Instructor: Rocco A. Servedio

Computer Science 4252: Introduction to Computational Learning Theory
Problem Set #3 Fall 2023

Due 11:59pm Wednesday, November 1, 2023

See the course Web page for instructions on how to submit homework. **Important:** To make life easier for the TAs, **please start each problem on a new page.**

Problem 1 Show that the following two variants on the definition of PAC learnability for a concept class C are equivalent (i.e. show that any concept class which satisfies the first definition also satisfies the second, and vice versa).

[1] The learning algorithm is given access to a single oracle $EX(c, \mathcal{D})$ for random examples of the target concept c. Each time $EX(c, \mathcal{D})$ is called it independently selects an instance x at random according to the distribution $\mathcal{D}$, and returns $(x, c(x))$. For all $0 < \delta, \varepsilon < 1$, for all distributions $\mathcal{D}$, for all target concepts $c \in C$, the algorithm must (in time polynomial in all the relevant parameters) output a hypothesis h that with probability at least $1 - \delta$ has error less than ε on random examples drawn according to $\mathcal{D}$. (This is just the usual definition of PAC learning.)

[2] The learning algorithm is given access to two oracles $POS(c, \mathcal{D}^+)$ and $NEG(c, \mathcal{D}^-)$. These return random positive and negative examples (respectively) for the target concept drawn independently according to the distributions $\mathcal{D}^+$ (a distribution over positive examples of c) and $\mathcal{D}^-$ (a distribution over negative examples of c). For all $0 < \delta, \varepsilon < 1$, for all target concepts $c \in C$, for all distributions $\mathcal{D}^+$ over the positive examples for c, for all distributions $\mathcal{D}^-$ over the negative examples for c, the algorithm must in time polynomial in all the relevant parameters output a hypothesis that with probability at least $1 - \delta$ has error less than ε on both random positive examples drawn according to $\mathcal{D}^+$ and random negative examples drawn according to $\mathcal{D}^-$.

Problem 2 Suppose that algorithm A is a PAC learning algorithm for concept class C which has running time and sample complexity $\text{poly}(1/\varepsilon, 1/\delta)$. Show that there is a PAC learning algorithm A' for concept class C which has running time and sample complexity $\text{poly}(1/\varepsilon, \log(1/\delta))$. (You may assume that any hypothesis h that A outputs when run with parameters ε, δ can be evaluated on any example x in time $\text{poly}(1/\varepsilon, 1/\delta)$.)

Problem 3 You are given a coin which has an unknown probability $p \in (0, 1)$ of coming up "heads" when it is tossed. You would like to get a coarse estimate of the value of p.

(a) Consider the following simple procedure: Toss the coin until you get "heads" for the first time and output "$\widehat{p} = 1/k$" where k is the number of tosses you made. Show that this procedure is indeed a somewhat-decent coarse estimator in the following sense: with probability at least 89%, your procedure outputs a value $\widehat{p}$ such that

$$\frac{\widehat{p}}{10} \leq p \leq 10\widehat{p}.$$

(b) Explain how the procedure from part (a) can be used to obtain an improved procedure that with probability at least $1 - \delta$ outputs a value $\widehat{p}$ with $\frac{\widehat{p}}{10} \leq p \leq 10\widehat{p}$.

Problem 4 Let S be a subset of $\{1, \ldots, n\}$. The *parity function corresponding to* S is the Boolean function $PAR_S : \{0,1\}^n \to \{0,1\}$,

$$PAR_S(x_1, \ldots, x_n) = \sum_{i \in S} x_i \mod 2$$

which counts whether the number of input bits in coordinates indexed by S is odd or even. Parity is sometimes written "$\oplus$" to denote exclusive-Or. For example, for the set $S = \{1, 3, 4\}$ the function $PAR_S(x) = x_1 + x_3 + x_4 \mod 2 = x_1 \oplus x_3 \oplus x_4$ computes the parity of the subset $S = \{x_1, x_3, x_4\}$, and we have $PAR_S(0010) = 1$, $PAR_S(1010) = 0$.

The class of *parity functions* $\mathcal{P}$ includes all functions that can be described in this way. Formally,

$$\mathcal{P} = \{f \mid f(x) = PAR_S(x), \text{ where } S \subseteq [n]\}.$$

Show that the concept class of parity functions $\mathcal{P}$ is PAC learnable by describing an algorithm and proving that it PAC learns this class. Your algorithm should have sample complexity and running time $\text{poly}(n, 1/\varepsilon, \log(1/\delta))$.

Problem 5 In this problem you'll consider a weakening of the notion of a "consistent hypothesis finder" and show that it is still strong enough for PAC learning. (As in the theorems about consistent hypothesis finders proved in class, you should assume in this problem that $\mathcal{H}$ is a finite hypothesis class.)

Let us say that an algorithm B is an "sort-of-decent hypothesis finder" for $\mathcal{C}$ using $\mathcal{H}$ if it has the following performance guarantee: Given any sample of m examples $(x^1, c(x^1)), \ldots (x^m, c(x^m))$ labeled according to some $c \in \mathcal{C}$, B outputs a hypothesis $h \in \mathcal{H}$ that is correct on at least $m - m^{1/4}$ of the m examples.

Show that a sort-of-decent hypothesis finder can be used to construct a PAC learning algorithm for $\mathcal{C}$ that uses $\text{poly}(\frac{1}{\varepsilon}, \ln|\mathcal{H}|, \ln(1/\delta))$ samples. Justify your answer; you need not try for the best possible bound, any $\text{poly}(\frac{1}{\varepsilon}, \ln|\mathcal{H}|, \ln(1/\delta))$ bound is fine.