private algorithms, potentially chosen sequentially and adaptively. Then M is $k\varepsilon$-differentially private.*

Proof. Fix two neighbouring datasets X and X', and consider some sequence of outputs $y = (y_1, \ldots, y_k)$. Then we have

$$\begin{aligned}\frac{\Pr[M(X) = y]}{\Pr[M(X') = y]} &= \prod_{i=1}^{k} \frac{\Pr[M_i(X) = y_i | (M_1(X), \ldots, M_{i-1}(X)) = (y_1, \ldots, y_{i-1})]}{\Pr[M_i(X') = y_i | (M_1(X'), \ldots, M_{i-1}(X')) = (y_1, \ldots, y_{i-1})]} \\ &\leq \prod_{i=1}^{k} \exp(\varepsilon) \\ &= \exp(k\varepsilon).\end{aligned}$$

□

Surprisingly, it is possible to do better: we can get away with paying a factor of $O(\sqrt{k})$ in the privacy parameter, rather than the k given above, However, this will require a relaxation of the privacy notion, which we will leave for next lecture.

References

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, pages 202–210, New York, NY, USA, 2003. ACM.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

CS 860: Algorithms for Private Data Analysis Fall 2020

Lecture 5 — Approximate Differential Privacy

Prof. Gautam Kamath *Scribe: Gautam Kamath*

We will now study a relaxation of ε-differential privacy, first proposed by Dwork, Kenthapadi, McSherry, Mironov, and Naor [DKM+06]. This relaxation will possess marginally weaker privacy guarantees, but allow us to add significantly less noise to achieve it. This will be the topic of the next lecture, but today we will focus on introducing this relaxation and some of the basic algorithms and properties.

Approximate Differential Privacy

A few lectures ago, we mentioned that statistical distance is not an appropriate notion of distance for differential privacy. In particular, if M is an algorithm, and X, X' are neighbouring datasets, then

$$\Pr[M(X) \in T] \leq \Pr[M(X') \in T] + \varepsilon$$

provides meaningless accuracy for small ε and weak privacy for large ε, see Section 1.6 of [Vad17] for more details. However, when used in combination with (pure) ε-differential privacy, it gives rise to the notion of (approximate) (ε, δ)-differential privacy.

Definition 1 (Approximate Differential Privacy)**.** *An algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ is (ε, δ)-differentially private (i.e., it satisfies* approximate differential privacy*) if, for all neighbouring databases $X, X' \in \mathcal{X}^n$, and all $T \subseteq \mathcal{Y}$,*

$$\Pr[M(X) \in T] \leq e^{\varepsilon} \Pr[M(X') \in T] + \delta.$$

How should we interpret this new definition? One way is to consider the *privacy loss* random variable.

Definition 2. *Let Y and Z be two random variables. The* privacy loss random variable $\mathcal{L}_{Y||Z}$ *is distributed by drawing $t \sim Y$, and outputting $\ln\left(\frac{\Pr[Y=t]}{\Pr[Z=t]}\right)$. If the supports of Y and Z are not equal, then the privacy loss random variable is undefined.*

This definition holds for continuous random variables as well, by considering the ratio of their densities. Though we say the privacy loss random variable will be undefined if the supports are not equal, we will (informally) state that the privacy loss is infinite when sampling an outcome that realizes this. While we state this for general random variables, we will apply it for Y and Z equal to $M(X)$ and $M(X')$, where, as usual, M is an algorithm and X and X' are neighbouring datasets. Intuitively, the realization of the privacy loss random variable indicates how much more (or less) likely X was to be the input dataset compared to X', based on observing the realization of $M(X)$.

From the definition of pure differential privacy, it is immediate to see that ε-DP corresponds to $\left|\mathcal{L}_{M(X)||M(X')}\right|$ being bounded by ε for all neighbouring X, X'. Succinctly, ε-DP says that the absolute value of the privacy loss random variable is bounded by ε with probability 1. While not

immediate, it can be shown (i.e., Lemma 3.17 of [DR14]) that (ε, δ)-DP is equivalent to saying that the absolute value of the privacy loss random variable is bounded by ε with probability $1 - \delta$.

The mystery at this point is, what can happen when this bad probability-δ event happens? And consequently, how small should δ be set to avoid this bad event? To address the former question: there's a wide range of possible options.

