CS 860: Algorithms for Private Data Analysis Fall 2020

Lecture 7 — Exponential Mechanism

Prof. Gautam Kamath *Scribe: Gautam Kamath*

The main algorithms for differential privacy we have seen so far, the Laplace mechanism and the Gaussian mechanism, have both focused on privatization of numerical queries. That is, we wish to output a value. However, in many natural situations, we might want to output an *object*, which has *quality* comparable to the best possible. For this purpose, we will introduce the *exponential mechanism* of McSherry and Talwar [MT07].

Presentation in these notes is based heavily off of Sections 3.4 and 10.2.1 of [DR14], and Section 8.1 of [Vad17].

Exponential Mechanism

To illustrate the type of setting the exponential mechanism tries to address, consider a *digital goods* auction, with unit demand buyers. In English: a retailer has unlimited copies of some item (i.e., digital copies of a book, movie, video game, etc.). There are n individuals who are all interested in buying one copy of this item (they aren't interested in duplicates), and individual i would pay at most their valuation v_i. How should the retailer price the item? They could simply look at the valuations and choose the price p which maximizes the revenue $\sum_{i:p\leq v_i} p$, but this is rather non-private. Indeed, if someone is either very rich or has a very high valuation for the item, the retailer's best option might be to choose a high value for p, thus revealing the presence of said individual in the dataset.

Naturally, we turn to differential privacy as a solution concept. But it turns out the revenue function may be rather sensitive to changes in the price. To illustrate this, suppose a retailer is selling a game, and there are 3 individuals who have the following valuations for the game: \$1, \$1, \$3.01. For a price p of \$1, the revenue would be \$3. But increasing it marginally, to \$1.01, the revenue drops to \$1.01. At a price of \$3.01, the revenue finally overtakes that at a price of \$1, reaching \$3.01. But go any further, to \$3.02, and the revenue drops to \$0! This is all to show that naive attempts to privatize the price by adding noise to it seem to be ineffective. This will be avoided by thinking of the prices as "objects" instead of "values." Prices of \$1 or \$3.01 are "high quality objects" (producing the best revenues) whereas prices of \$1.01 or \$3.02 are low quality.

More formally, the exponential mechanism takes in the following input:

- A dataset $X \in \mathcal{X}^n$,
- A set of objects $\mathcal{H}$,
- A score function $s : \mathcal{X}^n \times \mathcal{H} \to \mathbb{R}$.

The idea is that the score function takes in a dataset X and an object $h \in \mathcal{H}$, and outputs how "good" h is with respect to X. In the setting of the example above, the dataset would be the individuals' valuations, the set of objects would be all possible prices, and the score function would be the revenue obtained from that set of individuals at a given price.

It is important to be clear on what is and what is not private/sensitive in this setting. We assume the set of objects and the score function are publicly known, and there is no need to preserve their privacy. The only private information is the datset X. With this in mind, we define the sensitivity of the score function with respect to the dataset only:

$$\Delta s = \max_{h \in \mathcal{H}} \max_{X,X'} |s(X,h) - s(X',h)|,$$

where X and X' are neighbouring datasets. When s is clear from context, we will simply use the symbol Δ.

With this notation defined, the exponential mechanism can be defined very succinctly.

Definition 1. *The exponential mechanism M_E on inputs X, $\mathcal{H}$, s, selects and outputs some object $h \in \mathcal{H}$, where the probability a particular h is selected is proportional to* $\exp\left(\frac{\varepsilon s(X,h)}{2\Delta}\right)$.

While it is easy to state, this doesn't really help much when it comes to computation. Indeed, in general it requires one to compute this quantity for each $h \in \mathcal{H}$, in order to define the distribution, compute the normalization factor, and then sample from the distribution. This may be costly if $\mathcal{H}$ is very large, or potentially impossible if it is infinite (though we will see how discretization of the space may allow us to avoid this). Nonetheless, it often serves well as a baseline for many tasks, information theoretically proving that a certain error or sample complexity is achievable, leading to the question of efficient algorithms.

Privacy

Let's start by proving privacy of the exponential mechanism.

Theorem 2. *The exponential mechanism M_E is ε-differentially private.*

Proof. Fix X, X' as neighbouring datasets, and some outcome $h \in \mathcal{H}$. The we express the ratio of the probability of h being output under X and X' as follows:

$$\begin{aligned}
\frac{\Pr[M_E(X) = h]}{\Pr[M_E(X') = h]} &= \frac{\left(\frac{\exp\left(\frac{\varepsilon s(X,h)}{2\Delta}\right)}{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon s(X,h')}{2\Delta}\right)}\right)}{\left(\frac{\exp\left(\frac{\varepsilon s(X',h)}{2\Delta}\right)}{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon s(X',h')}{2\Delta}\right)}\right)} \\
&= \exp\left(\frac{\varepsilon(s(X,h) - s(X',h))}{2\Delta}\right)\left(\frac{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon s(X',h')}{2\Delta}\right)}{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon s(X,h')}{2\Delta}\right)}\right) \\
&\le \exp\left(\frac{\varepsilon}{2}\right)\exp\left(\frac{\varepsilon}{2}\right)\left(\frac{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon s(X,h')}{2\Delta}\right)}{\sum_{h' \in \mathcal{H}} \exp\left(\frac{\varepsilon s(X,h')}{2\Delta}\right)}\right) \\
&= \exp(\varepsilon).
\end{aligned}$$

