CS 391L: Machine Learning **Fall 2020**

Homework 1 - Theory

Lecture: Prof. Adam Klivans

Keywords: Boolean functions, mistake bounds, PAC learning

Instructions: Please either typeset your answers (LaTeX recommended) or write them very clearly and legibly and scan them, and upload the PDF on edX. Legibility and clarity are critical for fair grading.

1. Let D be an arbitrary distribution on the domain $\{-1,1\}^n$, and let $f, g : \{-1,1\}^n \to \{-1,1\}$ be two Boolean functions. Prove that

$$\mathbb{P}_{x\sim D}[f(x) \neq g(x)] = \frac{1 - \mathbb{E}_{x\sim D}[f(x)g(x)]}{2}.$$

   Would this still be true if the domain were some other domain (such as $\mathbb{R}^n$, where $\mathbb{R}$ denotes the real numbers, with say the Gaussian distribution) instead of $\{-1,1\}^n$? If yes, justify your answer. If not, give a counterexample.

2. Let f be a decision tree with t leaves over the variables $x = (x_1, \ldots, x_n) \in \{-1,1\}^n$. Explain how to write f as a multivariate polynomial $p(x_1, \ldots, x_n)$ such that for every input $x \in \{-1,1\}^n$, $f(x) = p(x)$. (You may interpret -1 as FALSE and 1 as TRUE or the other way round, at your preference.) *(Hint: try to come up with an "indicator polynomial" for every leaf, i.e. one that evaluates to the leaf's value if x is such that that path is taken, and 0 otherwise.)*

3. Compute a depth-two decision tree for the training data in table 1 using the Gini function, $C(a) = 2a(1-a)$ as described in class. What is the overall accuracy on the training data of the tree?

X	Y	Z	Number of positive examples	Number of negative examples
0	0	0	10	20
0	0	1	25	5
0	1	0	35	15
0	1	1	35	5
1	0	0	5	15
1	0	1	30	10
1	1	0	10	10
1	1	1	15	5

Table 1: decision tree training data

4. Suppose the domain X is the real line, $\mathbb{R}$, and the labels lie in $Y = \{-1,1\}$, Let $\mathcal{C}$ be the concept class consisting of simple threshold functions of the form h_θ for some $\theta \in \mathbb{R}$, where $h_\theta(x) = -1$ for all $x \leq \theta$ and $h_\theta(x) = 1$ otherwise. Give a simple and efficient PAC learning algorithm for $\mathcal{C}$ that uses only $m = O(\frac{1}{\epsilon}\log\frac{1}{\delta})$ training examples to output a classifier with error at most ϵ with probability at least $1 - \delta$.

5. In this problem we will show that mistake bounded learning is stronger than PAC learning, which should help crystallize both definitions. Let $\mathcal{C}$ be a function class with domain $X = \{-1,1\}^n$ and labels $Y = \{-1,1\}$. Assume that $\mathcal{C}$ can be learned with mistake bound t using algorithm A. (You may also assume at each iteration A runs in time polynomial in n, as well as that A only updates its state when it gets an example wrong.) We want to show how a learner, given A, can PAC-learn concept class $\mathcal{C}$ with respect to any distribution D on $\{-1,1\}^n$. The learner can use A as part of its output hypothesis and should run in time polynomial in n, $1/\epsilon$, and $1/\delta$. For the rest of the problem, fix some distribution D on X, and say the examples are labeled by an unknown $c \in \mathcal{C}$. For a classifier (i.e. function) $h : X \to Y$, let $\text{err}(h) = \mathbb{P}_{x \sim D}[h(x) \neq c(x)]$.

   (a) Fix a classifier $h : X \to Y$. If $\text{err}(h) > \epsilon$, what is the probability that h gets k random examples all correct? How large does k need to be for this probability to be at most δ'? The contrapositive view would be: unless the data is highly misleading, which happens with probability at most δ', it must be the case that $\text{err}(h) \leq \epsilon$. (Make sure this makes sense.)

   (b) How many times can A possibly update its state? That is, how many different classifiers can it possibly go through? And therefore, how many examples do we need to see before we can be sure of getting a block of k examples all correct? (Think about dividing the stream of examples into blocks of size k.)

   (c) Let E_i be the event that A in its ith state, call it h_i, has error greater than ϵ, yet we output h_i. (When would this happen?) Argue that the "failure event" of our overall PAC learner is $E = \cup_i E_i$.

   (d) Put everything together and fully describe a PAC learner that is able, with probability of failure at most δ, to output a classifier with error at most ϵ. How many examples does the learner need to use (as a function of ϵ, δ, and t)?