**Sublinear Time Algorithms** October 26, 2022

# Homework 4

*Lecturer: Ronitt Rubinfeld* *Due Date: November 9, 2022*

1. The goal of this problem is to carefully prove a lower bound on testing whether a distribution is uniform.

   (a) For a distribution $p$ over $[n]$ and a permutation $\pi$ on $[n]$, define $\pi(p)$ to be the distribution such that for all $i$, $\pi(p)_{\pi(i)} = p_i$.

   Let $\mathcal{A}$ be an algorithm that takes samples from a black-box distribution over $[n]$ as input. We say that $\mathcal{A}$ is *symmetric* if, once the distribution is fixed, the output distribution of $\mathcal{A}$ is identical for any permutation of the distribution.

   Show the following: let $\mathcal{A}$ be an arbitrary testing algorithm for uniformity (as defined in class, a testing algorithm passes distributions that are uniform with probability at least 2/3, and fails distributions that are $\epsilon$-far in $L_1$ distance from uniform with probability at least 2/3). Suppose $\mathcal{A}$ has sample complexity at most $s(n)$, where $n$ is the domain size of the distributions. Then, there exists a symmetric algorithm that tests uniformity with sample complexity at most $s(n)$.

   (b) Define a *fingerprint* of a sample as follows: Let $S$ be a multiset of at most $s$ samples taken from a distribution $p$ over $[n]$. Let the random variable $C_i$, for $0 \leq i \leq s$, denote the number of elements that appear exactly $i$ times in $S$. The collection of values that the random variables $\{C_i\}_{0 \leq i \leq s}$ take is called the *fingerprint* of the sample.

   For example, let $D = \{1, 2, \ldots, 7\}$ and the sample set be $S = \{5, 7, 3, 3, 4\}$. Then, $C_0 = 3$ (elements $1, 2$ and $6$), $C_1 = 3$ (elements $4, 5$ and $7$), $C_2 = 1$ (element 3), and $C_i = 0$ for all $i > 2$.

   Show the following: if there exists a symmetric algorithm $\mathcal{A}$ for testing uniformity, then there exist an algorithm for testing uniformity that gets as input only the fingerprint of the sample that $\mathcal{A}$ takes.

   (c) Show that any algorithm making $o(\sqrt{n})$ queries cannot have the following behavior when given error parameter $\epsilon$ and access to samples of a distribution $p$ over a domain $D$ of size $n$:

   - if $p = U_D$, then $\mathcal{A}$ outputs "pass" with probability at least 2/3.
   - if $|p - U_D|_1 > \epsilon$, then $\mathcal{A}$ outputs "fail" with probability at least 2/3

2. Suppose an algorithm has the following behavior when given error parameter $\epsilon$ and access to samples of a distribution $p$ over a domain $D = \{1, \ldots, n\}$:

   - if $p$ is monotone, then $\mathcal{A}$ outputs "pass" with probability at least 2/3.
   - if for all monotone distributions $q$ over $D$, $|p - q|_1 > \epsilon$, then $\mathcal{A}$ outputs "fail" with probability at least 2/3

   Show that this algorithm must make $\Omega(\sqrt{n})$ queries.

   *Hint: Reduce from the problem of testing uniformity.*

3. This problem concerns testing closeness to a distribution that is entirely known to the algorithm. Though you will give a tester that is less efficient than the one seen in lecture, this method employs a useful bucketing scheme. In the following, assume that $p$ and $q$ are distributions over $D$. The algorithm is given access to samples of $p$, and knows an exact description of the distribution $q$ in advance – the query complexity of the algorithm is only the number of samples from $p$. Assume that $|D| = n$.

   (a) Let $p$ be a distribution over domain $S$. Let $S_1, S_2$ be a partition of $S$. Let

$$r_1 = \sum_{j \in S_1} p(j) \quad \text{and} \quad r_2 = \sum_{j \in S_2} p(j).$$

   Let the restrictions $p_1, p_2$ be the distribution $p$ conditioned on falling in $S_1$ and $S_2$ respectively – that is, for $i \in S_1$, let $p_1(i) = p(i)/r_1$ and for $i \in S_2$, let $p_2(i) = p(i)/r_2$. For distribution $q$ over domain $S$, let

$$t_1 = \sum_{j \in S_1} q(j) \quad \text{and} \quad t_2 = \sum_{j \in S_2} q(j),$$

   and define $q_1, q_2$ analogously. Suppose that

$$|r_1 - t_1| + |r_2 - t_2| < \epsilon_1, \ ||p_1 - q_1||_1 < \epsilon_2, \text{ and } ||p_2 - q_2||_1 < \epsilon_2.$$

   Show that $||p - q||_1 \le \epsilon_1 + \epsilon_2$.

