Privacy Enhancing Technologies FS2025
Lecture 26-27-28 – Approximate Differential Privacy

Florian Tramèr

AGENDA

1. Approximate DP
2. The Privacy Loss Variable
3. The Gaussian Mechanism
4. Advanced Composition
5. The Exponential Mechanism

Recap

Last time we saw the definition of differential privacy and some of its nice properties as a privacy measure. We also saw two simple algorithms that achieve differential privacy: Randomized Response and the Laplace mechanism.

We will now see other popular algorithms (or "mechanisms") that achieve differential privacy, which improve upon the Laplace mechanism in two ways:

1. We will see a relaxation of DP, called approximate DP, which allows for much better error rates in some settings, at the cost of a very small probability of privacy violation.
2. We will introduce the Exponential Mechanism, which allows us to answer more general queries than numerical-valued ones.

1 Approximate Differential Privacy

It turns out that our previous analysis of the Laplace mechanism for counting queries is tight: one can show that *any* algorithm that is ε-differentially private and answers k normalized counting queries must incur error of magnitude $\Omega(k/\varepsilon n)$ (this result is due to [HT10], see [Vad17, §5.2] for a nice overview).

Unless we relax our notion of privacy, we cannot do any better. So that's exactly what we'll do! Notice that the definition of DP asks that *any* event can be at most e^{ε} times more likely if we swap out one individual. This includes, in particular, events with a tiny probability. For example, suppose that some y has probability 2^{-128} of being output on database D and probability 0 on database D'. Of course, observing y is a "smoking gun" that the input database cannot have been D, and so this algorithm is not ε-DP for any finite ε. And yet, should we really care about this? Our cryptographic schemes also fail with similar probability.

This motivates the following relaxation of DP, where we allow events to also deviate by a small *additive* factor δ:

Definition 1 (Approximate Differential Privacy [DKM+06]). An algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ is (ε, δ)-differentially private if, for all neighboring databases D and D' and all events $S \subseteq \mathcal{Y}$, we have

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \cdot \Pr[M(D') \in S] + \delta \,.$$

Interpreting approximate DP.

- $(\varepsilon, 0)$-DP is equivalent to the "pure" ε-DP definition from the previous lecture.
- We can think of δ as the "failure probability" of DP, and (informally) interpret (ε, δ)-DP as saying that the algorithm is ε-DP with probability at least $1 - \delta$, while providing no guarantee with probability at most δ (we will formalize this later on).

  However, this analogy can be overly pessimistic, and many algorithms that are (ε, δ)-DP (such as the Gaussian mechanism below) actually degrade "gracefully". That is, with probability δ, the algorithm still provides a DP guarantee, but with a larger ε.
- Ideally, we would set δ to be "cryptographically small" (e.g., 2^{-80}). In practice, however, we might set δ to be much larger because of the cost of composing many differentially private algorithms. For example, differentially private machine learning (which we will see later) commonly uses values such as $\delta = 1/n^2$.
- Notably, as you'll see in the homework, the factor δ *must* be $o(1/n)$ to get a meaningful privacy guarantee.
- Recall from the last lecture that when dealing with pure DP, the definition was equivalent (for discrete outputs) to just asking that $\Pr[M(D) = y] \leq e^{\varepsilon} \cdot \Pr[M(D') = y]$ for all $y \in \mathcal{Y}$. This is *not* the case for approximate DP, where we have to consider all possible subsets of outputs rather than just single outputs (see [DR14] for details).

Properties of approximate DP. Approximate DP has similar properties to pure DP. In particular, (ε, δ)-DP is closed under postprocessing and scales gracefully to larger groups (while the multiplicative factor ε still scales linearly with the group size, the scaling of δ is a bit weirder, and we won't dwell on it here, see [DR14] or [Vad17]).

