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:

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

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 $\{1, \ldots, 1/\delta\}$. 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*

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

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

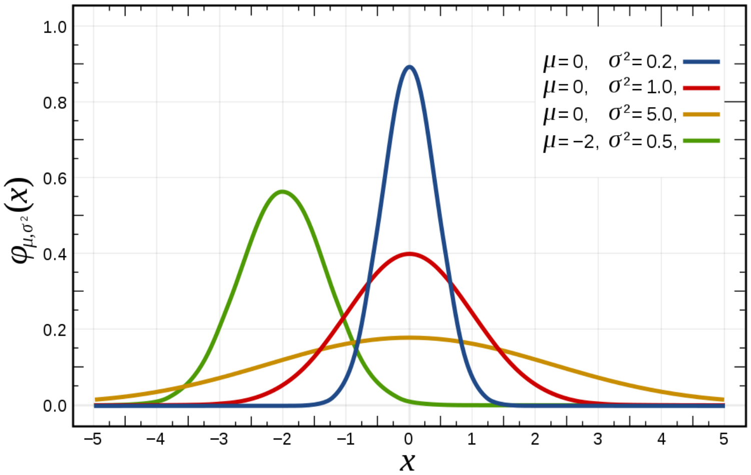

Figure 1: Figure from Wikipedia. Laplace distributions with various parameters.

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_GfjspEV2oaAuANKn2NJPYTDm1MekOq/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.