First, consider a very simple (and rather useless) algorithm. With probability $1-\delta$, it does nothing, i.e., outputs $\perp$. On the other hand, with probability δ, it outputs the entire dataset! As we can see, in the former case (which happens with probability $1 - \delta$) the privacy loss random variable will be 0. In the other case (which, non-technically speaking, is not at all private) we have infinite privacy loss, but this happens only with probability δ. Thus, it seems like it is possible that terrible things could happen when this probability δ event occurs, and we should set δ to be quite small.

But how small is "quite small"? The following "name and shame"[1] example shows that $\delta > 1/n$ is not meaningful. Suppose an algorithm NS_δ iterates over its input, and independently for each datapoint, outputs the datum (which could be the individual's SSN, emails, etc.) with probability δ. We will shortly prove that this algorithm is $(0, \delta)$-DP. However, the probability that at least one person has their data output is $1 - (1 - \delta)^n$, which by a Taylor expansion is roughly δn for small enough δ. Thus, we can see that unless $\delta \ll 1/n$, there's a non-trivial chance that at least one individual's data is output in the clear. Most would not consider an algorithm which publishes a random individual's data to satisfy a strong privacy guarantee, and thus we will consider $\delta \ll 1/n$. For instance, if we were in a situation like this, something like $\delta = 1/n^{1.1}$ is perhaps the largest δ we would tolerate. To draw a parallel with other security settings, we sometimes imagine δ as "cryptographically small."

Let us prove that NS_δ is $(0, \delta)$-DP, following presentation of Smith [Smi20]. Consider any two neighbouring datasets X and X', which differ in only entry i. Let T be a set of datapoints. Let E be the event that entry i is output. Conditioning on $\bar{E}$ (i.e., that event E does not happen), then the output distribution of NS_δ is identical under X and X'. To see this, observe that X and X' are identical except for the ith entry.

The proof concludes as follows:

$$\begin{aligned}
\Pr[NS_\delta(X) \in T] &= \Pr[NS_\delta(X) \in T|\bar{E}] \Pr[\bar{E}] + \Pr[NS_\delta(X) \in T|E] \Pr[E] \\
&= \Pr[NS_\delta(X') \in T|\bar{E}] \Pr[\bar{E}] + \Pr[NS_\delta(X) \in T|E] \Pr[E] \\
&\leq \Pr[NS_\delta(X') \in T|\bar{E}] \Pr[\bar{E}] + 1 \cdot \delta \\
&\leq \Pr[NS_\delta(X') \in T] + \delta.
\end{aligned}$$

In the two examples we've seen so far, when the privacy loss random variable exceeds ε, it is a "catastrophic failure." In the first example, we output the entire dataset with probability δ. In the latter, we output the single datapoint which distinguishes X and X' with probability δ. In both these cases, the privacy loss random variable is either 0, or ∞ with probability δ. Thus, given no further information, one should pessimistically assume that terrible things happen with probability δ. However, for many common algorithms (such as the Gaussian mechanism which we will cover shortly), the privacy loss random variable may decay gracefully – even if this probability-δ event occurs, the privacy loss might not be significantly more than ε. This is sometimes parameterized by multiple guarantees for the same algorithm – for instance, to make up some numbers, we might be

[1] I believe this excellent name (and potentially even the example) is due to Adam Smith.

told that an algorithm satisfies both $(1, 0.001)$-DP as well as $(2, 0.0001)$-DP. There are cleaner ways of characterizing the privacy loss of an algorithm (compared to the relatively crude "threshold" provided by (ε, δ)-DP), including Rényi DP [Mir17], concentrated DP [DR16, BS16]. We will likely discuss some of these later in the class, and I might write a blog post on this topic if I get any time this term (questionable at this point).

Before we proceed to the Gaussian mechanism, we comment on one difference between pure DP and approximate DP. In the definition of pure DP, it was equivalent to consider $\Pr[M(X) = t]$ for all outcomes $t \in \mathcal{Y}$ (switching from PMFs to PDFs for continuous distributions, if necessary), compared to the way we usually state it, $\Pr[M(X) \in T]$ for all event $T \subseteq \mathcal{Y}$. However, this is not the case for approximate DP. This can be seen by considering an algorithm which simply outputs the dataset X and a random number from $\{1, \dots, 1/\delta\}$. Since the probability of every outcome t of this algorithm is at most δ, this would satisfy the inequality $\Pr[M(X) = t] \leq \Pr[M(X') = t] + \delta$, but it would not satisfy differential privacy nor any other type of reasonable privacy guarantee. Note that using the equivalent formulation in terms of the privacy loss random variable allows us to consider outcomes $t \in \mathcal{Y}$ once again.