Approximate DP also satisfies a basic composition theorem:

Theorem 1 (Basic composition of approximate DP). Let $M = (M_1, M_2, \ldots, M_k)$ be a sequence of algorithms, where M_i is $(\varepsilon_i, \delta_i)$-differentially private, and the algorithms can be chosen sequentially and adaptively. Then M is $(\sum_{i=1}^k \varepsilon_i, \sum_{i=1}^k \delta_i)$-differentially private.

We won't prove this one here. As we'll see in a later lecture, approximate DP actually allows for a much more powerful composition theorem, where the factor ε scales as $O(\sqrt{k})$ instead of $O(k)$. This is a crucial advantage of approximate DP over pure DP: if we are composing many different differentially private algorithms, we can get away with a much smaller ε than with pure DP.

2 The Privacy Loss Variable

To prove that a mechanism was ε-DP, it was enough to just focus on singleton events $S = \{y\}$ (for a discrete output space $\mathcal{Y}$), since we have that

$$\begin{aligned} &\forall y \in \mathcal{Y}, \Pr[M(D) = y] \leq e^{\varepsilon} \cdot \Pr[M(D') = y] \\ \implies &\forall S \subseteq \mathcal{Y}, \Pr[M(D) \in S] \leq e^{\varepsilon} \cdot \Pr[M(D') \in S] \end{aligned}$$

Unfortunately this implication does not hold for approximate DP, so we now need to explicitly consider all possible subsets of outputs $S \subseteq \mathcal{Y}$. A useful strategy is to prove that if we draw $y \leftarrow M(D)$, then with high probability we have that $\Pr[M(D) = y] \leq e^{\varepsilon} \cdot \Pr[M(D') = y]$. We formalize this as follows:

Definition 2 (Privacy Loss Random Variable)**.** Let D and D' be two databases, and M be a mechanism. The *privacy loss random variable* $\mathcal{L}_{M(D)||M(D')}$ is distributed by drawing $y \leftarrow M(D)$, and outputting

$$\mathcal{L}_{M(D)||M(D')}(y) := \ln\left(\frac{\Pr[M(D) = y]}{\Pr[M(D') = y]}\right) \tag{1}$$

If either the numerator or denominator is 0, we say that the privacy loss is infinite.

We can also define this for mechanisms with continuous outputs, by considering the ratio of their densities.

Note that a mechanism M being ε-DP is equivalent to saying that $|\mathcal{L}_{M(D)||M(D')}|$ is bounded by ε for all neighboring databases D and D'.

For approximate DP, we have that a mechanism M is (ε, δ)-DP if for all neighboring databases D and D',

$$\Pr[|\mathcal{L}_{M(D)||M(D')}| > \varepsilon] \leq \delta \tag{2}$$

Note that the converse is not quite true: (ε, δ)-DP does not imply Equation (2), although a slightly more complex of this equivalence can be shown (see e.g., [Ste22]).

3 The Gaussian Mechanism

The nice thing about approximate DP is that it will allow us to work with Gaussian noise, which is much more convenient than Laplace noise (for reasons we won't get into here).

Due to the geometry of the (multivariate) Gaussian distribution, we will now consider functions that have small sensitivity in the ℓ_2 norm rather than the ℓ_1 norm.

Definition 3 (ℓ_2-Sensitivity)**.** The ℓ_2-sensitivity of a function $f\colon \mathcal{X}^n \to \mathbb{R}^k$ is defined as

$$\Delta_2 = \max_{D \sim D'} \|f(D) - f(D')\|_2 = \max_{D, D'} \sqrt{\sum_{i=1}^{k} (f(D)_i - f(D')_i)^2}\,,$$

where D and D' are neighboring databases.

