Privacy in Statistics and Machine Learning — Spring 2021
Lecture 5: Differential Privacy II

Adam Smith and Jonathan Ullman

1 The Exponential Mechanism

The Laplace mechanism works well when the computation we want to carry out returns a vector to which we can add noise, and computation's global sensitivity is not too high.

What happens when when adding noise to the result makes no sense? The *exponential mechanism* is the natural starting point for designing differentialy private algorithms.

We'll motivate the mechanism with two problems, both of which have a "selection" flavor:

Example 1.1 (Approval voting)**.** Suppose we are trying to vote on a name for our course mascot, a terrier-husky mix who is currently just called the "Differential Privacy Doggie". We have many names to choose from—students proposed a total of d names. We've decided to conduct the election using *approval voting* (AV): in "AV", each voter can vote for as many candidates as they want; the candidate with the most votes wins. If there are d candidates, one can think of each voter's input as a subset $x_i \subseteq [d]$. The score of candidate j is the number of voters who included j in their subset, that is, $q(j; \mathbf{x}) = |\{i : j \in x_i\}|$. The highest-scoring candidate wins.

We wish to run the election differentially privately. We won't necessarily be able to get the exact winner, but maybe we can get a name with almost the maximum number of votes. One approach is to use the Laplace mechanism to release noisy versions of all the scores. But the global sensitivity of the whole list is d, and then we would add noise d/ε to each score. Can we obtain the name someone whose score is much closer than d/ε to the highest?

Example 1.2 (Prices of a digital good)**.** With a shiny new home microphone, you just recorded "My Corona", a parody of The Knack's 1979 hit song, "My Sharona". Now you want to sell it online. In a ma-ma-ma-market study, you talk to n people and find out the price $x_i \in [0, 1]$ each person would willing to pay for a download of the song. Assuming that respondents answered truthfully, a reasonable estimate for the revenue you would get from selling the download at price p is

$$q(p; \mathbf{x}) = p \cdot \#\{i : x_i \geq p\} \ .$$

You would like to use a differentially private algorithm to publish a price $\hat{p} \in \{\$0.01, \$0.02, \ldots \$1.99\}$ such that $q(\hat{p}; \mathbf{x})$ is as large as possible.

Adding noise to the best price might not make sense: For example, if everyone had the same maximum price $x_i = \$0.70$ for your song, the best price for you to charge would be \$0.70. Charging \$0.69 would also be ok (you would still make nearly as much as possible), but charging \$0.71 would result in no one buying your song.

1.1 Selection Problems and the Exponential Mechanism

These examples share a common structure. They are both special cases of a general *selection problem*, specified by:

- A set $\mathcal{Y}$ of possible outputs;

- A score function $q : \mathcal{Y} \times \mathcal{U}^n \to \mathbb{R}$ which measures the "goodness" of each output for a data set. Given $\mathbf{x} \in \mathcal{U}^n$, our goal is to find $y \in \mathcal{Y}$ which approximately maximizes $q(y; \mathbf{x})$. (When $\mathcal{Y}$ is finite, we can also think of q as a collection of $\mathcal{Y}$ separate low-sensitivity queries.)
- A sensitivity bound $\Delta > 0$ such that $q(y; \cdot)$ is Δ-sensitive for every y. That is,

$$\sup_{y \in \mathcal{Y}} \sup_{\substack{\mathbf{x}, \mathbf{x}' \in \mathcal{U}^n \\ \text{adjacent}}} |q(y; \mathbf{x}) - q(y; \mathbf{x}')| \leq \Delta \,. \tag{1}$$

The table below shows how these parameters work out for our two examples:

	Approval Voting	Pricing a Digital Good
Possible outputs $\mathcal{Y}$	Mascot names	$\{\$0.01, \$0.02, \ldots \$1.99\}$
Score $q(y; \mathbf{x}) = \ldots$	Number of votes for candidate y	$y \cdot \#\{i : x_i \geq y\}$
Maximum Sensitivity Δ	1	1.99

Given these elements, we get Algorithm 2. The idea is that given the score function $q(\cdot; \mathbf{x})$ that assigns a number to each element $y \in \mathcal{Y}$, we define a probability distribution which generates each element in y in $\mathcal{Y}$ with probability proportional to $\exp(\frac{\varepsilon}{2\Delta} q(y; \mathbf{x})$; that is, we sample elements with a probability that grows exponentially with their score. The symbol "$\propto$" in Algorithm 2 means "proportional to".