The inequality holds based on the definition of Δ, applied once to each term. While the first application is immediate, the second is the following inequality applied to each term in the summation in the numerator:

$$\exp\left(\frac{\varepsilon s(X', h')}{2\Delta}\right) \leq \exp\left(\frac{\varepsilon}{2}\right)\exp\left(\frac{\varepsilon s(X, h')}{2\Delta}\right).$$

□

Utility

The exponential mechanism allows us to privately select an object with a score comparable to the best – in particular, we will lose a small additive amount, which depends on the sensitivity, ε, and the number of candidate objects. Its utility is described by the following theorem.

Theorem 3. *Let X be a dataset, and $OPT(X) = \max_{h\in\mathcal{H}} s(X, h)$ be the score attained by the best object h with respect to the dataset X. For a dataset X, let $\mathcal{H}^* = \{h \in \mathcal{H} : s(X, h) = OPT(X)\}$ be the set of objects which achieve this score. Then*

$$\Pr\left[s(M_E(X)) \leq OPT(X) - \frac{2\Delta}{\varepsilon}\left(\ln\left(\frac{|\mathcal{H}|}{|\mathcal{H}^*|}\right) + t\right)\right] \leq \exp(-t).$$

This leads to the following corollary for the same setting, since $|\mathcal{H}^*| \geq 1$.

Corollary 4.

$$\Pr\left[s(M_E(X)) \leq OPT(X) - \frac{2\Delta}{\varepsilon}\left(\ln\left(|\mathcal{H}|\right) + t\right)\right] \leq \exp(-t).$$

We proceed with the proof of Theorem 3.

Proof.

$$\begin{aligned}\Pr[s(M_E(X)) \leq c] &= \frac{\sum_{h:s(X,h)\leq c\wedge h\in\mathcal{H}} \exp\left(\frac{\varepsilon s(X,h)}{2\Delta}\right)}{\sum_{h'\in\mathcal{H}} \exp\left(\frac{\varepsilon s(X,h')}{2\Delta}\right)} \\ &\leq \frac{|\mathcal{H}| \exp(\varepsilon c/2\Delta)}{|\mathcal{H}^*| \exp(\varepsilon OPT(X)/2\Delta)} \\ &= \frac{|\mathcal{H}|}{|\mathcal{H}^*|} \exp\left(\frac{\varepsilon(c - OPT(X))}{2\Delta}\right).\end{aligned}$$

From this inequality, the theorem statement can be obtained by substituting in the prescribed value for c.

It remains to explain the first inequality. The numerator can be upper bounded since there are at most $|\mathcal{H}|$ terms in the summation, and the condition on $s(X, h)$ bounds each by $\exp\left(\frac{\varepsilon c}{2\Delta}\right)$. Similarly, the denominator can be lower bounded by considering only the $|\mathcal{H}^*|$ terms with value $\exp\left(\frac{\varepsilon OPT(X)}{2\Delta}\right)$. □

Applications

The exponential mechanism is very versatile, and we will see three different applications of it.

Laplace Mechanism

It's not hard to see that the Laplace mechanism is an instantiation of the exponential mechanism. Suppose we are trying to privately compute a sensitivity-Δ statistic $f : \mathcal{X}^n \to \mathbb{R}$ on a dataset X using the Laplace mechanism. The dataset is X, the set of objects $\mathcal{H}$ is the real line $\mathbb{R}$, and the score function $s(X, h)$ is $-|f(X) - h|$. This will result in the probability of a point $h \in \mathbb{R}$ being output being proportional to $\exp\left(-\frac{\varepsilon|f(X)-h|}{2\Delta}\right)$. This is precisely the density of the Laplace distribution for the Laplace mechanism, up to a factor of 2. As the direct analysis shows can be removed with more care, but it's essentially due to the normalization factor not changing on neighbouring datasets.

Selling One Digital Good

Let's revisit the digital goods auction mentioned before. The seller has an unlimited supply of some item. There are n individuals, each with value $v_i \in [0, 1]$ for the item. The seller needs to choose a price $p \in [0, 1]$: if $p \leq v_i$, individual i will buy a copy of the item and the seller receives revenue p from them, else they don't buy the item and the seller receives revenue 0. Thus, the overall revenue is $p|\{i : p \leq v_i\}|$. The seller is constrained to choose the price in a differentially private way. In addition to privacy concerns, this also has implications for the "truthfulness" of the auction – basically saying that the buyer has little incentive to lie when reporting their value to the seller. Intuitively, this is because differential privacy ensures their valuation has little effect on the final price, but see Section 10.1 of [DR14] for more details.

The first thing we have to do is discretize the domain. For the Laplace Mechanism example this wasn't necessary, since we were lucky that the score function was very nice, but here it might not be so conveniently. Since $p \in [0, 1]$, we let $\mathcal{H} = \{\alpha, 2\alpha, \ldots, 1\}$, for some parameter α to be set later. Observe that $|\mathcal{H}| = 1/\alpha$.

