These lecture notes have not undergone rigorous peer-review. Please email quanquan.liu@yale.edu if you see any errors.

1 Introduction

Today we'll conclude our brief overview of differential privacy with a discussion of the Gaussian mechanism (together with advanced composition), exponential mechanism, and privacy amplification via subsampling. Thus far, we have focused on randomized response and the Laplace/Geometric mechanisms. As we discussed in previous classes, randomized response is well-suited for the local differential privacy model and for settings where we are looking for a cumulative (counting) query. On the other hand, the Laplace and Geometric mechanisms are well-suited for situations where the sensitivity of a (deterministic) function (solving a problem) is small. The Laplace/Geometric mechanism generally results in less error than randomized response but does not (immediately) satisfy local privacy. Today we'll discuss *approximate differential privacy* and mechanisms/tools we can take advantage of in the approximate DP setting.

As a reminder of the definition of differential privacy, approximate DP is the setting where $\delta > 0$.

Theorem 1 ((Central) Differential Privacy (DP) Model [DMNS06])**.** *Let $\varepsilon > 0$ and $\delta \in [0, 1)$. A randomized algorithm $\mathcal{A}$ is (ε, δ)-*differentially private (DP) *(with respect to the neighbor relation on the universe of the datasets) if for all events S in the output space of $\mathcal{A}$ and all neighboring datasets X and X',*

$$\Pr[\mathcal{A}(X) \in S] \leq \exp(\varepsilon) \cdot \Pr[\mathcal{A}(X') \in S] + \delta.$$

The interpretation of the approximate DP definition is that with probability $1 - \delta$, we obtain the privacy guarantee afforded by pure DP. But with probability δ, we get *no guarantee at all*. No guarantee means that the entire privacy dataset could be the output of the algorithm! Thus, it is important to to pick a very small δ, ie. $\delta = \frac{1}{\text{poly}(n)}$. Approximate DP like pure DP satisfy sequential composition, parallel composition, and post-processing. The only difference is that composing two approximate DP algorithms with parameters ε_1, δ_1 and ε_2, δ_2 gives a $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-DP algorithm.

2 Gaussian Mechanism

The Gaussian mechanism is an alternative to the Laplace mechanism that *does not satisfy* ε-DP but does satisfy (ε, δ)-DP. The Gaussian mechanism requires a slightly different notion of sensitivity than what we have seen thus far in this class. Namely, it requires the ℓ_2 sensitivity defined as follows. Given that this class focuses on graphs, we let our input neighboring datasets be two edge-neighboring graphs, G, G':

Theorem 2 (ℓ_2-Sensitivity)**.** *Given a function, $f : G \to \mathbb{R}^n$, the ℓ_2-sensitivity of f, denoted by Δ_f^2, is the maximum ℓ_2 distance between the outputs of f on two neighboring datasets $G \sim G'$,*

$$\Delta_f^2 = \max_{G \sim G'} ||f(G) - f(G')||_2.$$

The Gaussian mechanism is defined over the Gaussian distribution $\mathcal{N}(0, \sigma^2)$ with mean 0 and standard deviation σ has density function defined by:

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

Below is a plot of the Gaussian distribution alongside the Laplace distribution:

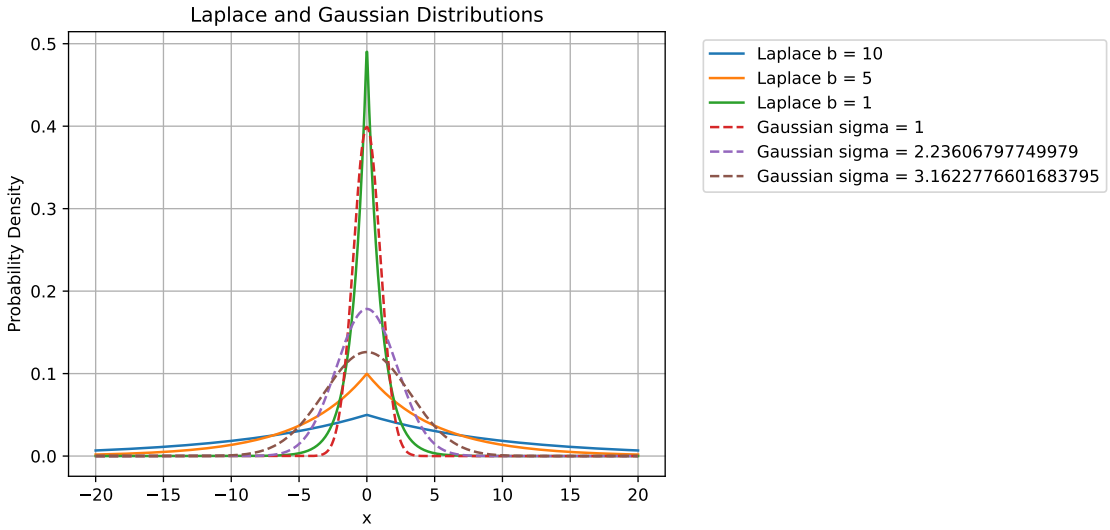

Figure 1: Laplace and Gaussian Distributions with Different Parameters

In high dimensional settings, where the output of the function is some n-dimensional vector of real values from $\mathbb{R}$, the Gaussian mechanism is defined as follows.

Theorem 3 (Gaussian Mechanism)**.** *Given a function $f : G \to \mathbb{R}^n$, the* **Gaussian mechanism** *defines an algorithm $M(G)$ which outputs*

$$M(G) = f(G) + (Y_1, \ldots, Y_n)$$

where each Y_i is an independent random variable drawn from $\mathcal{N}\left(0, \frac{2\ln(5/(4\delta))\left(\Delta_f^2\right)^2}{\varepsilon^2}\right)$.

As we showed above, the Gaussian mechanism has a flatter appearance compared to the Laplace mechanism; furthermore, it provides a less strong privacy guarantee. So why would we use it? Let me give you some intuition for situations where it is preferable to use the Gaussian mechanism over the Laplace mechanism. So far in this class, we've mainly considered functions which output a single real number. However, there are many cases for which we may want vector-valued functions. Such examples of vector valued functions include histograms. The output is a vector where each element of the vector is the number of elements in a bin. To compute the sensitivity of vector valued functions, we consider the ℓ_1 and ℓ_2 norms. As a recap, the ℓ_1 sensitivity is defined as $\Delta_f = \max_{G \sim G'} ||f(G) - f(G')||_1$ and the ℓ_2 sensitivity is defined as $\Delta_f^2 = \max_{G \sim G'} ||f(G) - f(G')||_2$. Suppose we have a function that returns a vector with *element-wise* sensitivity 1 (i.e. each coordinate has sensitivity 1, then the ℓ_1 sensitivity of this function is n. On the other hand, the ℓ_2 sensitivity of this function is $\sqrt{n}$.

Second, recall that approximate DP allows for a *catastrophic mode* where with probability δ, the entire private dataset could be released. However, the Gaussian mechanism *does not fail catastrophically*, instead it fails *gracefully*. With probability δ, it doesn't satisfy ε-DP but instead satisfies a weaker $c \cdot \varepsilon$-DP for some value c.

Finally, the Gaussian mechanism allows for the use of a stronger form of composition theorem called *advanced composition*. In advanced composition, instead of summing the privacy parameters of each of the

algorithms involved in the composition, we instead have the following lemma.

Theorem 2.1 (Advanced Composition)**.** *Let $M_1, M_2, \ldots, M_k$ be k randomized algorithms $M_i : G \to \mathcal{Y}$, where M_i is (ϵ, δ)-differentially private for each $i \in [k]$. Define $M(x) = (M_1(G), \ldots, M_k(G)))$ where each M_i is run independently. Then, for any $\delta' > 0$, the mechanism M is $(\epsilon', k \cdot \delta + \delta')$-differentially private, where*

$$\epsilon' = \sqrt{2k \ln(1/\delta')} \cdot \varepsilon + k\varepsilon(e^{\varepsilon} - 1).$$

This theorem shows that the privacy loss parameters ϵ and δ scale roughly with $\sqrt{k}$ and k, respectively, when composing k differentially private mechanisms.