Algorithm 1: Exponential Mechanism $A_{EM}(\mathbf{x}, q(\cdot;\cdot), \Delta, \varepsilon)$

Input: Assume that $q(y; \cdot)$ is Δ-sensitive for every $y \in \mathcal{Y}$.

1 Select Y from the distribution with $\Pr(Y = y) \propto \exp\left(\frac{\varepsilon}{2\Delta} q(y; \mathbf{x})\right)$;

2 **return** Y;

When is this algorithm even well defined? When $\mathcal{Y}$ is finite the algorithm is well-defined since we can set

$$P(Y = y) = \frac{e^{\frac{\varepsilon}{2\Delta} q(y; \mathbf{x})}}{\sum_{y' \in \mathcal{Y}} e^{\frac{\varepsilon}{2\Delta} q(y'; \mathbf{x})}} \,. \tag{2}$$

In fact, the mechanism makes sense over infinite domains, and even continuous ones. For inifinite discrete domains like the integers $\mathbb{Z}$, it must be that $\sum_{y \in \mathcal{Y}} e^{\frac{\varepsilon}{2\Delta} q(y; \mathbf{x})}$ is finite for every $\mathbf{x}$. Over continuous spaces like the real line, it must be that $\int_{y \in \mathcal{Y}} \exp\left(\frac{\varepsilon}{2\Delta} q(y; \mathbf{x})\right) dy$ is finite for every possible data set $\mathbf{x}$. We will see an example further below.

Now that we have a well-defined algorithm, we'll try to understand why it is differentially private, and why it is useful.

Theorem 1.3. *If q is Δ-sensitive (i.e., satisfies (1)) then the exponential mechanism is ε-differentially private.*

Proof. Assume for simplicity that $\mathcal{Y}$ is finite. For any output y and data set $\mathbf{x}$ we have $P(y|\mathbf{x}) = \frac{e^{\frac{\varepsilon}{2\Delta} q(y; \mathbf{x})}}{\sum_{y' \in \mathcal{Y}} e^{\frac{\varepsilon}{2\Delta} q(y'; \mathbf{x})}}$. Let $\mathbf{x}'$ be a data set adjacent to $\mathbf{x}$. Since the sensitivity of $q(y; \cdot)$ is at most Δ, we have

$$\frac{e^{\frac{\varepsilon}{2\Delta} q(y; \mathbf{x})}}{e^{\frac{\varepsilon}{2\Delta} q(y; \mathbf{x}')}} = \exp\left(\frac{\varepsilon}{2\Delta}(q(y; \mathbf{x}) - q(y; \mathbf{x}'))\right) \leq \exp\left(\frac{\varepsilon}{2\Delta} \cdot \Delta\right) = e^{\varepsilon/2} \tag{3}$$

and similarly, for the normalizing constants,

$$\frac{\sum_{y' \in \mathcal{Y}} e^{\frac{\varepsilon}{2\Delta} q(y';\mathbf{x}')}}{\sum_{y' \in \mathcal{Y}} e^{\frac{\varepsilon}{2\Delta} q(y';\mathbf{x})}} \leq \sup_{y'} \left(\exp\left(\frac{\varepsilon}{2\Delta} (q(y';\mathbf{x}') - q(y';\mathbf{x})) \right) \right) \leq e^{\varepsilon/2} .$$

Thus the ratio $\frac{Pr(y|\mathbf{x})}{P(y|\mathbf{x}')}$ is at most $e^{\varepsilon/2} \cdot e^{\varepsilon/2} = e^{\varepsilon}$. The case of an infinite domain is similar, with integrals over to the base measure replacing sums. □

1.2 Utility of the Exponential Mechanism

We now have a very general tool in our toolbox, which can be used to design an agorithm for any problem where we can assign possible outputs a score according to their desirability. The algorithm is always differentially private.

