CS 860: Algorithms for Private Data Analysis Fall 2020

Problem Set 2

Prof. Gautam Kamath *Deadline: 11:59 PM on October 27, 2020*

You are allowed to discuss the problems in pairs. List your collaborator for each problem. Every person must write up and submit their own solutions. Include a proof for everything the problem asks you to show. Allowed references are anything given on the course website. It might be possible to find solutions to these problems online, but please do not search for them. If you have already seen a solution before, solve it without referring to said reference.

1. **Amplification by subsampling.** Suppose we have an algorithm $M : \mathcal{X}^m \to \mathcal{Y}$ which is (ε, δ)-DP. Consider the following algorithm $M' : \mathcal{X}^n \to \mathcal{Y}$, where $n > m$. When run on an input $X \in \mathcal{X}^n$, it chooses a random subset of the input $X' \in \mathcal{X}^m$ of size m, and outputs $M(X')$. Prove that M' is $(O(\varepsilon m/n), O(\delta m/n))$-DP.

   This means that given a 1-DP algorithm, it can always be converted to a ε-DP algorithm by just expanding the dataset size by a factor of $O(1/\varepsilon)$. Thus, while there may be a qualitative difference between DP algorithms with large and constant ε, going from constant to small ε never experiences the same type of difference.

2. **Broken sparse vector is not DP.** It is notoriously hard to get the Sparse Vector algorithm right. In the version discussed in class, we noise the threshold T at the start of the algorithm to obtain a new threshold $\tilde{T}$, and we additionally noise each query. A common mistake is to not noise the threshold (just using T instead), and only the queries. For instance, this mistake is present in the original paper on Private Multiplicative Weights. Show that this variant is not ε-DP for any $\varepsilon > 0$.

3. **Tightness of Advanced Composition.** Advanced composition says that, if we run k ε-DP algorithms on the same dataset, the result will be $(1, \delta)$-DP if $k < 1/8\varepsilon^2 \log(1/\delta)$. Show that this is "tight": if we run k ε-DP algorithms on the same dataset where $k > C/\varepsilon^2$ for some large constant C, the result may not be $(1, 0.1)$-DP. If you like, you can also replace C with some polynomial in $\log(1/\varepsilon)$, if that makes your life easier.

4. **Median via the Exponential Mechanism.** Consider a dataset $X \in \mathbb{R}^n$ of size n, which is over the set of integers $[m] = \{1, \ldots, m\}$. We will use the exponential mechanism to privately find a median of this dataset. The score function will be $q(X, h) = -|\sum_{i \in [n]} \text{sign}(h - X_i)|$, where $\text{sign}(x) = 1$ if $x > 0$, 0 if $x = 0$, and -1 if $x < 0$.

   (a) Let $n = m = 6$. Draw the score function for the following three cases: i) all $X_i = 3$, b) $X_i = i$, c) $X_i = 1$ for $i \leq 3$ and $X_i = 6$ for $i \geq 4$. In general: what are the maximum and minimum values of the score function, and when is it achieved?

   (b) What is the sensitivity of the score function? Be sure to define what notion of neighbouring database you use.

   (c) Show that the exponential mechanism, when instantiated for this problem, has the following utility guarantee. Let $\text{position}(h, X)$ be the position in $\{0, 1, \ldots, n\}$ that h would have in the sorted order of X. For some constant $C > 0$, the probability that it outputs a point h such that $|\text{position}(h, X) - n/2| \geq C(\ln m + \ln(1/\beta))/\varepsilon$ is at most β.

(d) (Optional. Valid for extra credit, but will not take you above the maximum of 25% for homeworks.) Describe in detail how to modify this algorithm and analysis in order to estimate general quantiles, instead of just the median. What type of utility guarantees do you get?