An important thing to note is that the ℓ_2 norm is never more than the ℓ_1 norm, and can be as much as $\sqrt{k}$ times smaller:

$$\Delta_2 \leq \Delta_1 \leq \sqrt{k} \cdot \Delta_2$$

In particular, suppose that $f = (f_1, \ldots, f_k)$ is a sequence of k counting queries. Then f has ℓ_1-sensitivity $\Delta_1 = k/n$, while the ℓ_2-sensitivity is $\Delta_2 = \sqrt{k}/n$. As we will see, the Gaussian mechanism will let us scale the noise proportional to Δ_2 (and not Δ_1), and thus achieve a much better error.

Definition 4 (Gaussian Distribution)**.** The univariate Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and variance σ^2 has density

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad x \in \mathbb{R}\,.$$

Definition 5 (Gaussian Mechanism)**.** Let $f\colon \mathcal{X}^n \to \mathbb{R}^k$ be a function with ℓ_2-sensitivity Δ_2, and let $\varepsilon, \delta > 0$. The Gaussian mechanism is defined as

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

where the Y_i are i.i.d. samples from the $\mathcal{N}\left(0, \frac{2\log(1.25/\delta)\Delta_2^2}{\varepsilon^2}\right)$ distribution.

The values are a bit uglier here than for the Laplace mechanism, so let's quickly unwrap this. The standard deviation of the noise grows linearly with Δ/ε as for the Laplace mechanism, but we now use the ℓ_2-sensitivity instead of the ℓ_1-sensitivity, and we have an extra $O(\log(1/\delta))$ factor. So, in the univariate case (where $\Delta_2 = \Delta_1$), the Gaussian mechanism always adds noise of higher variance than the Laplace mechanism. The true benefit will come from answering multiple queries. Also note that the noise dependence on δ is only logarithmic, so we can set δ to be exponentially small while only incurring a constant multiplicative factor in the error.

Somewhat unsurprisingly at this stage:

Theorem 2. For any $\varepsilon \leq 1$ and $\delta > 0$, the Gaussian mechanism is (ε, δ)-differentially private.[a]

[a] We can also prove the theorem for $\varepsilon > 1$, with slightly different types of bounds.

Proof (informal). The proof is a bit involved in places, we'll do our best to get the gist of it. We'll also be somewhat informal with the constant $2\log(1.25)$ in the variance of the Gaussian noise. See [DR14] for a more rigorous proof.

We will focus on the univariate case for simplicity. Let σ be the standard deviation of the Gaussian noise and let D and D' be neighboring databases. Without loss of generality, we can assume that $f(D) = 0$ and $f(D') = \Delta_2$.

Let's now consider the privacy loss random variable, for $y \leftarrow M(D) = \mathcal{N}(0, \sigma^2)$:

$$\begin{aligned}
\mathcal{L}_{M(D)||M(D')}(y) &= \ln\left(\frac{\Pr[M(D) = y]}{\Pr[M(D') = y]}\right) \\
&= \ln\left(\frac{\exp\left(\frac{-y^2}{2\sigma^2}\right)}{\exp\left(\frac{-(y-\Delta_2)^2}{2\sigma^2}\right)}\right) \\
&= \frac{1}{2\sigma^2}\left((y-\Delta_2)^2 - y^2\right) \\
&= \frac{1}{2\sigma^2}\left(-2y\Delta_2 + \Delta_2^2\right)
\end{aligned}$$

Note that y is distributed as $\mathcal{N}(0, \sigma^2)$, and thus the Privacy Loss Random Variable is itself Gaussian,[1] with mean $\frac{\Delta_2^2}{2\sigma^2}$ and variance $\left(\frac{2\Delta_2}{2\sigma^2}\right)^2 \cdot \sigma^2 = \frac{\Delta_2^2}{\sigma^2}$.

Here's where we will get a bit informal. First, we'll pretend that the mean of this Gaussian is 0 (note that since $\varepsilon \leq 1$, the mean is indeed small). Second, we'll set $\sigma = \sqrt{2\log(2/\delta)}\Delta_2/\varepsilon$, thus slightly increasing the noise compared to Definition 5.