The question is, when is this approach actually useful? Does it help us address approval voting and price selection, the two examples problems we started out with?

Just how useful the exponential mechanism is depends a lot on the exact problem structure. But we can write down a few clean and generally useful bounds. The best we can hope for from a selection algorithm is that, on input a data set $\mathbf{x}$, it outputs an element $y \in \mathcal{Y}$ with the maximum possible score, denoted

$$q_{\max}(\mathbf{x}) \stackrel{\text{def}}{=} \max_{y \in \mathcal{Y}} q(y;\mathbf{x}) \tag{4}$$

We'll show that we can get an element with near-maximum score, with high probability.

Proposition 1.4. *Suppose $\mathcal{Y}$ is finite and has size d. Then for every Δ-sensitive score function q, for every data set $\mathbf{x}$, and every $t > 0$, the output of the exponential mechanism $Y \leftarrow A_{EM}(\mathbf{x}, q, \Delta, \varepsilon)$ satisfies:*

$$\mathop{\mathbb{P}}_{Y \leftarrow A_{EM}(\mathbf{x},q,\Delta,\varepsilon)} \left(q(Y;\mathbf{x}) < q_{\max}(\mathbf{x}) - \frac{2\Delta(\ln(d) + t)}{\varepsilon} \right) \leq e^{-t}, \quad \text{where } q_{\max}(\mathbf{x}) = \max_{y=1}^{d} q(y;\mathbf{x}) \tag{5}$$

In particular, we have

$$\mathop{\mathbb{E}}_{Y \leftarrow A_{EM}(\mathbf{x},q,\Delta,\varepsilon)} (q(Y;\mathbf{x})) \geq q_{\max}(\mathbf{x}) - \frac{2\Delta(\ln(d) + 1)}{\varepsilon} . \tag{6}$$

Proof. Fix a data set $\mathbf{x}$ and a score function q. To make the proof more readable, we'll drop the $\mathbf{x}$ symbol in the score function, writing $q(y)$ and $q_{\max}$ instead of $q(y;\mathbf{x})$ and $q_{\max}(\mathbf{x})$.

We can divide the possible outputs into sets G_t and B_t of "good" and "bad" outputs, where

$$G_t = \left\{ y \in \mathcal{Y} \ : q(y) > q_{\max} - \tfrac{2\Delta}{\varepsilon}(\ln(d) + t) \right\} \quad \text{and} \quad B_t = \left\{ y \in \mathcal{Y} \ : q(y) \leq q_{\max} - \tfrac{2\Delta}{\varepsilon}(\ln(d) + t) \right\}$$

To prove the first part of the Proposition, we need to show that $\mathbb{P}(B_t) \leq e^{-t}$. Let's write the probability of an element y as $\mathbb{P}(Y = y) = Ce^{\frac{\varepsilon}{2\Delta} q(y)}$, where C is the normalizing constant $C = \sum_{y \in \mathcal{Y}} e^{\frac{\varepsilon}{2\Delta} q(y)}$.

