Algorithms for Private Data Analysis — Spring 2021
Lecture 6: Solving Selection Problems Privately

Steven Wu

1 The Exponential Mechanism

The Laplace mechanism works well when the computation we want to carry out returns a vector of numeric values to which we can add noise, and the vector of interest has low global sensitivity.

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 (Heavy hitter)**.** Suppose we are trying to find out which website is the most popular among a set of users. If there are d websites, one can think of each user's input as a subset $x_i \subseteq [d]$. The score of website j is the number of users who included j in their subset, that is, $q(j; \mathbf{x}) = |\{i : j \in x_i\}|$. The winner has the highest score.

We wish to find a "heavy hitter" subject to differential privacy. We won't necessarily be able to get the exact top winner, but maybe we can identify a webiste with almost the maximum number of users. 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)**.** Suppose you made an iPhone app. Now you want to sell it online. In a survey, 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 app. 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 app, 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 app.

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 $q : \mathcal{Y} \times \mathcal{X}^n \to \mathbb{R}$ which measures the "goodness" of each output for a data set. Given $\mathbf{x} \in \mathcal{X}^n$, 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,

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

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

	Heavy Hitter	Pricing a Digital Good
Possible outputs $\mathcal{Y}$	Websites	$\{\$0.01, \$0.02, \ldots \$1.99\}$
Score $q(y;\mathbf{x}) = \ldots$	Number of users visiting y	$y \cdot \#\{i : x_i \ge y\}$
Maximum Sensitivity Δ	1	1.99

Given these elements, we get Algorithm 1. 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 1 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 form 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

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

and similarly, for the normalizing constants,

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

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 heavy hitter 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

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

□

Example 1.5 (Heavy Hitter, continued)**.** Let's apply our new Proposition to the heavy hitter example. The scores there are counts, and have sensitivity 1. Let $q_{\max}$ be the score of the most popular website, and suppose $d = 100$—a reasonable number of website—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 an iPhone app. 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.

Acknowledge This lecture note is built on the note written by Adam Smith and Jonathan Ullman.

References