So now we just need to show that this random variable is bounded by ε with probability at least $1 - \delta$.

$$\begin{aligned}
\Pr[|\mathcal{L}_{M(D)||M(D')}| > \varepsilon] &\approx \Pr_{y \leftarrow \mathcal{N}\left(0, \frac{\Delta_2^2}{\sigma^2}\right)}[|y| > \varepsilon] \\
&= \Pr_{y \leftarrow \mathcal{N}(0,1)}[|y| > \sigma\varepsilon/\Delta_2] \\
&= \Pr_{y \leftarrow \mathcal{N}(0,1)}\left[|y| > \sqrt{2\log(2/\delta)}\right] \leq \delta
\end{aligned}$$

The last inequality follows from the following standard Gaussian tail bound:

> If $Z \sim \mathcal{N}(0, \sigma^2)$, then for every $t > 0$: $\Pr(|Z| > t \cdot \sigma) \leq 2\exp(-t^2/2)$.

□

Another way of writing Theorem 2 is that, if we add Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the output of a function f with ℓ_2-sensitivity Δ_2, then the resulting function is $(\varepsilon, \delta(\varepsilon))$-differentially private for every $\varepsilon > 0$, where $\delta(\varepsilon) = 1.25\exp(-\varepsilon^2\sigma^2/2\Delta_2^2)$. Thus, the same mechanism can be interpreted as providing many different (ε, δ)-DP guarantees. This can make the analysis of the optimal composition of such mechanisms somewhat tricky; hence, there also exist definitions that more cleanly characterize the full privacy profile of such mechanisms (e.g., Rényi DP [Mir17]).

3.1 Answering Counting Queries With the Gaussian Mechanism

So what have we gained? Let's go back to our (multiple) counting queries example, where we apply k normalized counting functions $f = (f_1, \ldots, f_k)$ to the same dataset. The ℓ_1-sensitivity of the function f is $\Delta_1 = k/n$, while the ℓ_2-sensitivity is $\Delta_2 = \sqrt{k}/n$.

[1] This is a neat property of the Gaussian distribution, and is not true in general for other mechanisms.

So to get the same privacy level ε as with the Laplace mechanism (but with the additional δ probability of error), we need to add Gaussian noise of standard deviation $\tilde{O}(\Delta_2/\varepsilon) = \tilde{O}(\sqrt{k}/\varepsilon n)$ to each coordinate of the output (we ignore the $\log(1/\delta)$ factor here for simplicity). We thus get error $\tilde{O}(\sqrt{k}/\varepsilon n)$ in each coordinate with high probability—a factor of $\sqrt{k}$ smaller than the error of the Laplace mechanism! It turns out that this is also the best we can do, for arbitrary counting queries.

4 Advanced Composition

Another way of interpreting the result of Section 3.1 is through *composition* of the Gaussian mechanism. We answer each counting query with Gaussian noise of standard deviation $\sigma = \tilde{O}(\sqrt{k}/\varepsilon n)$. By Theorem 2, this means that each query individually is (ε', δ)-differentially private, where $\varepsilon' = \tilde{O}(\varepsilon/\sqrt{k})$.

If we relied on the basic composition theorem, we would get that the total privacy budget is $\tilde{O}(\varepsilon\sqrt{k})$, which is a factor of $\sqrt{k}$ worse than the bound we got.

It turns out that this scaling behavior is not unique to the Gaussian mechanism, and holds more generally for the composition of *arbitrary* (ε, δ)-differentially private mechanisms, even ones chosen adaptively based on the previous answers.

Theorem 3 (Advanced composition [DRV10])**.** For all $\varepsilon, \delta, \delta' > 0$, let $M = (M_1, \ldots, M_k)$ be a sequence of (ε, δ)-differentially private algorithms, where the M_i's are potentially chosen sequentially and adaptively. Then M is $(\tilde{\varepsilon}, \tilde{\delta})$-differentially private, where $\tilde{\varepsilon} = \varepsilon\sqrt{2k\log(1/\delta')} + k\varepsilon\frac{e^\varepsilon - 1}{e^\varepsilon + 1}$ and $\tilde{\delta} = k\delta + \delta'$.

