CIS 625 PROBLEM SET 1

Professor Kearns

Due Wednesday September 29, 2021 on Gradescope (code ERP5VV)

Problem 1.

In lecture, we proved that axis-aligned rectangles in the real plane are PAC learnable and briefly sketched the generalization to axis-aligned rectangles in n-dimensional real space $\mathbb{R}^n$. Formally, for the concept class $\mathscr{C}$ of axis-aligned rectangles of dimension n, there is an algorithm L that takes in samples $\langle x, c(x)\rangle$ for $x \in \mathbb{R}^n$ distributed according to a distribution F and $c \in \mathscr{C}$, error function ε, and outputs some hypothesis class $h \in \mathscr{C}$ that with probability $1-\delta$ has error $\varepsilon(h) < \epsilon$ with runtime $\text{poly}(\frac{1}{\varepsilon}, \frac{1}{\delta}, n)$ for $\epsilon, \delta > 0$.

For this problem, complete the generalization onto n-dimensional real space carefully — show that this class of concepts is PAC learnable by precisely describing and analyzing a learning algorithm L. Be sure to show all parts of the PAC definition, including running time of your algorithm and sample size analysis. Note that here you want to use the part of the PAC definition that allows your algorithm's running time and sample size to depend *polynomially* on n.

Problem 2.

Consider the class of concepts known as *unions of intervals*. Here the domain or input space X is simply the real line. Each concept c in the class can be described by a union of n intervals of the real line:

$$c = [a_1, b_1] \cup [a_2, b_2] \cup [a_3, b_3] \ldots \cup [a_n, b_n]$$

In other words, this is the set of positive examples of c and all other points are negative examples. Here $\cup$ is set union, and $[a_i, b_i]$ denotes all real numbers x such that $a_i \leq x \leq b_i$. So each concept is simply the union of n non-overlapping intervals of the real line, and let's say there are n or fewer of them.

Show that this class is PAC learnable, again by carefully describing and analyzing a learning algorithm and showing that it meets all parts of the PAC definition. Note that here you want to use the part of the PAC definition that allows your algorithm's running time and sample size to depend *polynomially* on n.

Problem 3.

Recall that for rectangles in the real plane, we showed that for any set of labeled examples, there is a "unique tightest fit" to the positive examples — i.e. a rectangle that includes all of the positive examples, none of the negative examples, and whose area is as small as possible.

Now consider rectangles that include all of the positives, none of the negatives, and whose area is as large as possible.

Show that these may not be unique, by demonstrating a sample of points generated by a target rectangle, in which there are 2 different rectangles consistent with the sample and with maximal area. For this problem, a clearly annotated picture suffices.

Problem 4.

Recall from lecture that we showed PAC learning 3-term DNF (of the form $T_1 \vee T_2 \vee T_3$ where T_1, T_2, T_3 are conjunctions over literals) is hard, in the sense that a PAC algorithm for it would imply a randomized polynomial time algorithm for graph 3-coloring, a notoriously hard (and formally NP-complete problem). Here we consider the problem of PAC learning **general** DNF formulas, whose status remains unresolved almost 40 years after first being considered!

The problem is as follows: the target function is a logical formula of the form

$$T_1 \vee T_2 \vee T_3 \ldots \vee T_m.$$

Here each T_i is a conjunction over n boolean variables $x_1, x_2, \ldots x_n$ and their negations. So instead of just 3 terms, the target function has some unknown and arbitrary number of terms, m.

Note that m could be much larger than n, since the number of possible conjunctions is 3^n (recall for each x_i we can either not include it, include it, or include its negation). This is an example of where the PAC definition allows the learning algorithm to run in time polynomial not only in n, but in m as well (as well as $\frac{1}{\epsilon}, \frac{1}{\delta}$).

As I said above, we *don't know* if such general DNF formula are PAC learnable, and I'm not going to ask you to resolve that here ☺. Instead you will be asked to show that certain restrictions on the problem do not make it any easier.

For this problem do the following:

(1) Consider monotone DNF formulas, in which no variable x_j appears negated in any of the terms T_i. Show that if monotone DNF is PAC learnable, so is general DNF.

(2) Consider read-once DNF formulas, in which each variable x_j appears at most once in the entire DNF — so if x_j or its negation appear in one term, they cannot appear in any other term. Show that if read-once DNF is PAC learnabl, so is general DNF.

To do this, you will need to give a reduction. In the case of the first problem, assume that you are given access to a PAC learning algorithm L for monotone DNF, prove you can use it in a subroutine for PAC learning general DNF. So you need to figure out how to simulate the learning of a monotone DNF for L in a manner that lets you learn a general DNF. You can assume that L returns a general DNF formula as its hypothesis.

Also, to make things a bit easier, you may assume that your algorithm for general DNF is given m as an input — i.e. you are told the number of terms/conjunctions in the target DNF. I'll have you fix that in the next problem ☺.

Hint: your reduction/simulation will need to use the fact that L must work for any distribution.

Problem 5.

Suppose you are given a learning algorithm L for PAC-learning general DNF, but L needs the number of terms/conjunctions m in the target as an input. Show that you can use L to construct an algorithm L' for PAC learning general DNF that doesn't know m.

Hint: the solution to this problem is sometimes known as the "doubling trick".