Gaussian Mechanism

Now, we introduce the Gaussian mechanism. As the name suggests, this privatizes a statistic by adding Gaussian noise. Before we get to that, we require a slightly different notion of sensitivity.

Definition 3. *Let $f : \mathcal{X}^n \to \mathbb{R}^k$. The ℓ_2-sensitivity of f is*

$$\Delta_2^{(f)} = \max_{X, X'} \|f(X) - f(X')\|_2,$$

where X and X' are neighbouring databases.

Recall that for the Laplace mechanism, we added noise proportional to the ℓ_1-sensitivity. The ℓ_2 and ℓ_1 norms enjoy the following relationship: for a vector $x \in \mathbb{R}^d$, $\|x\|_2 \leq \|x\|_1 \leq \sqrt{d}\|x\|_2$. Thus, the ℓ_2 sensitivity might be up to a factor $\sqrt{d}$ less than the ℓ_1 sensitivity, which we will investigate an implication of later.

We recall the Gaussian distribution:

Definition 4. *The univariate* Gaussian distribution $N(\mu, \sigma^2)$ *with mean and variance μ and σ^2, respectively, has the following density:*

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

Visualizations of the density of the Gaussian distribution are provided in Figure 1.

The Gaussian mechanism is as follows:

Definition 5. *Let $f : \mathcal{X}^n \to \mathbb{R}^k$. The* Gaussian mechanism *is defined as*

$$M(X) = f(X) + (Y_1, \dots, Y_k),$$

where the Y_i are independent $N(0, 2\ln(1.25/\delta)\Delta_2^2/\varepsilon^2)$ random variables.

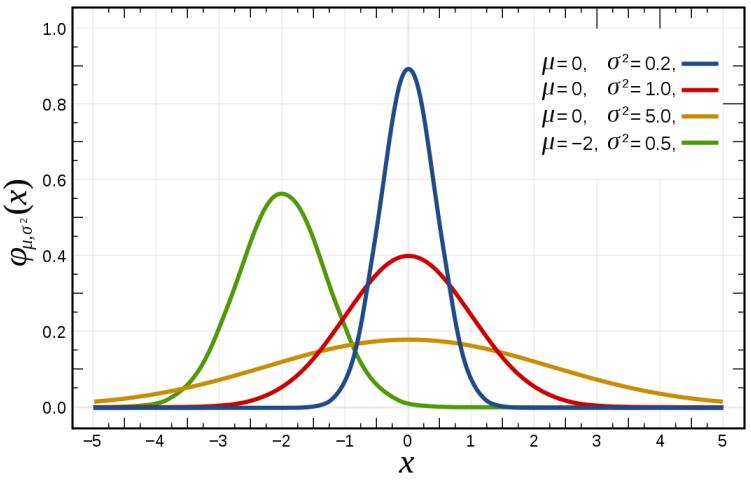

Figure 1: Figure from Wikipedia. Laplace distributions with various parameters.

Note that we can also write this using the multivariate Gaussian as $f(X) + Y$, where $Y \sim N(0, 2\ln(1.25/\delta)\Delta_2^2/\varepsilon^2 \cdot I)$. We claim that this algorithm is (ε, δ)-DP, which we will prove shortly:

Theorem 6. *The Gaussian mechanism is (ε, δ)-differentially private.*

To illustrate one difference between the Laplace and Gaussian mechanism, let's consider the problem of estimating the mean of a multivariate dataset. Suppose we have a dataset $X \in \{0,1\}^{n \times d}$, and we wish to privately estimate $f(X) = \frac{1}{n}\sum_{i=1}^n X_i$. The largest difference of this statistic between two neighbouring datasets is $\frac{1}{n}\vec{1}$. This is a vector with ℓ_1-norm of $\frac{d}{n}$, and ℓ_2-norm of $\frac{\sqrt{d}}{n}$, which define the ℓ_1 and ℓ_2 sensitivities, respectively. Using the Laplace mechanism to privatize f, we add Laplace noise of magnitude $\frac{d}{n\varepsilon}$ to each coordinate – this gives an ε-DP estimate of f with ℓ_2 error of magnitude $O(\frac{d^{3/2}}{n\varepsilon})$. On the other hand, if we use the Gaussian mechanism, we add Gaussian noise of magnitude $O(\frac{\sqrt{d\log(1/\delta)}}{n\varepsilon})$ to each coordinate – this gives an (ε, δ)-DP estimate of f with ℓ_2 error of magnitude (roughly) $O(\frac{d}{n\varepsilon})$. This example shows that the Gaussian mechanism can add a factor of $O(\sqrt{d})$ less noise (albeit for a marginally weaker privacy guarantee), thus indicating that in some cases it may be better suited for multivariate problems.