The formula for $\tilde{\varepsilon}$ is a bit complicated, so let's look at it asymptotically. If ε is small (say $\varepsilon \leq 1$), then the term $\frac{e^\varepsilon - 1}{e^\varepsilon + 1}$ is roughly $\varepsilon/2$. So we get that $\tilde{\varepsilon} = \Theta(\varepsilon\sqrt{2k\log(1/\delta')} + k\varepsilon^2)$, where the term $k\varepsilon^2$ that is linear in k is a lower-order term if ε is small enough. Specifically, if we have $\varepsilon < 1/\sqrt{k}$ and we set $\delta' = \delta$, then we get a clean bound of $\tilde{\varepsilon} = \Theta(\varepsilon\sqrt{k\log(1/\delta)})$ and $\tilde{\delta} = \Theta(k\delta)$.

This theorem applies to any (ε, δ)-differentially private algorithm! This thus shows a clear separation between pure DP (where composition does grow linearly in the worst-case) and approximate DP (where composition grows, roughly, as the square root of the number of steps).

We won't prove this theorem here, but we'll provide some intuition for it in the homework.

5 How Many Counting Queries Can We Answer Privately?

The table below summarizes the privacy-utility trade-offs for answering k counting queries, along with matching lower bounds (ignoring logarithmic factors in δ and k).

It turns out that the error bound for the Gaussian mechanism can further be beaten in settings where the data domain $|\mathcal{X}|$ is not too large. The state-of-the-art here is the *Private Multiplicative Weights* algorithm [HR10] which we won't cover (we'll see a weaker bound in the homework). This algorithm's error has only a logarithmic dependence on the number of queries k, and can thus answer *exponentially many* queries! (as long as the data domain is not exponentially larger than the database size n).

Guarantee	Error per query	Mechanism	Lower bound
Local ε-DP	$\tilde{O}\left(\frac{1}{\varepsilon\sqrt{n}}\right)$	Randomized Response	$\Omega\left(\frac{1}{\varepsilon\sqrt{n}}\right)$
ε-DP	$\tilde{O}\left(\frac{k}{\varepsilon n}\right)$	Laplace	$\Omega\left(\frac{k}{\varepsilon n}\right)$
(ε, δ)-DP	$\tilde{O}\left(\frac{\sqrt{k}}{\varepsilon n}\right)$	Gaussian	$\Omega\left(\frac{\sqrt{k}}{\varepsilon n}\right)$
	$\tilde{O}\left(\frac{\sqrt{\log \lvert\mathcal{X}\rvert \log k}}{\varepsilon n}\right)^{1/2}$	Private Multiplicative Weights	$\tilde{\Omega}\left(\frac{\sqrt{\log \lvert\mathcal{X}\rvert \log k}}{\varepsilon n}\right)^{1/2}$

6 Beyond Numerical Queries: The Exponential Mechanism

So far, we've seen mechanisms (Laplace, Gaussian) that are well suited for answering numerical queries with added noise. But what about situations where adding noise to an answer doesn't really make sense?

Let's consider a concrete example. Since we're in Switzerland, the example will obviously be related to voting. Suppose all students from ETH are asked to vote for their favorite class. Since students might like many classes, we'll use *approval voting*, where each student can cast a vote for as many classes as they want. If there are k classes, then each student's vote is a subset $x_i \subseteq [k]$. The score of the j-th class is the number of students who voted for it, i.e., $q(j; D) = |\{i : j \in x_i\}|$. The class with the most votes is the winner.

Let's now try to select the best class using differential privacy to protect each student's vote. We might not be able to select the absolute best class, but we can hope to select a class with a number of votes that is close to the highest (which will actually be the best class if there is a large enough gap between the best and second-best class).

