Instructor: Rocco Servedio
Teaching Assistant: Andrew Wan

Computer Science 4252: Introduction to Computational Learning Theory
Problem Set #3 Spring 2006

Due 5:00pm Monday, March 13, 2006

Problem 1
(i) Let $\mathcal{D}$ be an arbitrary probability distribution over $\{1, 2, \ldots, 2^n\}$. Let k be the maximum value obtained in m independent draws from $\mathcal{D}$. For what value of m do we have that with probability at least $1 - \delta$, $\Pr_{i \in \mathcal{D}}[i > k] < \epsilon$ (i.e. the total weight which distribution $\mathcal{D}$ puts on values $i > k$ is less than ϵ)?

(ii) Again let $\mathcal{D}$ be an arbitrary probability distribution over $\{1, \ldots, 2^n\}$. Let $r \in \{1, \ldots, 2^n\}$ be such that $\Pr_{i \in \mathcal{D}}[i > r] = \epsilon$. Consider m independent draws from $\mathcal{D}$. For what value of m do we have that with probability at least $1 - \delta$, at least an $\frac{\epsilon}{2}$ fraction of the m draws give values which are larger than r?

Problem 2 Let us say that an algorithm A efficiently "perhaps learns" a concept class C if for all $c \in C$, for all ϵ, for all distributions $\mathcal{D}$, if A is given access to an oracle $EX(c, \mathcal{D})$ for c, then A runs in time polynomial in $1/\epsilon$ and size(c) and with probability at least $3/4$ outputs a hypothesis h such that $Pr_{x \in \mathcal{D}}[c(x) \neq h(x)] \leq \epsilon$. (We have simply switched δ in the standard definition of efficient PAC learning to be the constant $1/4$, and omitted mention of n since it's not relevant for this problem.)

Show that if there is an efficient "perhaps learning" algorithm for a concept class C, then there is an efficient PAC learning algorithm for C.

Problem 3 Recall that the conversion from an online algorithm with mistake bound m to a PAC algorithm given in class works as follows: "Run A on a sequence of examples each drawn independently from $\mathcal{D}$. If hypothesis h ever survives $\frac{1}{\epsilon} \log \frac{m+1}{\delta}$ consecutive examples without making a mistake, stop and output h."

Now suppose that you have an online algorithm A with some finite mistake bound m, but you don't know what the value of m is. Explain how you can obtain a PAC algorithm from A. What is the best sample complexity (in terms of m, ϵ, and δ) that you can achieve for your PAC algorithm?

Problem 4 A parity function tests whether the parity of some subset $S \subseteq X$ of the input literals $X = \{x_1, \ldots, x_n\}$ is odd or even. In other words, if the number of the variables in S that have value 1 is odd then $f(x) = 1$, and if the number is even then $f(x) = 0$. For example, the function $f = x_1 \oplus x_3 \oplus x_4$ computes the parity of the subset $S = \{x_1, x_3, x_4\}$, and $f(0010) = 1$, $f(1010) = 0$.

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

$$\mathcal{P} = \{f \mid f = \oplus_{y \in S} y, \text{ where } S \subseteq X\}.$$

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. **Hint:** The parity operation is known as the XOR operation which computes addition modulo 2.

Problem 5

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 $0 < \delta, \epsilon < 1$, 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 that 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 $0 < \delta, \epsilon < 1$, 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 that ϵ on both random positive examples drawn according to $\mathcal{D}^+$ and random negative examples drawn according to $\mathcal{D}^-$.

Problem 6 Write a brief (1 paragraph) project proposal. This should be emailed directly to Andrew (`atw12@columbia.edu`) and need not be submitted with the rest of your homework set.

This proposal should be a rough description of what topic you will be investigating, what you plan to do, and what sources (books, papers, etc) you anticipate using at this point. The point is to get you thinking about what your project topic will be and to start you on the process of identifying useful sources.