COS 511: Foundations of Machine Learning

Homework #1

Due in class:
February 18, 2003

General comments: If you get stuck on a harder problem, you are welcome to work on it with someone else (although obviously you should only work with someone who does not already know the solution). However, please first read and think about every problem on your own since you will probably find that some of the problems are much easier than others. Even if you work with others, you must write up your solutions on your own in your own words, indicating who you worked with on which problems.

If you happen to have already seen one of these problems elsewhere (some of these problems are well known), please indicate this when you write up your solution; I won't take off credit, I just would like to know. Although background reading is always encouraged, please do not attempt to "solve" these problems by looking for the solutions in the literature.

Problem 0

If possible, please submit with your homework a copy of a picture of yourself (that I can keep), so that I can try to attach names to faces. Although I would prefer hardcopy, a URL pointing to an online photo would also be okay.

Problem 1

The Occam's razor result proved in class only applies to finite $\mathcal{H}$. Suppose now that $\mathcal{H}$ is discrete, i.e., either finite or countably infinite. Let $P : \mathcal{H} \to [0,1]$ 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 hypothesis concepts.

Let m be the number of given examples (each chosen at random, as usual, from some unknown distribution D).

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

$$\mathrm{err}_D(h) \le O\left(\frac{\ln(1/P(h)) + \ln(1/\delta)}{m}\right)$$

for all $h \in \mathcal{H}$ that are consistent with the observed data. As usual, $\mathrm{err}_D(h) = \Pr_{x \sim D}[h(x) \neq c(x)]$, and c is the target concept. Give explicit constants (in other words, give a bound that does not use $O(\cdot)$ notation).

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

$$\mathrm{err}_D(h) \le O\left(\frac{\ln(|\mathcal{H}|) + \ln(1/\delta)}{m}\right)$$

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

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

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

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

d. How does the bound in (c) reflect our intuition that "simpler" hypotheses should be prefered to more "complex" ones? How does the bound in (a) reflect our intuition that prior knowledge helps learning?

Problem 2

In the two-distribution version of the PAC model, the learner is given access to random positive and negative examples separately, each having its own distribution, and must find a hypothesis with low error on both. That is, formally, a class $\mathcal{C}$ is two-distribution learnable by $\mathcal{H}$ if there is an algorithm A such that for all $c \in \mathcal{C}$, for all distributions D^+ and D^- over the positive and negative examples (respectively), for all $\epsilon > 0$ and for all $\delta > 0$, when A is given m^+ positive and m^- negative random examples (distributed according to D^+ and D^-), where m^+ and m^- are polynomial functions of $1/\epsilon$ and $1/\delta$, A computes a concept $h \in \mathcal{H}$ such that, with probability at least $1 - \delta$:

$$\Pr_{x \sim D^+}[h(x) = 0] \leq \epsilon$$

and

$$\Pr_{x \sim D^-}[h(x) = 1] \leq \epsilon.$$

Assume that $\mathcal{H}$ includes the two constant functions h_0 and h_1 identically equal to 0 and 1, respectively. Prove that $\mathcal{C}$ is two-distribution learnable by $\mathcal{H}$ if and only if $\mathcal{C}$ is PAC learnable by $\mathcal{H}$. Your proof should also preserve efficiency; in other words, it should show that polynomial-time learnability in either model implies the same in the other model.