Assignment 3

COMP 599 Winter 2016 McGill University
Instructor: Prakash Panangaden

Due 7th March 2016

**Question 1** [20 points] Give a subset $S \subseteq \mathbb{R}^n$ of cardinality $n + 1$ that can be shattered by the class of linear threshold functions. Remember that linear threshold functions are of the form $\vec{v} \cdot \vec{w} + \geq b$ where $\vec{v}$ is the vector in $\mathbb{R}^n$ and $\vec{w}$ is some suitable weight vector which defines the hyperplane. In class we looked at cases where $b = 0$ but you are, of course, free to consider $b \neq 0$.

**Question 2** [30 points]

1. For each fixed $k$, what is the VC dimension of the class of subsets of the real line expressible as the union of $k$ or fewer closed intervals? Justify your answer.

2. Prove that the class of hyper-rectangles in $\mathbb{R}^n$, of the form $[a_1, b_1] \times \ldots \times [a_n, b_n]$, has VC dimension $2n$.

**Question 3** [20 points] For $i = 1, \ldots, n$, define $\vec{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$ by

\[ \vec{x}_i = ((-1)^i, \ldots, (-1)^i, (-1)^{i+1}, 0, \ldots, 0) \text{ and } y_i = (-1)^{i+1}. \]

Suppose that the Perceptron algorithm is run cyclically over the sequence $S = (\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$ until it makes no more mistakes. We are using perceptron in off-line mode. We run it again and again on this data in the given order until it makes no more mistakes.

Show that the total number of mistakes is at least $2^{n-3}$.

Just to make sure that you understand the $\vec{x}_i$, here are the first few:

$\vec{x}_1 = (1, 0, 0, \ldots, 0)$

$\vec{x}_2 = (1, -1, 0, 0, \ldots, 0)$

$\vec{x}_3 = (-1, -1, 1, 0, 0, \ldots, 0)$

$\vec{x}_4 = (1, 1, 1, -1, 0, 0, \ldots, 0)$

**Hint.** Note that this algorithm will only ever assign integer values to the weights so it suffices to consider vectors in $\mathbb{Z}$. Let $\vec{w} = (w_1, \ldots, w_n) \in \mathbb{Z}^n$ be (a normal vector of) any linear separator. Give lower bounds on the magnitude of the $w_i$. Then argue that the $n$-th component is increased by at most 1 in every update so the size of $w_n$ gives your lower bound.
Question 4: Winnow for Monotone Disjunctions [30 points] This question concerns a simple version of the Winnow algorithm for learning disjunctions of propositional variables, that is, disjunctions of positive literals. These are called monotone disjunctions.

1. Initialise all weights \( w_1, \ldots, w_n \) to 1.

2. Given an example \( \vec{x} \in \{0, 1\}^n \), Output 1 if
\[
 w_1 x_1 + \ldots + w_n x_n \geq n
\]
and output 0 otherwise.

3. If the algorithm makes a mistake:
   (a) If the algorithm predicts negative on a positive example, then for each \( x_i \) equal to 1, double the value of \( w_i \).
   (b) If the algorithm predicts positive on a negative example, then for each \( x_i \) equal to 1, halve the value of \( w_i \).


Suppose that the target concept is a disjunction of \( r \) variables.

1. Show that the algorithm makes at most \( r(1 + \log_2 n) \) mistakes on positive examples.

2. Show that the algorithm makes at most twice the number of mistakes on negative examples as on positive examples (Hint. Consider the effect of positive and negative mistakes on the total weight \( \sum_{i=1}^{n} w_i \)).

3. Argue that total number of mistakes is at most \( 2 + 3r(1 + \log_2 n) \).