15-859(B) Machine Learning Theory

Homework # 4 **Due: February 27, 2008**

Groundrules: Same as before. You should work on the exercises by yourself but may work with others on the problems (just write down who you worked with). Also if you use material from outside sources, say where you got it.

Exercises:

1. [**VC-dimension**] Show that if hypothesis class H has VC-dimension d, then the class $\mathrm{MAJ}_k(H)$ has VC-dimension $O(kd \log kd)$. Recall that $\mathrm{MAJ}_k(H)$ is the class of functions achievable by taking majority votes over k functions in H (let's say that we allow repetitions).

2. [**PAC learnability in principle**] When we defined the PAC model, we defined a "concept class" to be not just a set of functions but also a representation language for describing those functions. We then defined a class C to be PAC-learnable if there exists an algorithm $\mathcal{A}$ whose time and sample size needed to produce a hypothesis of error at most ϵ with probability at least $1-\delta$ is polynomial in $1/\epsilon$, $1/\delta$, the size n of examples, *and* the description-length of the target function.

   (a) There is no known polynomial-time algorithm to learn the class of Decision Trees over $\{0,1\}^n$. Describe a concept class C that describes the same *set* of functions over $\{0,1\}^n$ as Decision Trees, and yet can be learned efficiently in the PAC model.

   (b) Say that a concept class (representation language) C is "well-behaved" if given a string of bits c (interpreted as describing a function in C) and an example x, it is possible to evaluate $c(x)$ in time polynomial in the length of c and the length of x. Prove that if **P**=**NP**, then there is a generic algorithm to PAC-learn any well-behaved concept class.

Problems:

3. [**On the plausibility of boosting**] Suppose we have a finite hypothesis class H, a finite space of instances X (e.g., $X = \{0,1\}^n$), and some unknown target function f. Suppose that for any distribution D over X there exists an $h \in H$ with error at most $1/2 - \gamma$. Without going through the full boosting analysis, use the minimax theorem plus Hoeffding bounds to prove that for any distribution D there must *exist* a hypothesis in $\mathrm{MAJ}_k(H)$ with error at most ϵ for $k = O(\frac{1}{\gamma^2}\log(1/\epsilon))$.

   Note: our boosting results said something even stronger because they gave us a way to efficiently produce the desired hypothesis, given a weak-learning oracle.

4. **[On approximate Nash equilibria]** A two-player general-sum game is like a two-player zero-sum game except that the players do not necessarily have opposite payoffs (it is really more an "interaction" than a "game"). A Nash Equilibrium is a pair of distributions P and Q (one for each player) such that neither player has any incentive to deviate from its distribution assuming that the other player doesn't deviate from its distribution either.[1] Formally, a pair of distributions P (for the row player) and Q (for the column player) is a Nash equilibrium if the following holds: assuming the column player plays at random from Q, the expected payoff to the row player for each row r with $P(r) > 0$ is equal to the maximum payoff out of all the rows; and assuming the row player plays at random from P, the expected payoff to the column player for each column c with $Q(c) > 0$ is equal to the maximum payoff out of all the columns.

   Now, assume we have a game in which all payoffs are in the range $[0, 1]$. Define a pair of distributions P, Q to be an "ϵ-Nash" equilibrium if each player has *at most* ϵ incentive to deviate. That is, the expected payoff to the row player for each row r with $P(r) > 0$ is within ϵ of the maximum payoff out of all the rows, and vice-versa for the column player.

   Using the fact that Nash equilibria must exist (proven by Nash in 1950), show that there must exist an ϵ-Nash equilibrium in which each player has positive probability on at most $O(\frac{1}{\epsilon^2} \log n)$ actions (rows or columns), where n is the total number of rows and columns.

   Note: this fact immediately yields an $n^{O(\frac{1}{\epsilon^2} \log n)}$-time algorithm for finding an ϵ-Nash equilibrium. No PTAS (algorithm running in time polynomial in n for any fixed $\epsilon > 0$) is known, however.

[1] Feel free to use the Web or ask your friends to learn more about general-sum games if you haven't seen them before. Or see, e.g., http://www.cs.cmu.edu/~avrim/451/lectures/lect1204.pdf.