Homework 3 (due 11/23)

DS-563 / CD-543 @ Boston University

Fall 2021

Before you start...

Collaboration policy: You may verbally collaborate on required homework problems, however, you must write your solutions independently. If you choose to collaborate on a problem, you are allowed to discuss it with at most 4 other students currently enrolled in the class.

The header of each assignment you submit must include the field "Collaborators:" with the names of the students with whom you have had discussions concerning your solutions. A failure to list collaborators may result in credit deduction.

You may use external resources such as textbooks, lecture notes, and videos to supplement your general understanding of the course topics. You may use references such as books and online resources for well known facts. However, you must always cite the source.

You may **not** look up answers to a homework assignment in the published literature or on the web. You may **not** share written work with anyone else.

Submitting: Solutions should be submitted via Gradescope (entry code: BPDKV8). You are allowed to submit your solutions both in handwriting or typed. If you decide to hand-write your solutions, make sure they are as readable as possible. If you decide to submit a typed version, we suggest using LaTeX.

Grading: Whenever we ask for an algorithm (or bound), you may receive partial credit if the algorithm is not sufficiently efficient (or the bound is not sufficiently tight).

Questions (up to 10 points for each)

1. Hoeffding's inequality is a new addition to the handout with useful probabilistic inequalities on the course webpage. Familiarize yourself with it.

   Use the Chernoff bound to prove a weaker version of the Hoeffding's inequality that is mentioned in the handout. Namely, if $X_1, \ldots, X_n$ are independent random variables such that $X_i \in [0,1]$ for each $i \in [n]$ and $\epsilon \geq 0$, show that

   $$\Pr\left(\left|\sum_{i=1}^{n} X_i - E[X_i]\right| \geq \epsilon n\right) \leq 2\exp\left(-C\epsilon^2 n\right)$$

   for some constant $C > 0$.

   Hint: If $E[\sum X_i] < n/2$, consider using random variables $Y_i = 1 - X_i$, instead of X_i.

2. Let p and q be such that $0 \leq p \leq q \leq 1$. Suppose that there are two kinds of coins that look exactly the same, but coins of the first kind come up heads with probability at most p and coins of the second kind come up heads with probability at least q. Imagine that someone gives you a coin and you want to find out whether it is a coin of the first or second kind. Use Hoeffding's inequality to show that you can determine the kind of coin correctly with probability at least $1 - \delta$, for any $\delta \in (0, 1/2)$, by tossing it $O((q-p)^{-2} \log(1/\delta))$ times.

3. Consider the following greedy algorithm for finding a graph matching when the input is a stream of graph edges (with no deletions). Initially, a matching M is empty. When a new edge e arrives, if neither of its endpoints is used by edges in M, e is added to M. Otherwise, it is discarded.

   Show a stream on which this algorithm computes a matching of size exactly half the optimum (and hence, it does not obtain an approximation factor better than 2).

4. Consider a graph with at least T triangles, where a triangle is a triple of different vertices (u, v, w) in which each pair is connected with an edge. The input is provided as a stream of edges (in arbitrary order) and both edge insertions and deletions are allowed. How can you approximate the number of triangles up to a multiplicative factor of $1 + \epsilon$, for any $\epsilon \in (0, 1/2)$, using a streaming algorithm with $O(\epsilon^{-2} n^3 / T)$ space?

5. The goal in the k-center clustering problem is to find a set of at most k points—called *centers*—such that the maximum distance of any point in the input data set to the closest center is minimized. We saw how to construct a coreset for this problem that allowed for computing a multiplicative 8–approximation (i.e., a solution of cost at most 8 times the optimum) in the streaming model. Recall that the information we maintained was a set of at most k points and a distance parameter Δ.

   Consider now the distributed setting. Suppose that the data is partitioned across several machines and each of them computes this type of coreset for k-center clustering independently.