Let y^* be an output with score $q_{\max}$. We can bound $\mathbb{P}(B_t)$ as

$$\mathbb{P}(B_t) < \frac{\mathbb{P}(B_t)}{\mathbb{P}(y^*)} = \frac{\sum_{y \in B_t} \mathbb{P}(Y = y)}{\mathbb{P}(Y = y^*)} = \frac{\sum_{y \in B_t} \exp\left(\frac{\varepsilon}{2\Delta} q(y)\right)}{\exp\left(\frac{\varepsilon}{2\Delta} q_{\max}\right)} \tag{7}$$

Since the bad y's satisfy $q(y) \leq q_{\max} - \frac{2\Delta}{\varepsilon}(\ln(d)+t)$, the sum in the numerator is at most $|B_t| \exp(\frac{\varepsilon}{2\Delta} q_{\max} - (\ln(d)+t))$ and we get that

$$\mathbb{P}(B_t) < \frac{|B_t| \exp(\frac{\varepsilon}{2\Delta} q_{\max} - (\ln(d)+t))}{\exp\left(\frac{\varepsilon}{2\Delta} q_{\max}\right)} = |B_t| \cdot e^{-\ln(d)-t} \leq |B_t| \cdot \frac{1}{d} \cdot e^{-t}. \tag{8}$$

Since B_t contains at most $d-1$ elements, we get the desired bound on $\mathbb{P}(B_t)$.

The last part follows from the fact that for any nonnegative random variable Z, we have $\mathbb{E}(Z) = \int_{z \geq 0} \mathbb{P}(Z > z) dz$. Let's apply this to the random variable $Z = \frac{\varepsilon}{2\Delta}(q_{\max} - q(Y))$. The probability that it exceeds $z = \ln(d) + t$ is at most e^{-t} for $t > 0$, and at most 1 for for $t \leq 0$. So we get

$$\mathbb{E}(Z) = \int_{z=0}^{\infty} \mathbb{P}(Z > z) dz = \int_{t=-\ln(d)}^{\infty} \mathbb{P}(Z > \ln(d) + t) dt \leq \int_{t=-\ln(d)}^{0} 1 dy + \int_{t=0}^{\infty} e^{-t} dt = \ln(d) + 1.$$

□

Example 1.5 (Approval Voting, continued)**.** Let's apply our new Proposition to the approval voting example. The scores there are counts, and have sensitivity 1. Let $q_{\max}$ be the score of the most popular candidate, and suppose $d = 100$—a reasonable number of candidate names for the mascot—and $\varepsilon = 0.5$. Proposition 1.4 shows that with probability at least 0.99, we'll get a candidate whose score is at most $q_{\max} - \frac{2 \cdot 1}{0.5}(\ln(100) + \ln(1/0.01)) \approx q_{\max} - 36.8$. If the best candidate won by 39 or more votes, we would get their name with high probability. Compare this with the Laplace mechanism, where scores would be perturbed by about $\frac{d}{\varepsilon} = 200$, and the largest perturbation might be far bigger.

Example 1.6 (Pricing a digital good, continued)**.** Let's return to the problem of setting a price for "My Corona". There are d =200 possible prices, so we can apply Proposition 1.4 to show that we can get a price that leads to revenue within about $2\Delta(\ln(d)+1)/\varepsilon = 2 \cdot 1.999 \cdot \ln(200)/\varepsilon \approx \frac{31}{\varepsilon}$ of the best possible.

In fact, we can get a better bound for this problem. The key idea is that if a price p is good, then the prices slightly less than p are also pretty good. We won't work out the details here, but we will see exercises which use the idea.

1.3 More Examples of the Exponential Mechanism

The Laplace Mechanism and Randomized Reponse can also be seen—almost—as special cases of the exponential mechanism:

	Laplace Mechanism	Randomized Response
Possible outputs $\mathcal{Y}$	$\mathbb{R}^d$	$\{0,1\}^n$
Score $q(y;\mathbf{x}) = \ldots$	$\|y - f(\mathbf{x})\|_1$	$\#agree(y,\mathbf{x})$
Maximum Sensitivity Δ	GS_f	1

