The Exponential Mechanism for Medians

April 28, 2020

1 The Exponential Mechanism

Sometimes, the global sensitivity of a function is too great, so the Laplace mechanism will not produce meaningful results. The median is one such function. In many cases, the *Exponential mechanism* is an alternate approach that gives reasonable utility.[1] Introduced in 2007 by McSherry and Talwar, the exponential mechanism posits that for a given database, users prefer some outputs over others. That those preferences may be encapsulated with a utility score, where a high utility score indicates a higher preference for that output. The exponential mechanism releases outputs with probability proportional (in the exponent) to the utility score and the sensitivity of the utility function.

Definition 1. *Let $\mathcal{X}$ be a space of databases and let $[m, M]$ be an arbitrary range. Let $u : \mathcal{X} \times [m, M] \to \mathbb{R}$ be a utility function, which maps pairs of databases and outputs to a utility score. Let Δu be the sensitivity of u with respect to the database argument. The exponential mechanism outputs $r \in [m, M]$ with probability proportional to $\exp\left(\frac{\varepsilon u(x,r)}{2\Delta u}\right)$ [MT07, DR+14].*[2]

Theorem 1. *The exponential mechanism preserves $(\varepsilon, 0)$-differential privacy [MT07, DR+14].*[3]

Note that the exponential mechanism may not be tractable in many cases, as it assumes the existence of a utility function, and even if one exists it may not be efficiently computable.

[1] This is not the *only* advantage of the exponential mechanism. It is a way to compute differentially private queries on non-numeric data, unlike the Laplace mechanism it does not assume that the probability of outputting a response ought to be symmetric about the true response, etc.

[2] The original definition is from [MT07], but here we state the version rewritten in [DR+14] as it is slightly clearer.

[3] As written in [MT07], the mechanism actually preserves $(2\varepsilon\Delta u, 0)$-differential privacy; the main difference in the [DR+14] version is that it has the extra factor of $2\Delta u$ to avoid these extra terms.

2 An Exponential Mechanism for Quantiles

2.1 Defining a sensible utility function

First, consider the case where the desired quantile is the median. Note that a user will prefer an output that is closer to the true median over one that is further away. Let x be an (ordered) data set, and let r be a possible median output by our mechanism. Note that if r is exactly the median, there should be the same number of points in x to the left and to the right of r. As r decreases or increases, then the distance between the number of points to the right of r and to the left of r will increase. So, the distance between the number of points to the left and the number of points to the right of r encapsulates how close the output is to the true median.

Slightly more formally, let $\#(x < r)$ denote the number of points to the left of r in database x and let $\#(x < r)$ denote the number of points to the right of r in database x. Define the utility function u as

$$u(x, r) \;\; \left|\#(x < R) - \#(x > R)\right|. \tag{2.1}$$

To generalize this to an arbitrary quantile, let N be the size of x and let $\alpha \in (0, 1)$ indicate the desired quantile. Then, we can modify our initial utility function as follows:

$$u(x, r) = \max(\alpha, (1 - \alpha))N - \left|(1 - \alpha)\#(x < r) - \alpha\#(x > r)\right|. \tag{2.2}$$

Say for example that $\alpha = 0.25$. Then, if r is exactly at the first quantile, $\#(x < r)$ will be the count of one fourth of the data and $\#(x > r)$ will be three fourths of the data;

2.2 Sensitivity of the utility function

2.2.1 Neighboring Definition: Change One

Theorem 2. *Let u be defined as in Eq. 2.2. The ℓ_1-sensitivity of u in the change-one model is bounded above by 1.*

Proof. Say X and X' are neighboring databases which differ at some data point. Let Δu denote the ℓ_1-sensitivity of $u(\cdot, \cdot)$ with respect to the space of databases.
Let $C_1 = \#(Z < x), C_2 = \#(Z > x)$. Worst case, C_1 increases by 1 and C_2 decreases by 1. Then

$$\begin{aligned}
\Delta u &= \left||(1 - \alpha)(C_1 + 1) - \alpha(C_2 - 1)| - |(1 - \alpha)C_1 - \alpha C_2|\right| \\
&\leq |(1 - \alpha)(C_1 + 1) - \alpha(C_2 - 1) - (1 - \alpha)C_1 + \alpha C_2| \\
&\leq |C_1 + 1 - \alpha C_1 - \alpha - \alpha C_2 + \alpha - C_1 + \alpha C_2 + \alpha C_2| \\
&= 1
\end{aligned}$$

If instead C_2 increases by 1 and C_1 decreases by 1, the result is identical (negative sign falls out).

□

2.2.2 Neighboring Definition: Add/Drop One

Theorem 3. *Let u be defined as in Eq. 2.2. The ℓ_1-sensitivity of u in the add/drop one model is bounded above by* $\max(\alpha, 1 - \alpha)$..

Proof. If we add 1, there are two worse-cases: C_2 increases by 1, nothing happens to C_2.

$$\begin{aligned}\Delta u &= |(1-\alpha)(C_1+1) - \alpha(C_2)| - |(1-\alpha)C_1 - \alpha C_2| \\ &\leq |(1-\alpha)(C_1+1) - \alpha(C_2) - (1-\alpha)C_1 + \alpha C_2| \\ &= |C_1 + 1 - \alpha C_1 - \alpha - \alpha C_2 - C_1 + \alpha C_1 + \alpha C_2| \\ &= 1-\alpha\end{aligned}$$

b. nothing happens to C_1, C_2 increases by 1

$$\begin{aligned}\Delta u &= |(1-\alpha)(C_1) - \alpha(C_2+1)| - |(1-\alpha)C_1 - \alpha C_2| \\ &\leq |C_1 - \alpha C_1 - \alpha C_2 - \alpha - C_1 + \alpha C_1 + \alpha C_2| \\ &= \alpha\end{aligned}$$

Subtracting a point gives you same thing but with some negative signs inside the absolute values that come out in the wash. □

2.3 The Normalization Factor

References

[DFM+20] Wenxin Du, Canyon Foot, Monica Moniot, Andrew Bray, and Adam Groce. Differentially private confidence intervals. *arXiv preprint arXiv:2001.02285*, 2020.

[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.

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

[Smi11] Adam Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 813–822, 2011.