The above essentially allows for $\varepsilon' = O(k\varepsilon^2 + \sqrt{k\varepsilon^2})$ privacy guarantee which is significantly smaller than $k\varepsilon$ afforded by basic composition when k is large.

For the sake of time, we did not cover the proof of the privacy of the Gaussian mechanism in class, but for completeness, the detailed proof of the privacy of the Gaussian mechanism can be found on pg. 261-265 of the The Algorithmic Foundations of Differential Privacy textbook [DR+14].

3 Exponential Mechanism

So far, we have focused on adding noise to an aggregate cumulative value. However, for certain applications, we might not want to add noise to a precise value to obtain an approximate value but instead we want to be able to return the *precise exact* solution while preserving differential privacy. Suppose answers to a problem are ranked by value and we want to select the best possible answer out of the ranked values. The exponential mechanism allows just that: to select the best valued answer, exactly, but privately; however, sometimes the best valued answer is not selected in the interest of preserving privacy.

More formally, the exponential mechanism operates over a *scoring function* which outputs a score for each element in a set of elements. The mechanism then *approximately* maximizes the score of the element it returns. Take a concrete example use of this mechanism. Suppose we have a bipartite graph consisting of bidders and items and each edge represents a value of a bidder for an item. No bidder would buy an item priced at a greater value than their preference. We would like to find a pricing of the items to maximize profit. This problem corresponds with the maximum bipartite weighted matching problem where we want to return an (exact) price of values for each item to maximize the profit equal to its maximum weighted adjacent edge. One may think that one can add noise to a price to achieve privacy but suppose an item is connected to three bidders with valuations at \$1 , \$2, and \$5. Pricing the item at \$5 achieves maximum profit but increasing the price beyond \$5 (e.g. with noise) results in \$0 profit!

Thus, in this situation, we would like to return an approximately best ranked solution from the set without adding noise to the item prices. Hence, we define the exponential mechanism as follows:

Theorem 4 (Exponential Mechanism)**.** *The exponential mechanism takes as input a private input dataset X. It also takes a (public) scoring function $u : X \times \mathcal{H} \to \mathbb{R}$ where $\mathcal{H}$ is a (public) set of objects (all possible outcomes) and the scoring function returns for each object $h \in \mathcal{H}$ is with respect to X. The sensitivity is defined as*

$$\Delta_u = \max_{h \in \mathcal{H}} \max_{X \sim X'} |u(X, h) - u(X', h)|,$$

over neighboring datasets X and X'. The **exponential mechanism** *$M(X, \mathcal{H}, u)$ selects and outputs*

object $h \in \mathcal{H}$ with probability proportional to $\exp\left(\frac{\varepsilon \cdot u(X,h)}{2\Delta_u}\right)$.

There are some important characteristics of the exponential mechanism that we have not yet seen with our other mechanisms:

1. The privacy cost of the mechanism is ε, regardless of the size of the size of the item set $\mathcal{H}$.
2. The mechanism works for finite and infinite sets $\mathcal{H}$ but will be practically challenging to use if the dataset is infinite.
3. All other ε-differentially private mechanisms can be defined in terms of the exponential mechanism for appropriate definitions of the scoring function.
4. For many situations, the exponential distribution gives the "best" possible outcome, although it may not be computationally feasible.

We now prove the privacy and utility of the exponential mechanism.

Lemma 3.1. *The exponential mechanism is ε-differentially private.*

Proof. We fix the neighboring datasets X and X' as well as some output $h \in \mathcal{H}$. Then, given the exponential mechanism M with privacy parameter ε, we have that

