6.842 Randomness and Computation April 15, 2020

Homework 5

Lecturer: Ronitt Rubinfeld *Due Date: April 23, 2020*

Homework guidelines: You may work with other students, as long as (1) they have not yet solved the problem, (2) you write down the names of all other students with which you discussed the problem, and (3) you write up the solution on your own. No points will be deducted, no matter how many people you talk to, as long as you are honest. If you already knew the answer to one of the problems (call these "famous" problems), then let me know that in your solution writeup – it will not affect your score, but will help me in the future. It's ok to look up famous sums and inequalities that help you to solve the problem, but don't look up an entire solution.

1. *Dictator functions*, also called *projection functions*, are the functions mapping $\{+1,-1\}^n$ to $\{+1,-1\}$ of the form $f(x) = x_i$ for i in $[n]$.

   Consider the following test for whether a function f is a dictator: Given parameter δ, the test chooses $x, y, z \in \{1,-1\}^n$ by first choosing x, y uniformly from $\{1,-1\}^n$, next choosing w by setting each bit w_i to -1 with probability δ and $+1$ with probability $1-\delta$ (independently for each i), and finally setting z to be $x \circ y \circ w$, where $\circ$ denotes the bitwise multiply operation. Finally, the test accepts if $f(x)f(y)f(z) = 1$ and rejects otherwise.

   (a) Show that the probability that the test accepts is $\frac{1}{2} + \frac{1}{2}\sum_{s\subseteq[n]}(1-2\delta)^{|S|}\hat{f}(S)^3$.

   (b) Show that if f is a dictator function, then f passes with probability at least $1-\delta$.

   (c) Show that if f passes with probability at least $1-\epsilon$ then there is some S such that $\hat{f}(S)$ is at least $1-2\epsilon$ and such that f is ϵ-close to χ_S.

   (d) Why isn't this enough to give a dictator test? (i.e., what nondictators might pass?) Give a simple fix.

2. Consider the following graph-based linearity test. Let $G = (V, E)$ be a graph on $k = |V|$ vertices and let $f : \{\pm 1\}^n \to \{\pm 1\}$ be given.

   - Sample $x_1, \ldots, x_k \in_R \{\pm 1\}^n$
   - Query $f(x_i)$ for all $i \in [k]$ and $f(x_i \odot x_j)$ for all $(i,j) \in E$ where $x_i \odot x_j$ denotes the coordinate-wise product of x_i and x_j.
   - Accept if and only if $f(x_i)f(x_j) = f(x_i \odot x_j)$ for all $(i,j) \in E$.

   Note that if f is linear, then this graph-test always accepts.

   (a) Prove that: For all $S \subseteq E$ such that $S \neq \emptyset$, then

   $$E[\Pi_{(i,j)\in S} f(x_i)f(x_j)f(x_ix_j)] \leq \max_\alpha |\hat{f}(\alpha)|$$

   (b) Conclude that the probability that the above graph-test accepts is at most

   $$\frac{1}{2^{|E|}} + \max_\alpha |\hat{f}(\alpha)|$$

3. **Distribution Free Learning:** The goal of this problem is to present an algorithm that learns an unknown conjunction $f : \{0,1\}^n \to \{0,1\}$, for example:

$$f(\mathbf{x}) = x_i \wedge \neg x_j \wedge x_k \wedge \cdots$$

In this problem, the distribution $\mathcal{D}$ of $\mathbf{x} \in_{\mathcal{D}} \{0,1\}^n$ is *unknown*. The goal is to obtain (with probability at least $1-\delta$), a conjunction $h : \{0,1\}^n \to \{0,1\}$ such that:

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}}\left[f(\mathbf{x}) \neq h(\mathbf{x})\right] < \epsilon \tag{1}$$

Present an algorithm that takes in $m = poly(n, 1/\epsilon, 1/\delta)$ samples, and outputs a conjunction h in $poly(n, 1/\epsilon, 1/\delta)$ time, such that Equation 1 is satisfied with probability at least $1-\delta$.

Note that in class, we considered a special case of this problem, where $\mathcal{D}$ was the uniform distribution.

4. **Occam Learning of Decision Lists:** We are given a function $f : \{0,1\}^n \to \{0,1\}$ which is known to be a *decision list* (a special case of the decision trees we studied in class): Concretely, f must be constructed from a sequence of decisions, each of which relies on only one literal (see Figure 1).

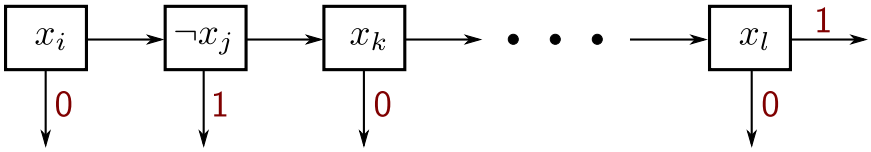

Figure 1: A sample decision list function.

(a) Starting with a set of samples S show how to find a *consistent* decision list h, such that for all $\mathbf{x} \in S$, we have $h(\mathbf{x}) = f(\mathbf{x})$. Note that there may be more than one decision list that is consistent with the data, but proof of Occam's razor shows that all *consistent* decision lists must be close to f. What value of $|S|$ is necessary to ensure (ϵ, δ) PAC learning?

(b) Present an *efficient* (runtime polynomial in n) algorithm to find a decision list h such that, with probability at least $1-\delta$:

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}}\left[f(\mathbf{x}) \neq h(\mathbf{x})\right] < \epsilon$$

Hint: Start with an empty decision list, and gradually add more literals, along with their *consequences* (the red output).