   (b) Let $k = \lceil \log(|D|/\epsilon)/(\log(1+\epsilon)) \rceil$.
   Define *Bucket*$(q, D, \epsilon)$ as a partition $\{D_0, D_1, \ldots, D_k\}$ of $D$ with

$$D_0 = \{i \mid q(i) < \epsilon/|D|\},$$

   and for all $i \in [k]$,

$$D_i = \left\{ j \in D \;\middle|\; \frac{\epsilon(1+\epsilon)^{i-1}}{|D|} \le q(j) < \frac{\epsilon(1+\epsilon)^i}{|D|} \right\}.$$

   Show that if one considers the restriction of $q$ to any of the buckets $D_i$, then the distribution is close to uniform. In other words, show that if $q$ is a distribution over $D$ and $\{D_0, \ldots, D_k\} =$ *Bucket*$(q, D, \epsilon)$, then for any $i \in [k]$ we have

$$|q_{|D_i} - U_{D_i}|_1 \le \epsilon, \quad \|q_{|D_i} - U_{D_i}\|_2^2 \le \epsilon^2/|D_i|, \quad \text{and} \quad q(D_0) \le \epsilon$$

   where $q(D_0)$ is the total probability that $q$ assigns to set $D_0$.
   *Hint: it may be helpful to remember that* $1/(1+\epsilon) > 1 - \epsilon$.

   (c) Let $(D_0, \ldots, D_k) =$ *Bucket*$(q, [n], \epsilon)$. Prove that for each $i \in [k]$, if

$$\|p_{|D_i}\|_2^2 \le (1 + \epsilon^2)/|D_i|$$

   then $|p_{|D_i} - U_{D_i}|_1 \le \epsilon$ and $|p_{|D_i} - q_{|D_i}|_1 \le 2\epsilon$.

(d) Show that for any fixed $q$, there is an $\tilde{O}(\sqrt{n} \cdot \text{poly}(1/\epsilon))$ query algorithm $\mathcal{A}$ with the following behavior:

Given an error parameter $\epsilon$ and access to samples of a distribution $p$ over domain $D$,

- if $p = q$, then $\mathcal{A}$ outputs "pass" with probability at least 2/3.
- if $|p - q|_1 > \epsilon$, then $\mathcal{A}$ outputs "fail" with probability at least 2/3

(e) Note that the last problem part generalizes uniformity testing. As a sanity check, what does the algorithm do in the case that $q = U_D$?

4. Let $p$ be a distribution over $[n] \times [m]$. We say that $p$ is *independent* if the induced distributions $\pi_1 p$ and $\pi_2 p$ are independent, i.e., that $p = (\pi_1 p) \times (\pi_2 p)$.[1]

Equivalently, $p$ is independent if for all $i \in [n]$ and $j \in [m]$, $p(i, j) = (\pi_1 p)(i) \cdot (\pi_2 p)(j)$.

We say that $p$ is *$\epsilon$-independent* if there is a distribution $q$ that is independent such that $|p - q|_1 \leq \epsilon$. Otherwise, we say $p$ is *not $\epsilon$-independent* or is *$\epsilon$-far from being independent.*

Given access to independent samples of a distribution $p$ over $[n] \times [m]$, an *independence tester* outputs "pass" if $p$ is independent, and "fail" if $p$ is $\epsilon$-far from independent (with error probability at most 1/3).

(a) Prove the following: let $A, B$ be distributions over $S \times T$. If $|A - B| \leq \epsilon/3$ and $B$ is independent, then $|A - (\pi_1 A) \times (\pi_2 A)| \leq \epsilon$.

*Hint: It may help to first prove the following. Let $X_1, X_2$ be distributions over $S$ and $Y_1, Y_2$ be distributions over $T$. Then $|X_1 \times Y_1 - X_2 \times Y_2|_1 \leq |X_1 - X_2|_1 + |Y_1 - Y_2|_1$.*

(b) Give an independence tester which makes $\tilde{O}((nm)^{2/3} \cdot \text{poly}(1/\epsilon))$ queries. (You may use the $L_1$ tester mentioned in class, which uses $\tilde{O}(n^{2/3} \cdot \text{poly}(1/\epsilon))$ samples, without proving its correctness.)

[1] For a distribution $A$ over $[n] \times [m]$, and for $i \in \{1, 2\}$, we use $\pi_i A$ to denote the distribution you get from the procedure of choosing an element according to $A$ and then outputting only the value of the the $i$-th coordinate.