If you plug the score function $q(y;\mathbf{x}) = \|y - f(\mathbf{x})\|_1$ directly into the exponential mechanism, you will sample from the distribution with probability proportional to $\exp\left(-\frac{\varepsilon}{2}\|y - f(\mathbf{x})\|_1\right)$. This is definitely dfferentially private, but it isn't quite the Laplace mechanism. The actual Laplace mechanism samples y with probability proportional to $\exp(-\varepsilon\|y - f(\mathbf{x})\|_1)$, effectively saving a factor of 2 in the exponent. Something similar happens with randomized reponse. This occurs because the normalization constants $C_{\mathbf{x}}$ by which one divides to get probability distributions are actually independent of $\mathbf{x}$. The general exponential mechanism must allow for a varying normalization constant, which is where the extra factor of two comes from.

2 Report Noisy Max

When the domain is finite, it is often more convenient to work with a another algorithm which behaves very similarly to the exponential mechanism. The setup is the same—we have a set of outcomes $\mathcal{Y}$ (now required to be finite) and a score function with sensitivty at most δ for each outcome. The idea is to add noise with expected magnitude Δ/ε to each item's score, independent of the number of possible outputs. The algorithm returns the *output with the highest noisy score*:

Algorithm 2: Report-Noisy-Max $A_{RNM}(\mathbf{x}, q(\cdot;\cdot), \Delta, \varepsilon)$

Input: Assume that $q(y;\cdot)$ is Δ-sensitive for every $y \in \mathcal{Y}$, and $\mathcal{Y} = \{1, ..., d\}$ is finite

Select $Z_1, ..., Z_d \sim \text{Exp}(2\Delta/\varepsilon)$ i.i.d. ;

return $\arg\max_{y\in\{1,...,d\}} \left(q(y;\mathbf{x}) + Z_y\right)$;

The distribution being used to generate noise is the *exponential distribution* $\text{Exp}(\lambda)$, a distribution over the nonegative real numbers $[0, +\infty)$ with density $h_\lambda(y) = \frac{1}{\lambda}\exp(-y/\lambda)$.

This algorithm is generally much easier to implement than the exponential mechanism, since it does not require explicitly computing any probabilities and can make use of standard libraries for sampling from the exponential distribution. It satisfies a very similar guarantee to the exponential mechanism.

Exercise 2.1. Show that report noisy max is ε-differentially private. [*Hint:* Suppose you fix (condition on) the values of the noise $Z_u = z_u$ for all $u \neq y$. Compute and compare the probability that the outcome will be y under two different data sets $\mathbf{x}$ and $\mathbf{x}'$.]

In addition to making implementation easier, the utility analysis of report-noisy-max is more intuitive. We just need to bound the probability that all d noise random variables are small:

Lemma 2.2 (Tail Bounds for Exponential Distributions)**.**

1. *If* $Z \sim \text{Exp}(\lambda)$, *then* $\Pr(Z \geq t\lambda) = e^{-t}$ *for all* $t \geq 0$.
2. *If* $Z_1, ..., Z_d \sim \text{Exp}(\lambda)$ *i.i.d., and* $Z_{\max} = \max_{i=1}^d Z_i$ *then* $\Pr(Z_{\max} > \lambda(\ln(d) + t)) = e^{-t}$ *for all* $t \geq 0$, *and* $\mathbb{E}(Z_{\max}) \leq \lambda(\ln(d) + 1)$.

Note that if Y_i are independent Laplace random variables with $Y_i \sim \text{Lap}(\mu_i, \lambda_i)$ and $Z_i = |Y_i - \mu_i|$, then the Z_i's will be exponentially distributed with parameter λ and so Lemma 2.2 above applies.

