

## Problem Set

Prof. Gautam Kamath

Deadline: 11:59 PM on Monday, October 24, 2022

You are allowed to discuss the problems in groups. List your collaborators for each problem. Every person must write up and submit their own solutions. 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. **A different private algorithm.** Suppose that we wanted to answer a count query:  $f(X) = \sum_{i=1}^n X_i$ , where  $X_i \in \{0, 1\}$ . In class, we learned the Laplace mechanism: simply add Laplace noise with scale parameter  $1/\varepsilon$ . But what if we didn't have access to Laplace noise? Suppose  $Z$  is a continuous uniform random variable, drawn uniformly from the interval  $[-3/\varepsilon, 3/\varepsilon]$ . Consider the statistic  $\tilde{f}(X) = \sum_{i=1}^n X_i + Z$ . Is  $\tilde{f}$   $O(\varepsilon)$ -differentially private? If yes, prove it, with the best constant you can give in the privacy guarantee. If no, explain why not.
2. **Randomized Response, re-revisited.** We'll see some generalizations of randomized response, beyond just binary alphabets. I will informally and vaguely describe an algorithm, you must rigorously define and specify the algorithm and prove that it is  $\varepsilon$ -differentially private.
  - (a) Let's start by revisiting the binary case. The analysis of randomized response we gave in class was sloppy in two ways: first, it used big-Oh notation, and only worked for sufficiently small  $\varepsilon$ . Give a randomized response algorithm and analysis which works for all  $\varepsilon > 0$ . More precisely: the vector  $(Y_1, \dots, Y_n) \in \{0, 1\}^n$  is output, where  $Y_i$  is equal to  $X_i$  with probability proportional to  $g(\varepsilon)$  (for some function  $g$  which you must specify), and equal to  $1 - X_i$  with probability proportional to  $1$ . Informally speaking, this algorithm will be "exact" – the differential privacy guarantee will hold with equality.
  - (b) Let's generalize this beyond the binary alphabet, assume  $X_i \in \{1, \dots, k\}$  for the remainder of this problem. The vector  $(Y_1, \dots, Y_n) \in \{1, \dots, k\}^n$  is output, where  $Y_i$  is equal to  $X_i$  with probability proportional to  $g(\varepsilon)$  (for some function  $g$  which you must specify), and equal to each  $s \in \{1, \dots, k\} \setminus X_i$  with probability proportional to  $1$ .
  - (c) Here's another way to generalize randomized response. The vector  $(Y_1, \dots, Y_n) \in \{0, 1\}^{kn}$  is output.  $Y_i \in \{0, 1\}^k$  is a vector generated in the following manner: each  $X_i$  is first converted to a "one-hot" vector  $\in \{0, 1\}^k$ , where coordinate  $j$  is 1 if  $j = X_i$  and 0 otherwise.  $Y_i$  generated from  $X_i$  by applying a bitwise randomized response (with appropriate parameter).
3. **Mean estimation with non-binary data.** In class, we saw how to estimate the mean of a dataset  $f(X) = \frac{1}{n} \sum_{i=1}^n X_i$  in the case when the  $X_i$ 's are binary. Here, we will see how to estimate the mean of a dataset when this may not be the case.
  - (a) Suppose we only knew the  $X_i \in \mathbb{R}$  were real numbers. Prove that, for all  $t \geq 0$ , there is no  $\varepsilon$ -differentially private algorithm  $M : \mathbb{R}^n \rightarrow \mathbb{R}$  such that  $\Pr[|M(X) - f(X)| \leq t] \geq 9/10$ , where  $\varepsilon = 1$ . **Optionally**, prove the same statement for finite all  $\varepsilon > 0$ .

(b) The previous problem showed that, in general, we can't privately estimate the mean of an unbounded dataset. Let's see how we can circumvent this issue. Give an algorithm  $A_2 : \mathbb{R}^n \rightarrow \mathbb{R}$  with the following guarantees. The algorithm is  $\varepsilon$ -differentially private, for all possible datasets  $(X_1, \dots, X_n) \in \mathbb{R}^n$ . If all  $X_i \in [-R, R]$ , then there exists some constant  $C > 0$  such that  $\Pr[|A_2(X) - f(X)| \leq \frac{CR}{\varepsilon n}] \geq 9/10$ . The parameter  $R$  is known to the algorithm. Observe that this algorithm must always be private, but only needs to be accurate when the input dataset satisfies some additional properties.