Of course, we can't add noise to the output of the selection mechanism, since the output is a class, not a number. Instead, we could add noise to the aggregate scores and then rely on post-processing to select the class with the highest noisy score. Suppose we do this with the Laplace mechanism. The sensitivity of the score vector is k (a student can change each class's score by at most 1), so we'd have to add noise of standard deviation $\tilde{O}(k/\varepsilon)$ to each score to get ε-DP. By relaxing to approximate DP, we could instead get by with noise of standard deviation $\tilde{O}(\sqrt{k}/\varepsilon)$.

This is rather high, and we'd like to do better. We'll see two approaches to this problem: the Exponential Mechanism (below) and the "Report Noisy Max" mechanism (in the homework).

6.1 The Exponential Mechanism

The voting problem above is a special case of a general *selection* problem, which we define as follows.

Definition 6 (Selection Problem)**.** A *selection problem* is specified by:

- A set $\mathcal{Y}$ of possible outcomes (this may be discrete or continuous);
- A score function $q\colon \mathcal{Y} \times \mathcal{X}^n \to \mathbb{R}$ which measures the "quality" of an output for a given input data set $D \in \mathcal{X}^n$;
- A sensitivity bound Δ such that $q(y;\cdot)$ has sensitivity at most Δ for every $y \in \mathcal{Y}$. That is, for all $y \in \mathcal{Y}$, and all neighboring inputs D and D', we have

$$|q(y;D) - q(y;D')| \leq \Delta$$

In our voting example, the outcomes $\mathcal{Y}$ are the k classes, the input space $\mathcal{X}$ is the set of votes cast by a student (i.e., a subset of $[k]$), the score function returns the number of votes for a class, and the sensitivity bound is $\Delta = 1$ (a student can change each class's score by 1).

We can turn any selection problem differentially private using the following Exponential Mechanism. Intuitively, the mechanism defines a probability distribution over the outcomes $\mathcal{Y}$, such that each $y \in \mathcal{Y}$ is assigned a probability proportional to $\exp(\frac{\varepsilon}{2\Delta} q(y;D))$. So outcomes with higher scores are more likely to be selected, and the probability of selecting an outcome decays exponentially with the score difference.

Definition 7 (Exponential Mechanism)**.** Let $(\mathcal{X}, \mathcal{Y}, q, \Delta)$ be a selection problem. Then, return a random sample from the distribution over $\mathcal{Y}$ defined by

$$\Pr[Y = y] \propto \exp\left(\frac{\varepsilon}{2\Delta} q(y;D)\right)$$

The notation $\propto$ means that the probability is proportional to the given expression. When the output space $\mathcal{Y}$ is finite, we have

$$\Pr[Y = y] = \frac{\exp\left(\frac{\varepsilon}{2\Delta} q(y;D)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\frac{\varepsilon}{2\Delta} q(y';D)\right)}$$

For people familiar with machine learning, this is a *softmax* over the scores with a "temperature" of $\frac{2\Delta}{\varepsilon}$. A high temperature (i.e., when ε is very small) means that the distribution is very uniform, and so the output leaks less information about the scores. A low temperature (i.e., when ε is very large) amplifies the differences between the scores and more closely approximates a true argmax.

We can also define the Exponential Mechanism for continuous (and infinite) output spaces $\mathcal{Y}$, as long as the normalization constant (either $\sum_{y' \in \mathcal{Y}} \exp(\ldots)$ or $\int_{y' \in \mathcal{Y}} \exp(\ldots)\, dy'$) is finite for all D.

Let's now see how private and useful the Exponential Mechanism is.

Privacy of the Exponential Mechanism.

Theorem 4. The Exponential Mechanism is ε-differentially private.

