CSC 2416 Machine Learning Theory
ASSIGNMENT # 2
Given: Oct 20, Due: November 10

1. The Occam's razor result proved in class only applies to finite $\mathcal{H}$. Suppose now that $\mathcal{H}$ is countably infinite or finite. Let $P : \mathcal{H} \rightarrow [0, 1]$ be any function such that $\sum_{h \in \mathcal{H}} P(h) = 1$. In other words, P defines a probability distribution over the space of hypotheses. Let m be the number of given examples, each chosen at random, as usual, from an unknown distribution $\mathcal{D}$.

   a. Prove that with probability at least $1 - \delta$

   $$Err_{\mathcal{D}}(h) \leq O(\left(\frac{ln(1/P(h)) + ln(1/\delta)}{m}\right)$$

   for all $h \in \mathcal{H}$ that are consistent with the observed data. As usual, $err_{\mathcal{D}}(h) = Pr_{x \in \mathcal{D}}[h(r) \neq c(r0]$, and c is the target concept. Give explicit constants if possible.

   b. When $\mathcal{H}$ is finite, show how to choose P to recover the bound proved in class:

   $$Err_{\mathcal{D}}(h) \leq O\left(\frac{ln(|\mathcal{H}|) + ln(1/\delta)}{m}\right)$$

   for any consistent $h \in \mathcal{H}$, with probability at least $1 - \delta$.

   c. Suppose that each hypothesis in $\mathcal{H}$ is encoded by bits and that $|h|$ denotes the number of bits needed to encode h. Show how to choose P to prove that

   $$err_D(h) \leq O\left(\frac{|h| + ln(1/\delta)}{m}\right)$$

   for all $h \in \mathcal{H}$ that are consistent with the observed data, with probability at least $1 - \delta$.

2. Consider the following learning problem: Let the domain be the real numbers and let the concept class $\mathcal{C}_s$ defined by unions of s intervals. That is, each concept c is defined by real numbers $a_1, b_1, \ldots a_s, b_s$ where $c(x) = 1$ if and only if $x \in [a_1, b_1] \cup \ldots \cup [a_s, b_s]$. Describe an efficient algorithm that learns the class $\mathcal{C}_s$ for every s, assuming that s is known ahead of time to the learner. You should describe a single algorithm that works for all $\mathcal{C}_s$, proved that s is known so that the learner can choose the number of examples needed as a function of ϵ, δ and s. You can use any hypothesis space that you wish. Prove that your algorithm is PAC, and derive an exact expression for the number of examples needed. Also argue briefly that your algorithm runs in time polynomial in $1/\epsilon$, $1/\delta$ and s.

3. For each $d = 0, 1, 2, \ldots$, give an example of a concept class C for which Sauer's Lemma is tight. That is, for which the VC-dimension of C is d, and for each m, $\Pi_C(m) = \sum_{i=0}^{d} \binom{m}{i}$.

4. In the *batch version* of the PAC model, the learner must specify how many examples it requires *before* seeing any of the data. Thus before learning begins, the learning algorithm specifies the number of examples needed, and cannot later ask for more examples.

   In contrast, in the *oracle version* of the PAC model (this is the one we defined in class), the learner is not provided with a fixed batch of examples, but instead is provided with an example oracle EX. The learner requests one example at a time from EX. Each call provides the learner with a single example. As usual, eacah example r is selected at random according to the distribution D, and both r and its label $c(r)$ are provided to the learner. Thus in this model, the learner is provided with ϵ, δ and the oracle EX and can request as many examples as it wishes from EX. As usual, the hypothesis that the learner outputs must have error at most ϵ with probability at least $1 - \delta$. Moreover, the runtime and hence the total number of examples requested must be bounded by a polynoimal.

   Thus the difference between these two models is that in the batch version, the learner must decide ahead of time how many examples it needs, while in the oracle version it can dynamically decide how many examples it needs based on the data so far.

   Let the domain be the reals, and let C_s be the class of concepts defined by unions of s intervals. That is, each concept c is defined by real numbers $a_1, b_1, \ldots, a_s, b_s$ where $c(x) = 1$ if and only if $x \in [a_1, b_1] \cup \ldots \cup [a_s, b_s]$. Efficient means that the runtime and number of samples are polynomial is $1/\epsilon$, $1/\delta$ and s.

   - In the batch version, describe an efficient algorithm that learns C_s for every s, as long as s is known ahead of time to the learner. Your learner can choose the number of examples as a function of ϵ, δ and s. Prove that your algorithm is an efficient PAC algorithm.
   - Still in the batch version, assume now that the learner does not know s ahead of time, so that the number of examples needed is only a function of ϵ and δ. In this case, show that there is no algorithm (whether efficient or not, and regardless of the hypothesis space used) that can PAC-learn the class C_s for every s.