We now prove Theorem 6. For the sake of presentation, we are a bit informal in our derivation of the constant factor in the noise. For full details, see Appendix A of [DR14].

Recall the following basic fact about "linearity" of Gaussian distributions:

Fact 7. *If X and Y are i.i.d. $N(0,1)$, and a, b are constants, then $aX + bY \sim N(0, a^2 + b^2)$.*

We start by proving the following lemma on the privacy loss random variable.

Lemma 8. *Let $X, X' \in \mathcal{X}^n$ be neighbouring datasets, and let $M(Y) = f(Y) + N(0, \sigma^2 I)$ for some function $f : \mathcal{X}^n \to \mathbb{R}^k$. Then the privacy loss random variable between $M(X)$ and $M(X')$ is distributed as $N\left(\frac{\|f(X)-f(X')\|_2^2}{2\sigma^2}, \frac{\|f(X)-f(X')\|_2^2}{\sigma^2}\right)$.*

Proof. Without loss of generality, assume $f(X) = f(X') + v$. Consider drawing a noise magnitude

$x \sim N(0, \sigma^2 \cdot I)$. Then the privacy loss random variable is distributed as:

$$\begin{aligned}
\ln\left(\frac{\Pr[M(X) = f(X) + x]}{\Pr[M(X') = f(X) + x]}\right) &= \ln\left(\frac{\exp\left(-\|x\|_2^2/2\sigma^2\right)}{\exp\left(-\|x+v\|^2/2\sigma^2\right)}\right) \\
&= \left(-\frac{1}{2\sigma^2}\right)\left(\|x\|_2^2 - \|x+v\|_2^2\right) \\
&= \left(-\frac{1}{2\sigma^2}\right)\left(\sum_{j=1}^{k} x_j^2 - (x_j+v_j)^2\right) \\
&= \left(\frac{1}{2\sigma^2}\right)\left(\sum_{j=1}^{k} 2x_j v_j + v_j^2\right)
\end{aligned}$$

At this point, if we wanted only the univariate case ($k = 1$), we could essentially stop now:

$$\left(\frac{1}{2\sigma^2}\right)\left(2xv + v^2\right) = \frac{v}{\sigma^2}x + \frac{v^2}{2\sigma^2}$$

Since $x \sim N(0, \sigma^2)$, this privacy loss random variable is distributed with mean $\frac{v^2}{2\sigma^2}$, and variance $\frac{v^2}{\sigma^4} \cdot \sigma^2 = \frac{v^2}{\sigma^2}$, as desired (Fact 7 is used to derive the variance). But let's be brave and continue with the mulrivariate case.

We first inspect the constant term, which does not multiply the x_j's:

$$\frac{1}{2\sigma^2}\sum_{j=1}^{k} v_j^2 = \frac{\|v\|_2^2}{2\sigma^2}.$$

This matches the desired mean of the distribution. Turning to the other term:

$$\left(\frac{1}{2\sigma^2}\right)\left(\sum_{j=1}^{k} 2x_j v_j\right) = \frac{y}{\sigma^2},$$

where $y \sim N\left(0, \sigma^2 \sum_{j=1}^{k} v_j^2\right) = N\left(0, \sigma^2\|v\|_2^2\right)$, and we used Fact 7 to sum the x_i's into a single Gaussian. Using this fact one more time gives the variance of y/σ^2 to be $\|v\|_2^2/\sigma^2$, completing the proof. □

The lemma says that, under the Gaussian mechanism, the privacy loss random variable is Gaussian (note that this is a nice coincidence, and doesn't hold in general). Letting $Z \sim N(0, 1)$, then the privacy loss random variable can be rewritten as

$$\frac{\|f(X) - f(X')\|_2}{\sigma} Z + \frac{\|f(X) - f(X')\|_2^2}{2\sigma^2}.$$

Recall: our goal is to prove (ε, δ)-DP, which is done by proving the absolute value of the privacy loss random variable exceeds ε with probability at most δ. With this in mind, we rewrite the probability it exceeds ε as

$$\Pr\left[|Z| \geq \frac{\varepsilon\sigma}{\|f(X) - f(X')\|_2} - \frac{\|f(X) - f(X')\|_2}{2\sigma}\right].$$

