

15-859(B) Machine Learning Theory

Homework # 4

Solutions

Exercises:

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

Solution: Let m be the VC-dimension of $\text{MAJ}_k(H)$, so by definition, there must exist a set S of m points shattered by $\text{MAJ}_k(H)$. We know by Sauer's lemma that there are at most m^d ways of partitioning the points in S using functions in H . Since each function in $\text{MAJ}_k(H)$ is determined by k functions in H , this means there are at most m^{kd} ways of partitioning the points using functions in $\text{MAJ}_k(H)$. Since S is shattered, we therefore must have $2^m \leq m^{kd}$, and so $m \leq 2kd \lg(kd)$ (for $kd \geq 4$).

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. E.g., let C be the class of all boolean functions described as truth-tables. This is easy to learn since we are now allowed to run in time $\text{poly}(2^n)$ which means we can just draw and memorize a large sample.

(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 $\mathbf{P}=\mathbf{NP}$, then there is a generic algorithm to PAC-learn any well-behaved concept class.

We can iteratively try all possible lengths for the target function. For each $i = 1, 2, \dots$, we will draw m_i examples, and call our \mathbf{NP} oracle to see if there is any string c of length at most i whose associated concept is consistent with our data. If so, output it. Else, we increment i . So long as we choose appropriate values of m_i , like $m_i = \frac{1}{\epsilon} \ln(\delta/2^i)$, there is at most a δ chance we are ever fooled.

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 $\text{MAJ}_k(H)$ with error at most ϵ for $k = O(\frac{1}{\gamma^2} \log(1/\epsilon))$.

Solution: Consider a matrix game in which the rows are the hypotheses in H , the columns are the examples in X , and the value in entry (h, x) equals 1 if $h(x)$ is wrong and 0 if $h(x)$ is correct. The values are payoffs to the column player (so the row-player is trying to minimize and the column player is trying to maximize). We are told that for any mixed strategy D of the column-player there exists a row such that the expected payoff to the column player is at most $1/2 - \gamma$. This implies by the minimax theorem that there exists a distribution P for the row-player such that for any column x , the expected payoff to the column player is at most $1/2 - \gamma$. That is, for all x , $\Pr_{h \sim P}[h(x) \neq f(x)] \leq 1/2 - \gamma$.

So, for any given x , choosing h from P is like flipping a coin of bias at most $1/2 - \gamma$. By Hoeffding bounds, if we flip k times, the probability we see more heads than tails is at most $e^{-2k\gamma^2}$. That is, for all x , $\Pr_{h_1, \dots, h_k \sim P}[\text{MAJ}(h_1(x), \dots, h_k(x)) \neq f(x)] \leq e^{-2k\gamma^2} = \epsilon$ for $k = \frac{1}{2\gamma^2} \ln \frac{1}{\epsilon}$. Since this is true for all x , it is certainly true on average for x chosen from D . This implies the *expected* error of a hypothesis in $\text{MAJ}_k(H)$ chosen this way is at most ϵ , and so a hypothesis in $\text{MAJ}_k(H)$ of error at most ϵ must exist.

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

¹Feel 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/lect1205.pdf>.

$P(r) > 0$ 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.

Solution: Consider some Nash equilibrium (P, Q) . Let S be a (multi-)set of k rows selected iid from P , and let T be a multi-set of k rows selected iid from Q . Let U_S denote the uniform distribution over S and let U_T denote the uniform distribution over T . The claim is that $k = O(\frac{1}{\epsilon^2} \log n)$ is sufficient so that with high probability, the pair (U_S, U_T) is an ϵ -Nash equilibrium (so such a pair must exist). In particular, by Hoeffding bounds, this value of k is sufficient so that with high probability, for every column c , its average payoff over the rows in S is within $\pm\epsilon/2$ of its expected payoff with respect to the distribution P . Similarly, with high probability, for every row r , its average payoff over the columns in T is within $\pm\epsilon/2$ of its expected payoff with respect to the distribution Q . So long as both conditions hold, this implies that the pair (U_S, U_T) has the property that each player has at most ϵ incentive to deviate.

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.