Computational Learning Theory
Hilary Term 2018
Week 3

Problem Sheet 1

Instructions: The problem sheets are designed to increase your understanding of the material taught in the lectures, as well as to prepare you for the final exam. You should attempt to solve the problems on your own after reading the lecture notes and other posted material, where applicable. Once you have given sufficient thought to a problem, if you are stuck, you are encouraged to discuss with others in the course and with the lecturer during office hours. Please avoid posting on Piazza until the Wednesday before the submission deadline. You are *not permitted* to search for solutions online.

1 Learning Hyper-rectangles

The concept class of hyper-rectangles over $\mathbb{R}^n$ is defined as follows:

$$C_n = \{[a_1, b_1] \times \cdots \times [a_n, b_n] \mid a_i, b_i \in \mathbb{R}, a_i < b_i\}$$

Generalise the algorithm discussed in class (for rectangles in $\mathbb{R}^2$) and show that it *efficiently* PAC learns the class of hyper-rectangles. Give bounds on the number of samples required to guarantee that the error is at most ϵ with probability at least $1 - \delta$.

Note: You may assume that the distribution D over $\mathbb{R}^n$ is defined using a density function that is "smooth" (absolutely continuous) over all of $\mathbb{R}^n$. (*Optional*: As an extra challenge, argue why the algorithm still works even when such an assumption regarding the distribution does not hold.)

2 PAC Learning : Confidence Parameter

An algorithm A "perhaps learns" a concept class C, if for all n, all $c \in C_n$, for every D over X_n and for every $0 < \epsilon < 1/2$, A given access to $\mathsf{EX}(c, D)$ and inputs ϵ and size(c), runs in time polynomial in n, size(c), $1/\epsilon$ and outputs a polynomially evaluatable hypothesis h such that with probability at least $3/4$, $\text{err}(h) \leq \epsilon$. In other words, we've set $\delta = 1/4$ in the standard definition of PAC-learning. Show that if C is "perhaps learnable" then C is also efficiently PAC-learnable.

Hint: You will have to use the Chernoff-Hoeffding bound.

3 Hardness of Learning Boolean Threshold Functions

We will consider the question of learning boolean threshold functions. Let $X_n = \{0,1\}^n$ and for $w \in \{0,1\}^n$ and $k \in \mathbb{N}$, $f_{w,k} : X_n \to \{0,1\}$ is a boolean threshold function defined as follows:

$$f_{w,k}(x) = \begin{cases} 1 & \text{if } \displaystyle\sum_{i=1}^{n} w_i \cdot x_i \geq k \\ 0 & \text{otherwise} \end{cases}$$

Let $\mathsf{TH}_n = \{f_{w,k} \mid w \in \{0,1\}^n, 0 \leq k \leq n\}$ and $\mathsf{TH} = \bigcup_{n \geq 1} \mathsf{TH}_n$. Show that unless $\mathsf{RP} = \mathsf{NP}$, there is no efficient PAC-learning algorithm for learning TH, if the output hypothesis is also required to be in TH, *i.e.,* a *proper* PAC-learning algorithm.

Hint: You should reduce from Zero-One Integer Programming (ZIP) which is known to be NP-complete. An instance of ZIP consists of an $s \times n$ matrix A with entries in $\{0,1\}$, a vector $\mathbf{b} \in \{0,1\}^s$ and a pair $(\mathbf{c}, B)$ (the objective). The decision problem is to determine wither there exists an assignment for the n variables $z_1, \ldots, z_n$, each variable taking a value in the set $\{0,1\}$, such that for each $1 \leq i \leq s$, $\displaystyle\sum_{j=1}^{n} A_{ij} z_j \leq b_i$ and $\displaystyle\sum_{j=1}^{n} c_j z_j \geq B$.