Proof. The first part follows from a direct computation of the CDF:

$$Pr(Z > \lambda t) = \int_{y\geq\lambda t} \tfrac{1}{\lambda} e^{-y/\lambda} dy = \tfrac{1}{\lambda}\left[-\lambda e^{-y/\lambda}\right]_{y=\lambda t}^{\infty} = e^{-t}.$$

The second part follows by a union bound: the probability that any particular Z_i exceeds $\lambda(\ln(d) + t)$ is $\frac{e^{-t}}{d}$ by part 1, so the probability that any of the Z_i's exceeds the bound is at most e^{-t}. The expectation calculation is essentially the same as in the proof the exponential mechanism's utility (Proposition 1.4). □

We can also use Lemma 2.2 to prove the following, which is essentially identical to what we proved about the exponential mechanism.

Theorem 2.3. *If* $q(y;\cdot)$ *is* Δ*-sensitive for every* $y \in \{1, ..., d\}$, *then for every data set* $\mathbf{x}$ *in* $\mathcal{U}^n$ *and every* $t > 0$, *the output of report-noisy-max* $Y \leftarrow A_{RNM}(\mathbf{x}, score, \Delta, \varepsilon)$ *satisfies*

$$\Pr\left(q_{max}(\mathbf{x}) - q(Y, \mathbf{x}) \geq \frac{2\Delta(\ln(d) + t)}{\varepsilon}\right) \leq e^{-t}, \text{ where } q_{max}(\mathbf{x}) = \max_{y=1}^{d} q(y;\mathbf{x}),$$

and

$$\mathbb{E}\left(q_{max}(\mathbf{x}) - q(Y, \mathbf{x})\right) \leq \frac{2\Delta(\ln(d)+1)}{\varepsilon}.$$

Exercise 2.4. Prove Theorem 2.3 using the bounds in Lemma 2.2.

2.1 RNM, The Exponential Mechanism, and Gumbel Noise

Report Noisy Max with Laplace Noise has essentially the same guarantees as the exponential mechanism (on the same discrete domain), but performs better in practice. It turns out that the expoential mechanism is *exactly equivalent* to RNM with noise added from a different distribution, the Gumbel distribution with parameter $\beta = \frac{2\Delta}{\varepsilon}$.

$$\text{Gumbel}(\beta): \qquad \text{PDF } h_\beta(y) = \frac{1}{\beta}\exp\left(-\frac{x}{\beta} - e^{x/\beta}\right), \qquad \text{CDF } \mathop{\mathbb{P}}_{Y\sim\text{Gumbel}(\beta)}(Y \leq y) = e^{-e^{-x/\beta}}. \tag{9}$$

This equivalence is known in the machine learnng literature as the "Gumbel max trick", where the exponential mechanism is called the Gibbs or softmax distribution. It turns out that RNM with Laplace noise is equivalent to a different process, "Permute and Flip" [MS20].

Exercise 2.5 (Gumbel Max Trick)**.** Show that RNM with Gumbel noise with parameter $\beta = \frac{2\Delta}{\varepsilon}$ generates exactly the same distribution as the exponential mechanism.

Additional Reading

- McSherry and Talwar's paper that defined the exponential mechanism [MT07]
- The "Permute-and-Flip" mechanism [MS20] is an equivalent algorithm to Report-Noisy-Max [Ste20]. McKenna and Sheldon [MS20] argue that the algorithm is optimal among a natural class of selection algorithms.
- Further optimizations on the exact privacy cost of the exponential mechanism can be found in [DDR20, DWX+20].

References

[DDR20] Jinshuo Dong, David Durfee, and Ryan Rogers. Optimal differential privacy composition for exponential mechanisms. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 2020.

[DWX+20] Zeyu Ding, Yuxin Wang, Yingtai Xiao, Guanhong Wang, Danfeng Zhang, and Daniel Kifer. Free gap estimates from the exponential mechanism, sparse vector, noisy max and related algorithms. *arxiv [CoRR]*, abs/2012.01592, 2020.

[MS20] Ryan McKenna and Daniel R. Sheldon. Permute-and-flip: A new mechanism for differentially private selection. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *IEEE Symposium on Foundations of Computer Science*, FOCS '07, 2007.

[Ste20] Thomas Steinke. Personal communication, November 2020.