How much do we lose by this discretization? Letting $\gamma = \max_p p|\{i : p \leq v_i\}|$, we have that $OPT(v) \geq \gamma - \alpha n$. To see this, observe that we if we "round down" p to the next multiple of α, the price changes by at most α, corresponding to revenue lost from at most n individuals. Note that no one will go from buying to not buying if the price is reduced.

With this discretization, the rest of the instantiation is easy. We naturally choose the score of a price p to be the revenue achieved at that price: $p|\{i : p \leq v_i\}|$. Since $p \leq 1$, and changing one individual can only affect $|\{i : p \leq v_i\}|$ by 1, the sensitivity $\Delta \leq 1$.

Simply applying Corollary 4 gives that the revenue will be at least $\gamma - \alpha n - \frac{\log(1/\alpha)}{\varepsilon}$ with high probability. Choosing $\alpha = \frac{\log n}{n\varepsilon}$ results in an overall revenue of $\gamma - \frac{\log n}{\varepsilon}$ – a fairly small loss in revenue compared to the best possible.

Private PAC Learning

We will now describe the setting of PAC learning [Val84], introduced by Valiant. We describe it non-privately first, and for now, we focus on Boolean functions over the Boolean hypercube, and the case of "proper learning." An algorithm is given a concept class $\mathcal{C} : \{c : \{0,1\}^d \to \{0,1\}\}$, which consists of a set of functions which map datapoints to labels. It is also given a dataset of n elements, consisting of points $(X_i, Y_i = c^*(X_i)) \in \{0,1\}^d \times \{0,1\}$, where c^* is some (unknown) function from $\mathcal{C}$ and the X_i's are drawn according to some (unknown) distribution D. The goal is for the algorithm to output a function $\hat{c} \in \mathcal{C}$ such that $\Pr_{x\sim D}[\hat{c}(x) \neq c^*(x)]$ is minimized. That is to say: if we drew another random point the distribution, we want the classifier we have learned to be likely to predict it correctly.

The following is a classic result in (non-private) learning theory.

Lemma 5. *If $n \geq \Omega(\log|\mathcal{C}|/\alpha^2)$, then any function in the concept class which correctly classifies the entire training dataset will have $\Pr_{x\sim D}[\hat{c}(x) \neq c^*(x)] \leq \alpha$.*

Proof. Consider some fixed function $h \in \mathcal{C}$. If we have $n \geq \Omega(t/\alpha^2)$, then a Chernoff bound gives

$$\Pr_{X_1,\ldots,X_n\sim D}\left[\left|\Pr_{x\sim D}[h(x) = c^*(x)] - \frac{|\{i : h(X_i) = c^*(X_i)\}|}{n}\right| \geq \alpha\right] \leq \exp(-t).$$

This means that, for any hypothesis in the class, the fraction of the input data it correctly classifies is within an additive α of the probability it would classify a new point correctly. Setting $t \gg \log|\mathcal{C}|$ implies that this holds simultaneously for all $h \in \mathcal{C}$ simultaneously with high probability. We know the algorithm will return something, since the true function c^* would never misclassify a point. Furthermore, the bound above implies that any function which doesn't misclassify a point has error at most α, completing the proof. □

Let's focus now on the differentially private setting. The neighbouring relationship between two datasets involves the change of a single point (X_i, Y_i) arbitrarily. Note that this is a worst-case modification: the new point (X_i', Y_i') does not have to be from the distribution D, and it is not required that $Y_i' = c(X_i')$.

Theorem 6. *If $n \geq \Omega(\log|\mathcal{C}|/\alpha^2 + \log|\mathcal{C}|/\alpha\varepsilon)$, then there exists an ε-differentially private algorithm which outputs a function $\hat{c}$ such that $\Pr_{x\sim D}[\hat{c}(x) \neq c^*(x)] \leq \alpha$.*

Unsurprisingly, to prove this theorem, we use the exponential mechanism. We let the set of objects be the concept class, i.e., $\mathcal{H} = \mathcal{C}$. The score function is the *negative* of the fraction of mis-classified points: $s((X,Y),h) = -|\{i : h(X_i) \neq Y_i\}|/n$. This makes sense: we're trying to output an object with the *largest* possible score, which in this case would be 0 for a function that makes no mistakes. It's easy to see that $\Delta = 1/n$ for this function. The correct function c^* has $s((X,Y),c^*) = 0$, so $OPT = 0$.

By the guarantees of the exponential mechanism, we will output a function $\hat{c}$ such that $s((X,Y),\hat{c}) \geq -\frac{2\ln|\mathcal{C}|}{\varepsilon n} \geq -\alpha/2$, and thus $|\{i : h(X_i) \neq Y_i\}|/n \leq \alpha/2$. Using the accuracy proof from the non-private case above, we can see that this is within an additive $\alpha/2$ of $\Pr_{x\sim D}[\hat{c}(x) \neq c^*(x)]$. Putting these together completes the proof.

References

[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.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society.

[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.

[Val84] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.