HW 2: Differential Privacy Foundations

CS 208 Applied Privacy for Data Science, Spring 2019

Version 1.2: Due Tuesday, March 12, 11:59pm.

Instructions: Submit a single PDF file containing your solutions, plots, analyses, and documented code. Also include a link to a public repository with your code (such as GitHub/GitLab). Make sure to list all collaborators and references.

1. **Mechanisms:** Consider the following mechanisms M that takes a dataset $x \in [0,1]^n$ and returns an estimate of the mean $\bar{x} = (\sum_{i=1}^n x_i)/n$.

   i $M(x) = [\bar{x} + Z]_0^1$, for $Z \sim \text{Lap}(2/n)$, where for real numbers y and $a \leq b$, $[y]_a^b$ denotes the "clamping" function:

   $$[y]_a^b = \begin{cases} a & \text{if } y < a \\ y & \text{if } a \leq y \leq b \\ b & \text{if } y > b \end{cases}.$$

   ii $M(x) = \bar{x} + [Z]_{-1}^1$, for $Z \sim \text{Lap}(2/n)$.

   iii

   $$M(x) = \begin{cases} 1 & \text{w.p. } \overline{x} \\ 0 & \text{w.p. } 1 - \overline{x}. \end{cases}.$$

   iv $M(x) = Y$ where Y has probability density function f_Y given as follows:

   $$f_Y(y) = \begin{cases} \frac{e^{-n|y-\bar{x}|/10}}{\int_0^1 e^{-n|y-\bar{x}|/10} dy} & \text{if } y \in [0,1]. \\ 0 & \text{if } y \notin [0,1]. \end{cases}$$

   (This is an instantiation of a continuous version of the exponential mechanism.)

   (a) Which of the above mechanisms meet the definition of $(\epsilon, 0)$-differential privacy for a finite value of ϵ, and what is the smallest value of ϵ (possibly as a function of n) for which they do?

   (b) For those that do not meet the definition, calculate the smallest value of δ (again possibly as a function of n) for which they satisfy (ϵ, δ) differential privacy for a finite value of ϵ.

   (c) Describe how you would modify the algorithms to have tunable privacy parameters ϵ (and δ in case of mechanisms that require it) and tunable data domain $[a,b]$ (rather than $[0,1]$).

   (d) Which of these algorithms do you consider to be "best" for releasing a mean and why? (There is not a single "right" answer for this problem.)

2. **Evaluating DP Algorithms with Synthetic Data:** Consider a dataset $x \in \mathbb{N}^n$ drawn from a Poisson process, which has probability distribution $\Pr[x_i = k] = 10^k e^{-10}/k!$ for natural numbers k (where we consider $k = 0$ to be a natural number and define $0! = 1$).

   (a) Write a *data generating process* (DGP) function that generates a dataset $x \in \mathbb{N}^n$ according the above Poisson process.

   (b) Pick one of your differentially private mechanisms from question 1 (generalized to allow for arbitrary choices of ϵ and data range $[a, b]$ as parameters) that releases an estimate of $\bar{x}$. Implement this mechanism as a function which is given a vector of values $x \in [a, b]^n$ and an ϵ and makes a differentially private release. You can assume the sample size n is public knowledge. To apply your mechanism to unbounded data $x \in \mathbb{R}^n$, you will have to clamp x to a chosen range $[a, b]$. For simplicity, we will fix $a = 0$ and only consider the effect of varying b.

   (c) Recall the discussion on clamping from class; if the range is large, the sensitivity increases, so noise increases and utility drops. However, if you clip the values too aggressively the answer will be biased, and again utility will drop. For $n = 200$ and $\epsilon = .5$, plot the root mean squared error as a function of the upper bound b. Identify the approximate optimal value b^* of b for this data distribution.

   (d) Suppose we have an actual (not synthetic) dataset $x \in \mathbb{N}^n$ for which we want to release a differentially private mean, and we don't know the underlying distribution of x. Again, we need to select the parameter b and want to do so in a way that minimizes the error. A natural idea is to use a *(nonparametric) bootstrap*[1] to generate many datasets that are "similar" to x in place of the data-generating process above, and optimize the choice of b as above. Once we find an optimal value b^*, we then do our differentially private release on the dataset x itself.

   Explain why this approach is not safe in general and may violate differential privacy.