(c) You have now shown that, if the data is in some known bounded range, we can privately estimate its mean. However, this can still be wasteful if  $R$  is large, but the data is actually concentrated in a much tighter range. The latter is often the case: for instance, given Gaussian data sampled from  $N(0, 1)$ , almost all of the data will lie in the range  $[-3, 3]$ . Thus, the last two parts of this problem will attempt to reduce the dependence on  $R$  when something like this holds.

Suppose we are given a dataset  $X \in \mathbb{R}^n$ . Give an algorithm  $A_1$  with the following guarantees. The algorithm is  $\varepsilon$ -differentially private, for all possible datasets  $(X_1, \dots, X_n) \in \mathbb{R}^n$ . If there exists some interval  $I \subset [-R, R]$  such that all  $X_i \in I$  and the width of the interval  $I$  is bounded by 2, and if  $n \geq \frac{C \log R}{\varepsilon}$  for some constant  $C > 0$ , then with probability at least  $9/10$  the algorithm outputs an interval  $J$  such that  $I \subset J$  and the width of  $J$  is bounded by some constant (you can choose the constant, but prove it explicitly). Again, assume the parameter  $R$  is known to the algorithm, but the interval  $I$  is not.

(d) Give an algorithm  $A$  with the following guarantees. The algorithm is  $\varepsilon$ -differentially private, for all possible datasets  $(X_1, \dots, X_n) \in \mathbb{R}^n$ . If all  $X_i \in I$  for some interval  $I \subset [-R, R]$  of width at most 2, and if  $n \geq \frac{C_1 \log R}{\varepsilon}$  for some constant  $C_1 > 0$ , then there exists some constant  $C_2 > 0$  such that  $\Pr[|A(X) - f(X)| \leq \frac{C_2}{\varepsilon n}] \geq 4/5$ . (This problem should be easy, given the answer to the previous two parts.)

4. **Amplification by subsampling.** Suppose we have an algorithm  $M : \mathcal{X}^m \rightarrow \mathcal{Y}$  which is  $(\varepsilon, \delta)$ -DP. Consider the following algorithm  $M' : \mathcal{X}^n \rightarrow \mathcal{Y}$ , 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  $\varepsilon$ -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  $\varepsilon$ , going from constant to small  $\varepsilon$  never experiences the same type of difference.

5. ~~**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  $\varepsilon$ -DP for any  $\varepsilon > 0$ .

6. **Tightness of Advanced Composition.** Advanced composition says that, if we run  $k$   $\varepsilon$ -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  $k$   $\varepsilon$ -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.

7. **Noisy Report Max.** Suppose we are in the same setting as the exponential mechanism. That is, we have a dataset  $X \in \mathcal{X}^n$ , a set of objects  $\mathcal{H}$ , and a score function  $s(\cdot, \cdot) : \mathcal{X}^n \times \mathcal{H} \rightarrow \mathbb{R}$  which maps from a pair of a dataset and an object to a real value. Suppose that, for any  $h \in \mathcal{H}$ , the sensitivity of  $s$  is at most  $\Delta$ . Here is an algorithm which is very similar to the exponential mechanism, but may be easier to implement in some cases. Return  $\arg \max_{h \in \mathcal{H}} \{s(h, X) + Z_h\}$ , where  $Z_h \sim \text{Exp}(2\Delta/\varepsilon)$  drawn i.i.d. for each  $h \in \mathcal{H}$ . In words: compute the score of each object, noise each score with an independent draw from the exponential distribution, and return the object which has the highest (noisy) score. [Hint: Suppose you fix (condition on) the values of the noise  $Z_y = z_y$  for all  $y \neq h$ . Compute and compare the probability that the outcome will be  $h$  under two neighbouring data sets  $X$  and  $X'$ .]

Note that this algorithm, known as noisy report max, is *not* identical to the exponential mechanism, this claim is sometimes incorrectly made. If you replace the exponential noise with Gumbel noise, then the resulting algorithm is *exactly* equivalent to the exponential mechanism. **Optionally**, prove that this is true.