Machine Learning: Foundations | **Spring Semester, 2001/2**

Homework 2: April 18

Lecturer: Yishay Mansour

Homework number 2.

Programming assignment:
Write a program to implement the AdaBoost algorithm. Run the program on two different data sets.
Weak Learner: Use as a weak learning hypothesis the predicates $x_i \geq \alpha$, where x_i is an attribute and α a real value. (In each iteartion, simlpy enumerate all the possible weak learner, and compute the error of each, and select the weak learner with the lowest error.)
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: Take a random part of the input to be a *training set*, to train AdaBoost. The remaining examples are a *test set*. Given the training set compute (using AdaBoost) a classifier. For each example in the test set, compute the classifier prediction, and compare it to the true value. The fraction of the examples in the test set on which the classifier errs, is an estimate of the error rate. *Run the program a few times and take the average.* The size of the training set should be a parameter that you can adjust.

Plan you experiments such that you will be able to answer the following questions:

1. What happens when you increase the number of iterations.
2. What happens when you increase the training set.
3. [**Bonus**] How does the distribution changes through the different iterations (qualitatively).

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 sample 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 contradiction 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 $\mathcal{C}_d$, as a function of d.

The homework is due in two weeks