HW 5: Beyond Noise Addition

CS 208 Applied Privacy for Data Science, Spring 2022

Version 3.1: Due Fri, Mar. 4, 5:00pm.

Instructions: Submit a single PDF file to Gradescope containing your solutions, plots, and analyses. As usual, the notebooks from class and section may be helpful starting points for coding solutions. Submit any code files and notebooks separately on Gradescope. Make sure to list all collaborators and references.

1. **Continuous Exponential Mechanism:** In class, we saw how to use the exponential mechanism to compute a differentially private estimate of a median of a dataset on a discrete domain $\mathcal{X}$, using the following score function for each possible output $y \in \mathcal{Y}$, where we set $\mathcal{Y} = \mathcal{X}$ for median computations: $s(x, y) = \min\{\#\{i : x_i \leq y\}, \#\{i : x_i \geq y\}\}$.

   In this problem, you will apply the exponential mechanism to estimate medians on *continuous* data. If we have a continuous output domain $\mathcal{Y} = [-1, 1]$, the exponential mechanism produces output $M(x) = Y$ where Y has probability density function f_Y given as follows:

$$f_Y(y) = \begin{cases} \frac{\exp\left(\frac{\epsilon \cdot s(x,y)}{2 \cdot GS_s}\right)}{\int_{-1}^{1} \exp\left(\frac{\epsilon \cdot s(x,z)}{2 \cdot GS_s}\right) dz} & \text{if } y \in [-1, 1]. \\ 0 & \text{if } y \notin [-1, 1]. \end{cases}$$

   (a) Write a function that takes in a dataset x, and outputs the private median of $x \in [-1, 1]^n$ using the above mechanism. (Hint: observe that f_Y is piecewise constant.)

   (b) Generate a dataset $x \in \mathcal{X}^n$ from the truncated normal distribution $\left[\mathcal{N}(1/2, \sigma^2)\right]_{-1}^{1}$. Vary σ from .1 to .5, and for each σ, compute the private median of x using your function from part (a) with $\epsilon = .15$ and $n = 200$. Run many Monte Carlo (at least 100) trials to estimate the RMSE of the private median, and then plot the error against σ.

   (c) Notice from part (b) that the exponential mechanism does not output very accurate answers when the data is too concentrated. Briefly explain the reason for this phenomenon.

2. **DP Synthetic Data:** In this problem, you will create and analyze DP synthetic data. You will compare the results of running a regression on the synthetic data with your DP regression algorithm from HW4.

   (a) **Write a DP synthetic data function.** Write a function that takes a data bound b, a dataset $z = ((x_1, y_1), \ldots, (x_n, y_n)) \in (([-b, b] \times [-b, b])^n$ a parameter $\epsilon \geq 0$, a binning parameter k, and does the following: (a) bins the datapoints using a $k \times k$ grid, (b) constructs a ϵ-DP histogram h of of the dataset, and (c) post-processes h to construct an ϵ-DP synthetic dataset $\tilde{z} = ((\tilde{x_1}, \tilde{y_1}), \ldots, (\tilde{x_m}, \tilde{y_m})) \in ([-1, 1] \times [-1, 1])^m$.

(b) **Perform and evaluate linear regression on the synthetic data.** Given a DP synthetic data generator from Part 2a, we can perform DP linear regression by post-processing, running standard OLS regression on $\tilde{z}$ to obtain an ϵ-DP slope $\tilde{\beta}$ minimizing $\sum_i (\tilde{y}_i - \tilde{\beta}\tilde{x}_i)^2$. Evaluate the resulting DP regression estimates exactly as in HW4 Problem 2d, using parameters $\epsilon = .1$, $b = 1$, and $k = 20$, and many Monte Carlo trials of the following: For each $n = 100, 200, 300, \ldots, 5000$, generate a dataset z, where the x_i's are uniform in $[-1/2, 1/2]$ and $y_i = [x_i + \mathcal{N}(0, .2)]_{-1}^{1}$. Plot the bias and standard deviation of both the OLS estimate $\hat{\beta}$ and the DP estimate $\tilde{\beta}$ obtained by post-processing the DP histogram. Your plot should have n on the x-axis, and bias and standard deviation on the y-axis on a scale from -1.0 to 1.0. Try to run enough trials to obtain smooth curves.

(c) **Compare the error.** Compare the bias and standard deviation of the above DP-histogram-based regression with the results obtained on HW4. Give an intuitive explanation for the differences you find.

3. **Composition:**

(a) Suppose you have a global privacy budget of $\varepsilon = 1$ and are willing to tolerate $\delta = 10^{-9}$ and you want to release k count queries (i.e., sums of Boolean predicates[1]) using the Laplace mechanism with an individual privacy loss of ε_0. By basic composition, you can set $\varepsilon_0 = \varepsilon/k$. Using the advanced composition theorem, you can set $\varepsilon_0 = \varepsilon/\sqrt{2k\ln(1/\delta)}$. For the two choices (basic and advanced composition), plot (on the same graph) the standard deviation of the Laplace noise added to each query as a function of k.

(b) As we saw, for the 2020 Decennial Census, the US Census Bureau used a variant of differential privacy called zCDP (Zero-concentrated Differential Privacy), that is tailored to analyzing the Gaussian mechanism and its compositions. The formal definition of zCDP is not needed for this problem, but only that zCDP has a single privacy-loss parameter $\rho \geq 0$ and has the following properties:

i. The Gaussian mechanism with noise of variance $\sigma^2 = (\mathrm{GS}_q)^2/2\rho$ is ρ-zCDP, where GS_q is the global sensitivity of the query q.

ii. Suppose $\mathcal{M}_1$ satisfies ρ_1-zCDP and $\mathcal{M}_2$ satisfies ρ_2-zCDP. Then their composition $(\mathcal{M}_1, \mathcal{M}_2)$ satisfies $(\rho_1 + \rho_2)$-zCDP.

iii. If a mechanism $\mathcal{M}$ satisfies ρ-zCDP, then for every $\delta > 0$, it satisfies (ε, δ)-DP for $\varepsilon = \rho + \sqrt{4\rho\ln(1/\delta)}$.

We can calculate the smallest value of σ that ensures $(\varepsilon = 1, \delta = 10^{-9})$-DP when using the above properties to analyze the Gaussian mechanism for answering k counting queries. To see the benefit one gets from using zCDP, plot (on the same graph) the standard deviation of the Gaussian noise added to each query as a function of k using the composition of zCDP against that of basic and advanced composition for approximate DP (from part (a)). From your plot, for what value of k does the Gaussian mechanism outperform advanced composition (from part (a))?

[1] A Boolean predicate is a function that returns a 0 or a 1. An example of a count query might be the number of Harvard college students that live in the Quad.