Choosing $\sigma = \frac{\Delta_2 t}{\varepsilon}$ (for some t to be specified) allows us to upper bound this as

$$\Pr\left[|Z| \geq t - \frac{\varepsilon}{2t}\right].$$

At this point, we will be a bit informal and drop the latter term for the sake of presentation – we now consider

$$\Pr\left[|Z| \geq t\right].$$

But this is amenable to standard Gaussian tail bounds, such as

$$\Pr[Z \geq v] \leq \exp(-v^2/2).$$

Using this statement with $t = \sqrt{2\log(2/\delta)}$ gives

$$\Pr\left[|Z| \geq t\right] \leq \delta,$$

thus proving (ε, δ)-differential privacy.

Properties of Approximate Differential Privacy

Many of the convenient properties of pure differential privacy carry over to the approximate differential privacy setting. We simply state and discuss them, one can refer to [DR14, Vad17] for proofs.

Post-Processing

Closure under post-processing still holds: if an algorithm is (ε, δ)-DP, then any post-processing is also (ε, δ)-DP.

Theorem 9. *Let $M : \mathcal{X}^n \to \mathcal{Y}$ be (ε, δ)-differentially private, and let $F : \mathcal{Y} \to \mathcal{Z}$ be an arbitrary randomized mapping. Then $F \circ M$ is (ε, δ)-differentially private.*

Group Privacy

Group privacy, when we consider datasets which differ in k entries instead of 1, is not quite as clean under approximate DP in comparison to pure DP. As we have already seen, ε scales linearly with k, but the δ has an additional factor of $e^{(k-1)\delta}$.

Theorem 10. *Let $M : \mathcal{X}^n \to \mathcal{Y}$ be an (ε, δ)-differentially private algorithm. Suppose X and X' are two datasets which differ in exactly k positions. Then for all $T \subseteq \mathcal{Y}$, we have*

$$\Pr[M(X) \in T] \leq \exp(k\varepsilon)\Pr[M(X') \in T] + ke^{(k-1)\varepsilon}\delta.$$

(Basic) Composition

Finally, we revisit composition, in which we run k private analyses on the same sensitive dataset. Conveniently, the εs and δs add up to give a final privacy guarantee.

Theorem 11. *Suppose $M = (M_1, \ldots, M_k)$ is a sequence of algorithms, where M_i is $(\varepsilon_i, \delta_i)$-differentially private, and the M_i's are potentially chosen sequentially and adaptively.*[2] *Then M is $(\sum_{i=1}^k \varepsilon_i, \sum_{i=1}^k \delta_i)$-differentially private.*

Now, we have the language to describe the advanced composition theorem [DRV10], though we will only formally state and prove it next lecture. If all $\varepsilon_i = \varepsilon$, and all $\delta_i = \delta$, then M will overall be $(\varepsilon\sqrt{8k\ln(1/\delta')}, k\delta + \delta')$-DP. Observe that this only pays a multiplicative $O(\sqrt{k})$ factor in the value of ε, compared to basic composition which incurs a factor of k.

References

[BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Proceedings of the 14th Conference on Theory of Cryptography*, TCC '16-B, pages 635–658, Berlin, Heidelberg, 2016. Springer.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '06, pages 486–503, Berlin, Heidelberg, 2006. Springer.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[DR16] Cynthia Dwork and Guy N. Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 51–60, Washington, DC, USA, 2010. IEEE Computer Society.

[Mir17] Ilya Mironov. Rényi differential privacy. In *Proceedings of the 30th IEEE Computer Security Foundations Symposium*, CSF '17, pages 263–275, Washington, DC, USA, 2017. IEEE Computer Society.

[RRUV16] Ryan M. Rogers, Aaron Roth, Jonathan Ullman, and Salil Vadhan. Privacy odometers and filters: Pay-as-you-go composition. In *Advances in Neural Information Processing Systems 29*, NIPS '16, pages 1921–1929. Curran Associates, Inc., 2016.

[Smi20] Adam Smith. Lectures 9 and 10. https://drive.google.com/file/d/1M_GfjspEV2oaAuANKn2NJPYTDm1Mek0q/view, 2020.

[2] While the algorithms themselves may be sequentially and adaptively chosen, the privacy parameters may not be – see [RRUV16] for more discussion.

[Vad17] Salil Vadhan. The complexity of differential privacy. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, chapter 7, pages 347–450. Springer International Publishing AG, Cham, Switzerland, 2017.