Problem 1  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 $H$ is a finite hypothesis class.)

Let us say that an algorithm $B$ is an “sort-of-decent hypothesis finder” for $C$ using $H$ 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 $c \in C$, $B$ outputs a hypothesis $h \in H$ 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 $C$ that uses $\text{poly}\left(\frac{1}{\varepsilon}, \ln |H|, \ln(1/\delta)\right)$ samples. Justify your answer; you need not try for the best possible bound, any $\text{poly}\left(\frac{1}{\varepsilon}, \ln |H|, \ln(1/\delta)\right)$ bound is fine.

Problem 2  Given a distribution $D$ and a target concept $c$, the distribution $D^+$ is defined to be the distribution $D$ restricted to the positive examples $\{x : c(x) = 1\}$.

A concept class $C$ is said to be learnable from positive examples only if there is an algorithm $A$ with has the following property: For any $\varepsilon, \delta > 0$, any distribution $D$ and any target concept $c \in C$, given $\varepsilon, \delta$ and access to the oracle $\text{EX}(c, D^+)$, algorithm $A$ outputs a hypothesis $h \in H$ such that $\Pr_{x \in D}[h(x) \neq c(x)] \leq \varepsilon$ with probability $1 - \delta$. (In other words, the algorithm PAC learns successfully with respect to distribution $D$ given only positive examples from $D^+$; the algorithm given in class for learning intervals is an example of such an algorithm.)

Suppose that concept class $C$ is PAC learnable from positive examples only by some learning algorithm $A$. Show that then algorithm $A$ is also a PAC learning algorithm for the concept class $\{c_1 \cap c_2 | c_1, c_2 \in C\}$.

Problem 3  Do Exercise 1.5 of the Kearns and Vazirani textbook. In other words, let $C$ be a concept class over $\{0,1\}^n$ which is equipped with a notion of the size of concepts in $C$. Suppose that there is an efficient PAC learning algorithm $A$ for $C$ which is given as input $\varepsilon, \delta$, and $\text{size}(c)$ where $c \in C$ is the unknown target concept. Show that then there is an efficient PAC learning algorithm $A'$ for $C$ which is given only $\varepsilon, \delta$ as input. To be precise, your algorithm should have the property that given $\varepsilon, \delta$ as input parameters, with probability at least $1 - \delta$ the algorithm (i) runs for at most $\text{poly}(n, 1/\varepsilon, 1/\delta, \text{size}(c))$ time steps and (ii) outputs an $\varepsilon$-accurate hypothesis.

1
Problem 4  Throughout this problem the domain is some finite set $X$ and the concept class $C$ is the class of all $2^{|X|}$ Boolean functions mapping $X$ to $\{0, 1\}$.

Consider a procedure with the following property: given a sample $S$ of $m$ labelled examples $(x^1, c(x^1)), \ldots, (x^m, c(x^m))$, where each $x^i$ belongs to $X$ and $c$ is some concept in $C$, the algorithm returns a lookup table with these examples hard-coded into it; i.e. the hypothesis is “given $x$, if $x = x^i$ for some $i = 1, \ldots, m$ then return the corresponding value $c(x^i)$ seen in the sample, otherwise return 0.”

(i) True or false: if $|X| = n$ then this procedure yields a PAC learning algorithm with sample complexity $m = \text{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$. (Justify your answer as thoroughly as you can.)

(ii) True or false: if $|X| = 2^n$ then this procedure yields a PAC learning algorithm with sample complexity $m = \text{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$. (Justify your answer as thoroughly as you can.)

Problem 5  Let $S$ be a subset of $\{1, \ldots, n\}$. The parity function corresponding to $S$ is the Boolean function $\text{PAR}_S : \{0, 1\}^n \rightarrow \{0, 1\}$,

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

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

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

$$\mathcal{P} = \{ f \mid f(x) = \text{PAR}_S(x), \text{ where } S \subseteq X \}.$$ 

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 $\text{poly}(n, 1/\varepsilon \log(1/\delta))$. 
