Machine Learning: Foundations **Spring Semester, 1996/7**

Homework 2: March 26

Lecturer: Yishay Mansour

Homework number 2.

Programming assignment:
Write a program to implement the AdaBoost algorithm. Use as a weak learning hypothesis the predicates $x_i \geq \alpha$, where x_i is an attribute and α a real value. Run the program on **all** the data sets in `http://www.math.tau.ac.il/~mansour/ml-course`.
File format: For each database you have a data file and an information file, that describing the input. In a nutshell, each line in the data file is a record, and the last value in the line is the one you would like to predict.
Methodology: You take a random part of the input to be a *tarining set*, this will be used in order to predict the classification. You take a different random part to be the *test set*. For each point in the test set, you compute the prediction, based on the training set, and compare it to the true value. (This is how you get an estimate to the number of errors.) *Run the program a few times and take the average.* The size of the taining set should be a parameter that you can adjust.

Answer the following questions:

1. What happens when you increase the number of iterations.
2. How does the distributions generated on the way change (quatatively).
3. What happens when you increase the training set.

Theory question:
For $0 \leq a < b \leq 1$, let $[a, b]$ be a subinterval of $[0, 1]$ that includes all the points $x \in \mathbf{R}$ such that $a \leq x \leq b$. Let $\mathcal{C}_d$ the concept class over $[0, 1]$ in which each concept is the union of at most d subintervals, i.e., every $c \in \mathcal{C}_d$ can be characterize by $(a_1, b_1) \cdots (a_d, b_d)$, such that $c(x) = 1$ iff there is an i such that $x \in [a_i, b_i]$.

1. Give a PAC algorithm for learning $\mathcal{C}_d$ using the hypothesis class $\mathcal{H} = \mathcal{C}_d$.
2. Show that there is no PAC algorithm for learning $\mathcal{C}_d$ using the hypothesis class $\mathcal{H} = \mathcal{C}_d$, that uses a smaple size that does not depend on d. (Hint: Assume, for contradiction, that such an algorithm exists. Fix $\epsilon = 0.1$ and $\delta = 0.1$, then the algorithm draws a fixed sample size m. Show a contradistion by exhibiting a specific distribution for which this is not true.)
3. [**Bonus**] What are the *best* lower bound and upper bound you can derive for the sample size for $\mathcal{C}_d$.

The homework is due in two weeks