Privacy in Statistics and Machine Learning Spring 2025
In-class Exercises for Lecture 6 (Exponential Mechanism and RNM)
February 6, 2025

Adam Smith (based on materials developed with Jonathan Ullman)

Problems with marked with an asterisk (*) are more challenging or open-ended.

1. Suppose we run the exponential mechanism to choose between just two outputs (say, election candidates Alice and Bob) in a simple election where each voter votes for just one candidate. Suppose Alice gets a votes and Bob gets b votes (where $a + b = n$ is the total number of votes).

   What are:

   (a) The output set $\mathcal{Y}$? The score function q? The sensitivity bound Δ for q?

   (b) The "odds ratio" $\frac{\Pr(Y=\text{'Alice'})}{\Pr(Y=\text{'Bob'})}$, assuming Y is the outcome of the exponential mechanism with for this problem with input $\varepsilon = 0.1$? (Express your answer as a function of $a - b$).

   (c) How big of a margin $a - b$ must Alice have for her name to be output with probability at least 95%?

2. The exponential mechanism is often used in private machine learning. Suppose our data set consists of pairs (x_i, z_i) where x_i is an image (e.g., represented as a grid of pixels), and z_i is a label (perhaps indicating whether the picture is of Beyoncé (labeled "+1") or Harry Styles (labeled "-1", even though we're not judgy that way).

   We are given a collection of k possible classifiers $f_1, \ldots, f_k$ (perhaps representing different settings of weights in a neural network) among which we want to choose and output one with low *training error*. The training error of a classifier f is the fraction of misclassified points in the input data set:

$$\text{error}(f) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{z_i = f(x_i)\}\,.$$

   How can you cast this as a selection problem? If you use the exponential mechanism or report noisy max, what bounds do the lecture notes imply for the (expected, asymptotic) difference between the error of the classifier that is output and the error of the best classifier? Express your answer as a function of k, n and ε.

3. Suppose you have a graph with a fixed vertex set V, and where each individual data point x_i is an undirected edge $\{u, v\} \in V \times V$. For example, the nodes might represent locations, and an edge $\{u, v\}$ might represent the locations between which an individual travels most often.

   Consider the problem of finding a near-minimum cut in the graph. This is a partition of V into two disjoint sets A, B of nodes. The *weight* of the cut is the number of edges that cross from A to B (so $u \in A$ and $v \in B$ or vice versa). The weight of a cut can be as large as the size of the data set n, and n can be as large as $\Omega(|V|^2)$.

(a) Use the exponential algorithm (or report noisy max) to design an algorithm that returns a cut with expected weight min-weight + $O(|V|/\varepsilon)$.
It's OK if your algorithm runs in time polynomial in $2^{|V|}$.

(b) **(**)** There can be multiple distinct minimum cuts in a graph. However, one neat (and highly non-trivial to prove) fact is that if w^* is the number of edges in the minimum cut, the number of distinct cuts with weight $\leq cw^*$ is at most $O(|V|^{2c})$. Using this fact, prove that the error of the exponential mechanism (or RNM) is actually much better, and it outputs a cut with expected weight min-weight + $O(\log(|V|)/\varepsilon)$.

4. **Medians.** Suppose we want to find the median of a list of real numbers $\mathbf{x} = (x_1, ..., x_n)$ that lie in the set $\{1, ..., R\}$.

   Consider an instantiation of the exponential mechanism based on the following score function: For every $y \in \{1, ..., R\}$, let

$$q(y; \mathbf{x}) = -\left|\sum_{i=1}^{n} sign(y - x_i)\right|, \qquad \text{where} \qquad sign(z) = \begin{cases} 1 & \text{if } z > 0\,, \\ 0 & \text{if } z = 0\,, \\ -1 & \text{if } z < 0\,. \end{cases}$$

   If all the input values are distinct, this score is 0 exactly when y is a valid median for $\mathbf{x}$. In general, the score will be minimized at the true median.

   (a) Show that q has global sensitivity 1 when neighboring data sets are allowed to differ by the insertion or deletion of one entry. (Alternatively, show that it has global sensitivity 2 when neighboring datasets differ by replacing one value.)

   (b) Let A_ε be the algorithm one gets by instantiating the exponential mechanism with score q, parameter ε and output set $\mathcal{Y} = \{1, ..., R\}$. Show that there is a constant $c > 0$ such that: for every data set $\mathbf{x}$, for every R and $\varepsilon < 1$, and for every $\beta \in (0, 1)$, the probability that $A_\varepsilon(\mathbf{x})$ samples a value y with $|rank_{\mathbf{x}}(y) - n/2| > c \cdot \frac{\ln(R)+\ln(1/\beta)}{\varepsilon}$ is at most β. Here $rank_{\mathbf{x}}(y) \in \{0, 1, ..., n\}$ is the position y would have in the sorted order of $\mathbf{x}$.
   For this part, it is ok to assume distinct data values, so that the rank of a value is uniquely defined. [*Hint:* How does $rank_{\mathbf{x}}(\cdot)$ relate to $q(\cdot; \mathbf{x})$? Look at the ratio between the probability mass of a true median and the probability mass of an element with very low or high rank.]

5. **(*)** Prove that Report Noisy Max with exponential noise (Alg. 2 in the notes) is differentially private.

6. Show that the accuracy guarantees we showed for the exponential mechanism (and RNM) are basically tight in general. Specifically, consider the family of data sets $\left\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(d)}\right\}$ defined as follows: in $\mathbf{x}^{(j)}$, one candidate j receives $q_{\max} = n = \frac{\ln(d)}{2\varepsilon}$ votes and all others receive 0 votes.

   (a) Show that on such inputs, the algorithm A_{EM} will return a candidate other than j (that is, a candidate who received 0 votes) with constant probability, independent of d.

   (b) **(*)** Show that for **every** ε-differentially private A algorithm, if we choose J uniformly at random in $[d]$, then with constant probabililty $A(\mathbf{x}^{(J)})$ will return a candidate other than J. [That is, A will fail to find the winner for many datasets of this form.]