COS 511: Foundations of Machine Learning

Homework #3 Due: March 2, 2004
Learning models & Chernoff

Problem 1

In the *batch version* of the PAC model, which is the one that we have been studying in class, 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, the learner is not provided with a fixed batch of examples, but is instead 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, each example x is selected at random according to the distribution D, and both x and its label $c(x)$ 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 total number of examples requested must always be bounded by a polynomial.

So, 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 received so far. This problem explores this difference for a simple example studied on a previous problem set.

As on HW#1 (problem 2), let the domain be $X = \mathbb{R}$, and let $\mathcal{C}_s$ 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 $x \in [a_1, b_1] \cup \cdots \cup [a_s, b_s]$. For the purposes of this problem, "efficient" means that the time and sample requirements are polynomial in $1/\epsilon$, $1/\delta$ and s.

Note that the previous problem involving this concept class showed that in the batch version, there exists an efficient algorithm that learns the class $\mathcal{C}_s$ for every s when s is known ahead of time to the learner.

a. [10] 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 $\mathcal{C}_s$ for every s.

b. [15] Turning now to the oracle version, let us continue to assume that the learner does *not* know s ahead of time. Describe an efficient algorithm that learns the class $\mathcal{C}_s$ for every s even though s is not known by the learner. Although s is not known at the beginning of the learning process, show that by the time the algorithm halts, the total number of examples requested does not exceed a polynomial in $1/\epsilon$, $1/\delta$ and s. Prove that your algorithm is PAC, and derive a "big-Oh" expression for the number of examples needed. Also argue that your algorithm halts in time polynomial in $1/\epsilon$, $1/\delta$ and s. Be sure to show carefully that the total probability of your algorithm failing to find a hypothesis with error at most ϵ cannot exceed δ.

Problem 2

Let D be a distribution over $X \times \{0,1\}$, and let $S = \langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ be a random sample from D. Let

$$\begin{aligned} \mathrm{err}(h) &= \Pr_{(x,y)\sim D}\left[h(x) \neq y\right] \\ \widehat{\mathrm{err}}(h) &= \frac{1}{m}|\{i : h(x_i) \neq y_i\}|. \end{aligned}$$

For simplicity, we will assume that $\mathcal{H}$ is finite, although the results of this problem can be carried over to the infinite case. Note that none of the results depend on $|\mathcal{H}|$.

Let $\hat{h}$ and h^* be the hypotheses in $\mathcal{H}$ with minimum training error and generalization error, respectively:

$$\begin{aligned} \hat{h} &= \arg\min_{h\in\mathcal{H}} \widehat{\mathrm{err}}(h) \\ h^* &= \arg\min_{h\in\mathcal{H}} \mathrm{err}(h). \end{aligned}$$

Be sure to keep in mind that, unlike h^*, $\hat{h}$ is a *random variable* that depends on the random sample S.

a. [10] Prove that
$$\mathrm{E}\left[\widehat{\mathrm{err}}(\hat{h})\right] \leq \mathrm{err}(h^*) \leq \mathrm{E}\left[\mathrm{err}(\hat{h})\right].$$

b. [15] Prove that, with probability at least $1 - \delta$,
$$\left|\widehat{\mathrm{err}}(\hat{h}) - \mathrm{E}\left[\widehat{\mathrm{err}}(\hat{h})\right]\right| \leq O\left(\sqrt{\frac{\ln(1/\delta)}{m}}\right).$$

Give explicit constants, and be sure to end up with a result that does not depend on $|\mathcal{H}|$.

c. [5] Explain in words the meaning of what you proved in both of the preceding parts, and how we would expect training error to compare to test error when using a machine learning algorithm on actual data.

Problem 3 – Extra Credit

[15] Let D be a distribution over $X \times \{0,1\}$, and let $\mathcal{H}$ be a hypothesis space with VC-dimension d. Let $(x_1, y_1), \ldots, (x_m, y_m)$ be a random sample from D, where $m \geq d$. Prove that, with probability at least $1 - \delta$, for every hypothesis $h \in \mathcal{H}$,

$$|\mathrm{err}(h) - \widehat{\mathrm{err}}(h)| \leq \epsilon$$

where $\mathrm{err}(h)$ and $\widehat{\mathrm{err}}(h)$ are as in Problem 2, and

$$\epsilon = O\left(\sqrt{\frac{d\ln(m/d) + \ln(1/\delta)}{m}}\right).$$

Give explicit constants for your choice of ϵ.