   (a) If you receive a coreset of this type (up to k points and a distance parameter) from each machine, what approximation to the k–center problem can you compute?

   (b) For the solution you found, was the distance parameter useful, or would receiving just the up to k points be sufficient?

6. Suppose that you cannot directly draw samples from a distribution, but you can instead query its probability mass function, i.e., for any $i \in [n]$, you can query the probability of i.

   (a) How many queries do you need to distinguish the uniform distribution on $[n]$ from one that is at least ϵ-far from uniform (in total variation distance)?

   (b) Suppose you have access to two probability distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ on $[n]$, i.e., you can query their probability mass functions. How many queries do you need to distinguish $\mathrm{d_{TV}}(\mathcal{D}_1, \mathcal{D}_2) = 0$ from $\mathrm{d_{TV}}(\mathcal{D}_1, \mathcal{D}_2) = 1$?

   Note: If you claim a lower bound, you do not have to make it fully formal, but your example should be very *convincing*.

7. For simplicity, assume that the probability mass function of an unknown distribution $\mathcal{D}$ on $[n]$ is non-zero for each $i \in [n]$, even if it's arbitrarily small. We consider the following stronger sampling model in which the algorithm can specify for each sample the subset from which it should be drawn. More

specifically, before receiving each sample, the algorithm specifies a non-empty $Q \subseteq [n]$ and receives an independent sample from the distribution $\mathcal{D}^\star$ in which

$$\mathcal{D}^\star_i = \begin{cases} \frac{\mathcal{D}_i}{\sum_{j \in Q} \mathcal{D}_j} & \text{if } i \in Q \\ 0 & \text{if } i \notin Q \end{cases}$$

for all $i \in [n]$.

We now develop an algorithm for testing the uniformity of $\mathcal{D}$ that uses a number of samples independent of n.

(a) Consider a distribution $\mathcal{D}$ such that $\mathrm{d_{TV}}(\mathcal{D}, \mathcal{U}) \geq \epsilon$, where $\mathcal{U}$ is the uniform distribution on $[n]$. Show that if x is drawn from $\mathcal{D}$, $\Pr[\mathcal{D}_x \geq (1 + \epsilon/2)\frac{1}{n}] \geq \epsilon/2$. **This inequality has been updated by adding "/2" in two places. You will receive full credit by solving this part and part (b) with constants greater than 2. This does not affect the rest of the problem.**

(b) Consider again a distribution $\mathcal{D}$ such that $\mathrm{d_{TV}}(\mathcal{D}, \mathcal{U}) \geq \epsilon$. Show that if y is selected uniformly at random from $[n]$, $\Pr[\mathcal{D}_y \leq (1 - \epsilon/2)\frac{1}{n}] \geq \epsilon/2$.

(c) If $\mathrm{d_{TV}}(\mathcal{D}, \mathcal{U}) \geq \epsilon$, how can you find a set of at most $O(1/\epsilon^2)$ pairs $x, y \in [n]$ such that with probability at least 0.99 for one of them, $\mathcal{D}$ restricted to $Q = \{x, y\}$ is $\Omega(\epsilon)$-far from uniform on $\{x, y\}$? How many samples do you need to achieve this?

(d) How can you distinguish between $\mathcal{D}$ uniform on $\{x, y\}$ and $\Omega(\epsilon)$-far from uniform, using $O(\epsilon^{-2} \log(1/p))$ samples with probability $1 - p$, for any $p \in (0, 1/2)$?

(e) How can you use this information to design a uniformity tester that distinguishes the uniform distribution on $[n]$ from one that is ϵ–far with probability at least 0.9, using $O(\epsilon^{-4} \log(1/\epsilon))$ samples.

(f) **(Optional, no credit)** Can you use fewer samples?

Note: If you want to desing a better tester that does not follow the approach above, feel free to do this.

8. How much time (approximately) did you spend on this homework? Was is too easy or too hard?

Note: You will not be evaluated based on your answer to this question.