Coursework	**Spring 2004**
Computational Learning Theory (150CLT)	**Roni Khardon**
Department of Computer Science, Tufts University	

Assignment 4

This assignment is due back by Tuesday May 4. You can hand it in the main CS/ECE office in my mailbox.

Note: Make sure you write and justify all your statements in a precise, formal way.

Problem 1

Suppose the leaves of a 1-decision list are replaced by other 1-decision lists. Call such a structure a rank-2 1-decision list. Describe a polynomial time Occam algorithm for learning rank-2 1-decision lists with hypotheses in rank-2 1-decision lists and prove its correctness.

Problem 2

(i) Let $\mathcal{C}$ be the class of conjunctions over n Boolean variables. Show that the size of any set shattered by $\mathcal{C}$ is at most n.

(ii) Let $\mathcal{D}_k$ be the class of k-term DNF expressions over n Boolean variables. Show that there is a set of size $k(n - \log k)$ which is shattered by $\mathcal{D}_k$.

Problem 3

An axis parallel rectangle (with two sides parallel to the x-axis and two sides parallel to the y-axis) represents a concept over the domain $\Re^2$ where the positive examples are those points inside the rectangle (see Section 1.1 of the text by Kearns and Vazirani).

The class NDAPR_k of nested difference of k axis parallel rectangles is composed of concepts expressible as $(A_1 \setminus (A_2 \setminus \ldots (A_{k-1} \setminus A_k) \ldots)$ where all A_i are axis parallel rectangles.

Show that the VC dimension of NDAPR_k is exactly $4k$.

Hint: You may want to use induction; the base case appears in the text.

Problem 4

Describe a polynomial time PAC learning algorithm for NDAPR_k and prove its correctness.

Hint: By Problem 4 it suffices to find a consistent hypothesis in NDAPR_k.

Problem 5

For $x \in \{0,1\}^n$ let $t(x) = \wedge_{x_i=1} x_i$. For example for $x = 101$, $t(x) = x_1 x_3$. For two assignments $x, y \in \{0,1\}^n$, $x \wedge y$ denotes their bitwise AND. For example for $x = 101$ and $y = 110$, $x \wedge y = 100$. Show that the following algorithm efficiently learns the class of monotone DNF expressions and give bounds for the numbers of equivalence and membership queries.

1. Initialize $S = \emptyset$ and $h = 0$.
2. While $EQ(h)$ provides a (positive) counterexample x
   (a) For each element s in S do:
      - if $MQ(s \wedge x)$ returns "positive" then replace s with $s \wedge x$ and quit for loop.

   (b) if no element was replaced then add x to S.
   (c) Let $h = \vee_{s \in S} t(s)$
3. Output h

Hint: Argue that two elements of S cannot satisfy the same term of the target. Then try to quantify how you make "progress" after every counter example.

Problem 6 (extra credit)

Define a notion of reducibility appropriate for learning from Equivalence and Membership Queries. Using this notion show that if C^1 reduces to C^2 and C^2 is efficiently learnable from Equivalence and Membership Queries then C^1 is efficiently learnable from Equivalence and Membership Queries. Note that you would want this notion of reducibility to be useful in the next problem.

Problem 7 (extra credit)

A read-3 DNF formula over $\{0,1\}^n$ is a DNF formula using literals in $x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}$ in which every variable appears at most 3 times (whether negated or unnegated). For example $x_1 x_2 \overline{x_3} x_5 \vee \overline{x_2} x_4 \overline{x_5} \vee x_3 x_4 x_5 \vee \overline{x_1}\ \overline{x_2}\ \overline{x_4}$ is read-3 but $x_1 x_2 \overline{x_3} x_5 \vee \overline{x_2} x_4 \overline{x_5} \vee x_3 x_4 x_5 \vee \overline{x_1}\ \overline{x_2}\ \overline{x_5}$ is not (since x_5 appears 4 times). The class of read-3 DNF is composed of the union of such formulae for $n \geq 1$.

Let $p()$ be any polynomial and let C be a sub-class of DNF formulae. We say that C is $p()$-bounded if every formulae $c \in C$ over $\{0,1\}^n$ has at most $p(n)$ terms. Show that if C is $p()$-bounded then it reduces to read-3 DNF (using reducibility of the previous question).