$$\begin{aligned}
\frac{\mathsf{Pr}(M(X)=h)}{\mathsf{Pr}(M(X'=h)} &= \frac{\frac{\exp\left(\frac{\varepsilon \cdot u(X,h)}{2\Delta_u}\right)}{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon \cdot u(X,h')}{2\Delta_u}\right)}}{\frac{\exp\left(\frac{\varepsilon \cdot u(X',h)}{2\Delta_u}\right)}{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon \cdot u(X',h')}{2\Delta_u}\right)}} \\
&= \exp\left(\frac{\varepsilon \cdot (u(X,h) - u(X',h))}{2\Delta_u}\right)\left(\frac{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon \cdot u(X',h')}{2\Delta_u}\right)}{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon \cdot u(X,h')}{2\Delta_u}\right)}\right) \\
&\leq \exp\left(\frac{\varepsilon}{2}\right) \cdot \left(\frac{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon \cdot u(X',h')}{2\Delta_u}\right)}{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon \cdot u(X,h')}{2\Delta_u}\right)}\right) \quad \text{by our definition of } \Delta_u \\
&\leq \exp\left(\frac{\varepsilon}{2}\right) \cdot \left(\frac{\exp\left(\frac{\varepsilon}{2}\right) \cdot \sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon \cdot u(X,h')}{2\Delta_u}\right)}{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon \cdot u(X,h')}{2\Delta_u}\right)}\right) = \exp(\varepsilon).
\end{aligned}$$

The last inequality follows since $\exp\left(\frac{\varepsilon \cdot u(X',h')}{2\Delta_u}\right) \leq \exp\left(\frac{\varepsilon \cdot u(X,h')}{2u(X',h')}\right) \cdot \exp\left(\frac{\varepsilon \cdot u(X',h')}{2\Delta_u}\right) = \exp\left(\frac{\varepsilon}{2}\right) \cdot \exp\left(\frac{\varepsilon \cdot u(X',h')}{2\Delta_u}\right)$. □

Finally, we compute the utility of the exponential mechanism to be:

Lemma 3.2. *Let X be a dataset and $\mathrm{OPT}(X) = \max_{h \in \mathcal{H}} u(X,h)$ be the best score obtained from all possible outcomes. Let $\mathcal{H}^* = \{h \in \mathcal{H} \mid u(X,h) = \mathrm{OPT}(X)\}$ be the set of objects which achieve $\mathrm{OPT}(X)$. Then,*

$$\mathsf{Pr}\left(u(M(X)) \leq \mathrm{OPT}(X) - \frac{2\Delta_u}{\varepsilon}\left(\ln\left(\frac{|\mathcal{H}|}{|\mathcal{H}^*|}\right) + t\right)\right) \leq \exp(-t).$$

Proof. First, we bound the probability that the score of $M(X)$ is less than a general parameter C. We can compute this probability by first calculating the probability that we pick an outcome h with value less than or equal to C. This probability is given by $\sum_{h\in\mathcal{H}|u(X,h)\leq C}\exp\left(\frac{\varepsilon u(X,h)}{2\Delta_u}\right)$. This expression is upper bounded by $|\mathcal{H}|\cdot\exp\left(\frac{\varepsilon C}{2\Delta_u}\right)$ since there are at most $|\mathcal{H}|$ possible outcomes and the score is upper bounded by C. Then, the denominator consists of all possible outcomes $\sum_{h\in\mathcal{H}}\exp\left(\frac{\varepsilon u(X,h)}{2\Delta_u}\right)$. This denominator is lower bounded by $|\mathcal{H}^*|\exp\left(\frac{\varepsilon \mathrm{OPT}(X)}{2\Delta_u}\right)$ since there are at least $|H^*|$ outcomes with score at least $\mathrm{OPT}(X)$. Altogether, combining these two expressions we obtain,

$$\mathsf{Pr}(u(M(X))\leq C)\leq\frac{|\mathcal{H}|\cdot\exp\left(\frac{\varepsilon C}{2\Delta_u}\right)}{|\mathcal{H}^*|\exp\left(\frac{\varepsilon\mathrm{OPT}(X)}{2\Delta_u}\right)}=\frac{|\mathcal{H}|}{|\mathcal{H}^*|}\cdot\exp\left(\frac{\varepsilon(C-\mathrm{OPT}(X))}{2\Delta_u}\right).$$

Substituting the given C gives our desired result. □

4 Privacy Amplification by Subsampling

The idea behind privacy amplification by subsampling is that we run a DP algorithm on some random subset of the input data. This random sampling procedure inherently introduces additional uncertainly, which we can use for privacy. In particular, sampling allows for the possibility that any one datapoint is not used in the algorithm, which benefits the privacy of your data. Furthermore, the adversary does not *know* whether your data is being used, further benefitting privacy. Subsampling arises in the sketching literature and hence arise in many algorithms that deal with massive datasets. Subsampling also has applications to stochastic gradient descent methods that use minibatch training.

