HW 5: Beyond Noise Addition

CS 208 Applied Privacy for Data Science, Spring 2022

Version 1.0: Due Fri, Mar. 4, 5:00pm.

Instructions: Submit a single PDF file to Gradescope containing your solutions, plots, and analyses. Submit any code files and notebooks separately on Gradescope. Make sure to list all collaborators and references.

1. **Continuous Exponential Mechanism:**

   In HW3, we saw an instantiation of a continuous version of the exponential mechanism. To privately compute the median of a fixed dataset $x \in \mathcal{X}^n$, we consider the following score function: [1]

   $$s(x, y) = \min\{\#\{i : x_i \leq y\}, \#\{i : x_i \geq y\}\}.$$

   The sensitivity of $s(x, y)$ is defined as

   $$GS_s = \max_{x \sim x', y} |s(x, y) - s(x', y)|,$$

   where $x \sim x'$ are neighboring datasets that differ in one row. Recall that the exponential mechanism $\mathcal{M}(x)$ with privacy parameter ϵ selects and outputs an element $y \in \mathcal{Y}$ with probability proportional to $\exp\left(\frac{\epsilon \cdot s(x,y)}{2 \cdot GS_s}\right)$. Suppose each datapoint is within $[0, 1]$ (i.e., $\mathcal{X} = [0, 1]$) and the potential output space for releasing the median is $\mathcal{Y} = [0, 1]$. The instantiation of a continuous version of the exponential mechanism for releasing a private median is $M(x) = Y$ where Y has probability density function f_Y given as follows:

   $$f_Y(y) = \begin{cases} \frac{\exp\left(\frac{\epsilon \cdot s(x,y)}{2 \cdot GS_s}\right)}{\int_0^1 \exp\left(\frac{\epsilon \cdot s(x,z)}{2 \cdot GS_s}\right) dz} & \text{if } y \in [0, 1]. \\ 0 & \text{if } y \notin [0, 1]. \end{cases}$$

   (a) Write a function that takes in a dataset x, and outputs the private median of $x \in \mathcal{X}^n$ using the above mechanism. [2]

   (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). 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 behind this observation.

[1] Note that, using the score function, every element—whether in or out of x—has score at most $n/2$ (including the median).

[2] This notebook might be helpful as a starting point: `https://github.com/opendp/cs208/blob/main/spring2022/examples/wk5_exponential.ipynb`.

2. **Synthetic Data:** In this problem, you will expand the template from class to 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) **Create synthetic data.** First, generate similar data from HW4, where the x_i variables are generated uniformly random from $[-5, 5]$ and the y_i's are generated according to a linear model with slope 1, intercept 0, and Gaussian noise, but clipped to $[-10, 10]$:

   $$y_i = [x_i + \mathcal{N}(0, .2)]_{-10}^{10}.$$

   We will treat this data as sensitive. Then, create a DP histogram[3] release of income and education. You do not need to graph this histogram, just compute the release for each binned combination of the variables. From this, you should be able to generate synthetic data of these two variables. (For continuous data, you will need to do binning to apply a histogram.)

   (b) **Run a linear regression algorithm.** Run a linear regression as a post-process on your synthetic data, predicting income from education [4] using the equation:

   $$y_i = \beta_1 x_i + \nu_i; \qquad \nu_i \sim \mathcal{N}(0, \sigma^2) \tag{1}$$

   (c) **Compute the error.** Let $\beta^* = \beta_1^*$ be the true coefficients in the full sensitive data, while $\tilde{\beta} = \tilde{\beta}_1$ is the DP release we generate. The mean-squared error of a DP release of $\tilde{\beta}$ can be decomposed into the contributions of bias and variance as:

   $$\text{MSE}(\tilde{\beta}) = \text{bias}(\tilde{\beta})^2 + \text{var}(\tilde{\beta}) = (\text{E}[\beta^* - \tilde{\beta}])^2 + \text{E}[(\bar{\tilde{\beta}} - \tilde{\beta})^2], \tag{2}$$

   where $\bar{\tilde{\beta}}$ is the average of the generated DP estimates $\tilde{\beta}$. For this calculation, we are taking the (sensitive) regression coefficients β^* on the entire dataset as the true values of β. Show the contributions to MSE of the bias and variance of the DP-regression coefficients.[5]

   (d) **Analyze the error.** As a baseline to decide if these squared bias and error terms are large, compare with the error on your HW4 DP regression algorithm. How do the bias and variance terms compare?

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[6]) 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, plot (on the same graph) the standard deviation of the Laplace noise added to each query as a function of k.

[3] That is, a histogram representation counting the occurrences of having all possible combinations of the two binned variables.

[4] You will likely find that `log(income)` has a more linear relationship with education, so feel free to shift from `income` to `log(income)` if you prefer. However, you will need to decide how to treat zero values in income; one option is to clip the lower bound of income to some small positive value.

[5] To numerically compute the expectations, simply repeat your simulation many times and average.

[6] 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.

(b) There is another variant of DP, 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 note 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 $\varepsilon = \rho + \sqrt{4\rho \log(1/\delta)}$.

Using these properties, we can calculate the smallest value of σ that ensures $(\varepsilon = 1, \delta = 10^{-9})$-DP when using the Gaussian mechanism to answer 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 the 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))?