Instructor: Rocco A. Servedio

Computer Science 4252: Introduction to Computational Learning Theory
Problem Set #2 Fall 2023

Due 11:59pm Wednesday, October 11, 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 Let $\mathcal{C}_W$ denote the class of all linear threshold functions $w \cdot x \geq \theta$ over the domain $\{0,1\}^n$ such that each w_i is a nonnegative integer and $\sum_{i=1}^n w_i \leq W$.

(i) Describe how you would use the Winnow2 algorithm to learn this class (you may assume the value of W is known to the algorithm). Try to get a mistake bound of $O(W^3 \cdot \log n)$.

(ii) Now give a different analysis achieving a better asymptotic bound, of $O(W^2 \log n)$.

(If you solve part (ii), of course that suffices as a solution to part (i) as well, and there's no need to write out a separate solution to part (i). But working on part (i) first may help you towards part (ii).)

Problem 2 In this problem you'll show that the Perceptron algorithm can be quite inefficient even for learning some simple linear threshold functions over the Boolean hypercube.

(i) Let $f(x,y)$ be the following function: given two n-bit strings $x, y \in \{-1,1\}^n$, let i be the first index in $\{1,\ldots,n\}$ such that $x_i \neq y_i$. The value $f(x,y)$ is 1 if $x_i = 1, y_i = -1$ and is -1 if $y_i = 1, x_i = -1$ (and is 1 if $x = y$). Show that f is a linear threshold function.

(ii) Suppose that $\text{sign}(w_1x_1 + \cdots + w_nx_n + v_1y_1 + \cdots + v_ny_n - \theta)$ is a linear threshold function representation of $f(x,y)$ where each w_i and each v_i is an integer.[1] Show that we must have $\sum_{i=1}^n |w_i| + |v_i| = \Omega(2^n)$. (Hint: Consider the possible "behaviors" of the LTF as a function of y for different fixed settings of the x-variables.)

(iii) Use part (ii) to give a lower bound on the number of mistakes that the Perceptron algorithm may make when it is used to learn some linear threshold function over $\{-1,1\}^n$.

Problem 3 Recall that a "feature expansion" is a mapping $\Phi : \mathbb{R}^n \to \mathbb{R}^N$. The kernel function K corresponding to Φ is $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined as $K(x,y) = \Phi(x) \cdot \Phi(y)$.

[1] Recall that the function sign(z) takes value 1 if $z \geq 0$ and value -1 otherwise.

(a) Let $\Phi : \{0,1\}^n \rightarrow \{0,1\}^{\sum_{i=0}^{k} \binom{n}{i}}$ be the feature expansion which has one feature for every possible monotone conjunction of length at most k over the input variables $x_1, \ldots, x_n$. For example, if $n = 3$ and $k = 2$ then $\Phi(x_1, x_2, x_3)$ equals

$$(1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3).$$

(Note that the empty conjunction is equivalent to the always-true function, i.e. the constant-1 function).

Show that the kernel function $K(x,y)$ for this Φ can be computed in time poly(n).

(ii) Let $S \subseteq \{1, 2, \ldots, n\}$. The *parity function* $PAR_S : \{0,1\}^n \rightarrow \{0,1\}$ tests whether the parity of the Boolean variables corresponding to the elements of S is odd or even. In other words, if the number of variables in S that have value 1 is odd then $PAR_S(x) = 1$, and if the number is even then $PAR_S(x) = 0$; or in more mathematical notation, $PAR_S(x) = (\sum_{i \in S} x_i) \mod 2$.

For example, the function $f = PAR_{\{1,3,4\}}$ computes the parity of the subset $S = \{x_1, x_3, x_4\}$, and we have $f(0010) = 1$, $f(1010) = 0$.

The class of *parity functions* $\mathcal{P}$ consists of all 2^n functions over $\{0,1\}^n$ that can be described in this way. Formally,

$$\mathcal{P} = \{PAR_S : S \subseteq \{1, 2, \ldots, n\}\}.$$

Let $\Phi : \{0,1\}^n \rightarrow \{0,1\}^{\binom{n}{0} + \cdots + \binom{n}{k}}$ be the feature expansion which has one feature for every possible parity function over at most k of the input variables $x_1, \ldots, x_n$. Show that the kernel function $K(x,y)$ for this Φ can be computed in poly(n) time.

Problem 4 Suppose you are running the Randomized Halving Algorithm to learn an unknown target concept c that belongs to a known finite concept class $\mathcal{C}$ with N concepts (i.e. $|\mathcal{C}| = N$) in an "oblivious adversary" setting (the target concept and sequence of examples that will be given to you are fixed once and for all ahead of time). We showed in class that the expected number of mistakes that the R.H.A. makes is at most $\frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{N}$, which is at most $\ln N$. It's natural to want a bound that gives us more information - in particular, it would be nice if we knew something about *how likely* it is that the R.H.A. makes a "large" number of mistakes.

(i) True or false: For every concept class $\mathcal{C}$ with $|\mathcal{C}| = N$, every target concept $c \in \mathcal{C}$, and every sequence of examples $x^1, x^2, \ldots,$ the probability that the R.H.A. makes more than $100 \ln N$ mistakes is at most $1/\text{poly}(N)$. (Justify your answer.)

(ii) True or false: There is a concept class $\mathcal{C}$ with $|\mathcal{C}| = N$, a target concept c, and a sequence of examples $x^1, x^2, \ldots$ such that the probability that the R.H.A. makes more than $100 \ln N$ mistakes is at least $1/\text{poly}(N)$. (Justify your answer.)

Problem 5

(i) Consider the instance space $X = \{1, 2, \ldots, 999\}$. Let $\mathcal{C}$ be a concept class consisting of 10 concepts, c_0 through c_9. A number n in X is an element of c_i if the normal decimal representation of n contains the digit i. So, for example, the number "778" is an element of c_7 and c_8.

What is the VC dimension of $\mathcal{C}$? Justify your answer.

(ii) Now consider the domain $X = \Re$ (the real numbers). Fix a positive integer k, and let C_k be the concept class consisting of all unions of k closed intervals, i.e. if $k = 3$ then a concept in C might be $c = [-3, -2] \cup [1, 4] \cup [5, 100]$ (here we are viewing the concept c as being a subset of X.) Determine the exact value of the VC dimension of C_k.