We'll go over a simple version of the privacy amplification argument. Namely, we can amplify the privacy of an ε-DP algorithm M by creating a new algorithm M' that is $2\varepsilon^2$-DP that runs M on a random subsample of size $\varepsilon\cdot n$. We define our neighboring datasets to be two datasets X, X' where $X\oplus X'=\{j\}$; in words, there exists exactly one element $j\in X$ that is not contained in X' and all elements in X' are contained in X.

Lemma 4.1 (Privacy Amplification by Subsampling)**.** *Given $\varepsilon\in(0,1)$ and an ε-DP algorithm $\mathcal{A}$, then an algorithm $\mathcal{A}'$ which samples each element of dataset X (with n elements) with probability ε and runs $\mathcal{A}$ on the sampled dataset is $2\varepsilon^2$-DP.*

Proof. We fix the output y. Given two input datasets $X\sim X'$ that differ in element j, we consider a run of $\mathcal{A}'$ on input X. We introduce a coupling between the random sampling process on X and X'. Namely, if an entry i is sampled from X, then we also add the entry i to the sample from X'. Since the samples are determined independently, we easily see that this is a correct coupling. For ease of exposition, we denote the random variable representing the sample from X as S_X and the sample from X' as $S_{X'}$. If data point j is not included in S_X, then, the distribution of outputs on S_X and $S_{X'}$ are identical since each data point is sampled independently. If instead, entry j is included in S_X, then the probability distributions are a factor of e^ε off from each other by the guaranteed privacy of the algorithm. Formally, we have

$$\mathsf{Pr}[\mathcal{A}(S_X)=y\mid j\notin S_X]=\mathsf{Pr}[\mathcal{A}(S_{X'})=y],$$

and, let $X'=X\setminus\{j\}$ by our coupling and since $\mathcal{A}$ is ε-DP,

$$e^{-\varepsilon} \cdot \mathsf{Pr}[\mathcal{A}(S_{X'}) = y] \leq \mathsf{Pr}[\mathcal{A}(S_X) = y \mid j \in S_X] \leq e^{\varepsilon} \cdot \mathsf{Pr}[\mathcal{A}(S_{X'}) = y].$$

Now, using the above and $\mathsf{Pr}[j \in S_X]$, we can show

$$\begin{aligned}
\mathsf{Pr}[\mathcal{A}(S_X) = y] &= (1-\varepsilon) \cdot \mathsf{Pr}[\mathcal{A}(S_X) = y \mid j \notin S_X] + \varepsilon \cdot \mathsf{Pr}[\mathcal{A}(S_X) = y \mid j \in S_X] \\
&\leq (1-\varepsilon) \cdot \mathsf{Pr}[\mathcal{A}(S_{X'}) = y] + \varepsilon \cdot e^{\varepsilon} \cdot \mathsf{Pr}[\mathcal{A}(S_{X'}) = y] \\
&\leq (1 + \varepsilon \cdot (e^{\varepsilon} - 1)) \cdot \mathsf{Pr}[\mathcal{A}(S_{X'}) = y] \\
&\leq e^{2\varepsilon} \cdot \mathsf{Pr}[\mathcal{A}(S_{X'}) = y].
\end{aligned}$$

and

$$\begin{aligned}
\mathsf{Pr}[\mathcal{A}(S_X) = y] &\geq (1-\varepsilon) \cdot \mathsf{Pr}[\mathcal{A}(S_{X'}) = y] + \varepsilon \cdot e^{-\varepsilon} \cdot \mathsf{Pr}[\mathcal{A}(S_{X'}) = y] \\
&= (1 - \varepsilon \cdot (1 - e^{-\varepsilon})) \cdot \mathsf{Pr}[\mathcal{A}(S_{X'}) = y] \\
&\geq e^{-\varepsilon} \cdot \mathsf{Pr}[\mathcal{A}(S_{X'}) = y],
\end{aligned}$$

when $\varepsilon \in (0, 1)$. □

References

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

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.