Proof. The result follows by showing that for any neighboring databases D and D',

$$\frac{\Pr[Y = y \mid D]}{\Pr[Y = y \mid D']} \leq \exp(\varepsilon) .$$

Focusing on the case of a discrete output space $\mathcal{Y}$, we have that for any $y \in \mathcal{Y}$ and any neighboring databases D and D',

$$\exp\left(\frac{\varepsilon}{2\Delta} q(y; D)\right) \leq \exp\left(\frac{\varepsilon}{2\Delta}(q(y; D') + \Delta)\right) = \exp\left(\frac{\varepsilon}{2}\right) \exp\left(\frac{\varepsilon}{2\Delta} q(y; D')\right) .$$

And thus,

$$\begin{aligned}
\frac{\Pr[Y = y \mid D]}{\Pr[Y = y \mid D']} &= \frac{\exp\left(\frac{\varepsilon}{2\Delta} q(y; D)\right)}{\exp\left(\frac{\varepsilon}{2\Delta} q(y; D')\right)} \cdot \frac{\sum_{y' \in \mathcal{Y}} \exp\left(\frac{\varepsilon}{2\Delta} q(y'; D')\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\frac{\varepsilon}{2\Delta} q(y'; D)\right)} \\
&\leq \frac{\exp\left(\frac{\varepsilon}{2}\right) \exp\left(\frac{\varepsilon}{2\Delta} q(y; D')\right)}{\exp\left(\frac{\varepsilon}{2\Delta} q(y; D')\right)} \cdot \frac{\sum_{y' \in \mathcal{Y}} \exp\left(\frac{\varepsilon}{2}\right) \exp\left(\frac{\varepsilon}{2\Delta} q(y'; D)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\frac{\varepsilon}{2\Delta} q(y'; D)\right)} \\
&= \exp\left(\frac{\varepsilon}{2}\right) \cdot \exp\left(\frac{\varepsilon}{2}\right) = \exp(\varepsilon) .
\end{aligned}$$

□

Utility of the Exponential Mechanism. Consider the highest score, denoted as

$$\mathrm{OPT}(D) := \max_{y \in \mathcal{Y}} q(y; D)$$

The Exponential Mechanism guarantees that the score of the selected element is close to the highest score with high probability:

Theorem 5. Suppose $\mathcal{Y}$ is finite and has size k. Then for every Δ-sensitive score function q, for every data set D, and every $\delta > 0$, the output of the Exponential Mechanism $M(D)$ satisfies:

$$\Pr_{y \leftarrow M(D)}\left[q(y; D) \leq \mathrm{OPT}(D) - \frac{2\Delta(\log k + \log(1/\delta))}{\varepsilon}\right] \leq \delta .$$

Coming back to our voting example, we see that with probability $1 - \delta$, the score of the selected class is at least $\mathrm{OPT} - \frac{\log k + \log(1/\delta))}{\varepsilon}$.

So instead of noise on the order of $\tilde{O}(\sqrt{k}/\varepsilon)$, we can get by with noise on the order of $\tilde{O}(\log k/\varepsilon)$, a significant improvement when k is large!

For example, if there are $k = 1{,}000$ classes, and we want $\varepsilon = 0.5$ privacy, then with probability at least 99%, the Exponential Mechanism selects a class with at least $\mathrm{OPT} - \frac{2}{0.5}(\log(1{,}000) + \log(100)) \approx \mathrm{OPT} - 46$ votes. In contrast, the Gaussian mechanism would require noise of standard deviation over $\frac{\sqrt{1{,}000}}{0.5} \approx 63$ applied to each vote count.

References

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In

Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25, pages 486–503. Springer, 2006.

[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.

[DRV10] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 51–60. IEEE, 2010.

[HR10] Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 61–70. IEEE, 2010.

[HT10] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 705–714, 2010.

[Mir17] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

[Ste22] Thomas Steinke. Composition of differential privacy & privacy amplification by subsampling. *arXiv preprint arXiv:2210.00597*, 2022.

[Vad17] Salil Vadhan. The complexity of differential privacy. *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pages 347–450, 2017.