   (e) Propose some alternative methods for determining a good upper bound b for a given sensitive dataset x, while continuing to satisfying the definition of differential privacy.

3. **Regression:** Consider a dataset where each of its n rows is a pair of real numbers (x_i, y_i), where x_i is drawn from a Poisson distribution as in Problem 2, and y_i is then a noisy linear function of x_i, specifically:

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

   for unknown parameters α, β, σ.

   One simple way to estimate the parameters α and β without privacy is by a simple linear regression, using the following estimators:

$$\hat{\beta} = \frac{S_{xy}}{S_{xx}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}; \tag{3}$$

   (a) In class and section, we saw an algorithm and implementation to produce a differentially private version of $\hat{\beta}$.[2] Augment this implementation to also produce a differentially private

[1] In a nonparametric bootstrap, we generate new datasets by sampling with replacement from x itself.

[2] See `/examples/wk3_dp_foundations/laplaceRegressionRelease.ipynb` and `laplaceRegressionRelease.r` for the regression implementations shown in section, building on the code from class.

version of $\hat{\alpha}$, so that the overall method for computing both $\hat{\alpha}$ and $\hat{\beta}$ is ϵ-DP, for an input parameter ϵ. If your algorithms use several basic differentially private algorithms as subroutines, divide the overall privacy budget of ϵ among them evenly. You can clamp both the x_i's and y_i's using reasonable values of your choosing (perhaps following your choice from Problem 2 for x). Describe the reasoning behind your choice in your write up.

(b) Evaluate the performance of your algorithm using a Monte Carlo simulation with synthetic data as in Problem 2 Parts (a)–(c), for the parameters $\alpha = \beta = \sigma = \epsilon = 1$ and $n = 1000$. Measure utility by the mean-squared residuals:

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\beta}x_i - \hat{\alpha}\right)^2.$$

(The non-private method for simple linear regression described above is chosen to minimize this quantity, hence the term "least-squares regression".) Plot and compare the distributions of mean-squared residuals you get with your differentially private simple linear regression and with a non-private simple linear regression.

(c) Now, run experiments to see if there is a different partition of your privacy budget ϵ in your algorithm that yields better utility. Use a grid search[3] to explore different partitions of ϵ and see if you find one that is convincingly better (in terms of mean-squared residuals) than an equal partition under the given data distribution. Show and explain your results.

4. **DP vs. Reconstruction Attacks:** Suppose $M : \{0,1\}^n \to \mathcal{Y}$ is an (ϵ, δ)-DP mechanism and $A : \mathcal{Y} \to \{0,1\}^n$ is an adversary that is trying to reconstruct the sensitive bits in the dataset $x \in \{0,1\}^n$ from the output $M(x)$. Suppose the dataset is a random variable $X = (X_1, \ldots, X_n)$ consisting of n iid draws from a Bernoulli(p) distribution, for a known value of p. Prove that the expected fraction of bits that the adversary successfully reconstructs is not much larger than the trivial bound of $\max\{p, 1-p\}$ (which can be achieved by guessing the all-zeroes or all-ones dataset). Specifically:

$$\mathrm{E}\left[\#\{i \in [n] : A(M(X))_i = X_i\}/n\right] \leq e^{\epsilon} \cdot \max\{p, 1-p\} + \delta.$$

(Hint: write the quantity inside the expectation as an average of indicator random variables, and for each i, consider running M on the dataset $X^{(i)}$ where we replace the i'th row of X with the fixed value 0.)

5. **Final Project Next Step:** You will have received comments on your initial project sketch from homework 1. Using these comments, and rereading the "Final Project Guidelines" (`http://seas.harvard.edu/~salil/cs208/spring19/project-guidelines.pdf`) document on the course website, refine your topic ideas and, if you wish, seek out 1-2 collaborators on Piazza. Your group should submit a revised project proposal (or two) approximately a half-page long. (All group members should submit the same proposal.) Include at least one potential dataset you might use for experiments (unless you're doing a purely theoretical project), three citations to related works that either describe the use case (these could be popular press articles), the privacy risks, and/or methods you might build on. Also ask concrete questions where you could use additional pointers or guidance from us.

[3] Grid search is a term from optimization and machine learning that refers to an exhaustive search through the hyperparameter space discretized into a grid (to make the search finite).