Privacy in Statistics and Machine Learning Spring 2021
In-class Exercises for Lecture 16 (Synthetic data generation and MW-EM)
March 25/26, 2021

Adam Smith and Jonathan Ullman

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

1. In the previous lecture we showed that MWEM is (ε, δ)-differentially private and can answer a set of k queries on a dataset in $\mathcal{U}^n$ with error at most $\leq \alpha$ on every query (with high probability), provided that

$$n \gtrsim \frac{(\log |\mathcal{U}|)^{1/2}(\log \frac{1}{\delta})^{1/2}(\log k)}{\varepsilon \alpha^2}$$

Modify the analysis of the algorithm to ensure $(\varepsilon, 0)$-differential privacy? Prove a similar guarantee to above, showing that the algorithm is accurate provided that

$$n \gtrsim \frac{(\log |\mathcal{U}|)^a (\log k)^b}{\varepsilon^c \alpha^d}$$

for some constants a, b, c, d. What parts of the algorithm and its analysis have to change?

2. Consider a two-player zero-sum game described by a payoff matrix $M \in \mathbb{R}^{|\mathcal{R}| \times |C|}$ and let $(\mathbf{r}, \mathbf{c})$ be a pair of equilibrium strategies. The support of a strategy is the set of actions with non-zero probability, so

$$\text{supp}(\mathbf{r}) = \{i : \mathbf{r}_i > 0\}$$

and likewise for $\text{supp}(\mathbf{c})$. Prove that every i in the support of $\mathbf{r}$ is a best-response to $\mathbf{c}$. That is

$$\forall i \in \text{supp}(\mathbf{r}) \underset{j \sim \mathbf{c}}{\mathbb{E}} \left(M_{i,j}\right) = \max_{i' \in \mathcal{R}} \underset{j \sim \mathbf{c}}{\mathbb{E}} \left(M_{i',j}\right)$$

Note that the analogous statement (with min in place of max) will be true for all actions in the support of $\mathbf{c}$ by symmetry, but don't spend time proving it separately.

3. Consider the two-player zero-sum game with two actions for each player described by the payoff matrix

$$\begin{bmatrix} +2 & -1 \\ -2 & +3 \end{bmatrix} \tag{1}$$

Compute a pair of equilibrium strategies $(\mathbf{r}, \mathbf{c})$ for this game. [***Hint:*** How does the property you proved in Question 2 help you find the equilibrium?]

4. The minmax theorem is the basis of one of the most widely used approaches for proving *lower bounds* on the performance of algorithms. Suppose we want to design an algorithm A that takes inputs from some finite set $\mathcal{X}$ and computes something about x. We have some real-valued *cost function* $c(x, A)$ describing

the cost of running A on input x. (This cost could be running time, space usage, some measure of "error," how "pretty" the solution is. It doesn't really make a difference.) For a given algorithm A, the *worst-case cost* is

$$\max_{\text{inputs } x} c(x, A)$$

(a) Show that if there exists a distribution $\mathbf{x}$ on inputs such that for every algorithm A

$$\mathop{\mathbb{E}}_{x \sim \mathbf{x}} (c(x, A)) \geq T$$

then for every (randomized) algorithm A, the worst-case cost is at least T.

(b) Show that if for every distribution on algorithms[1] $\mathbf{A}$, there exists an input x such that

$$\mathop{\mathbb{E}}_{A \sim \mathbf{A}} (c(x, A)) \geq T$$

then there exists a distribution on inputs $\mathbf{x}$ such that for every algorithm A

$$\mathop{\mathbb{E}}_{x \sim \mathbf{x}} (c(x, A)) \geq T$$

In other words, if every (randomized) algorithm has to have high cost on some input, then there is a single distribution on inputs such that every (randomized) algorithm has to have high expected cost on that distribution.

[1]For purposes of this question, let's assume that algorithms are defined by Boolean circuits of some finite size, so the set of algorithms is finite.