15-850(C) Machine Learning Theory

Lecturer: Avrim Blum
Scribe: Scott Davies

Lecture 4: September 28, 1995

---

1 Additional notes on decision lists

Last lecture, we saw that we can PAC-learn any concept class C with a limited (i.e., $2^{poly(n)}$) number of hypotheses, simply by requesting m examples and returning any hypothesis consistent with those examples, for m satisfying the equation

$$m \geq \frac{1}{\epsilon}[\ln(|C|) + \ln(\frac{1}{\delta})].$$

We also discussed the concept class of decision lists, along with the following simple learning algorithm: find any rule consistent with the data and place it at the bottom of the rule set we have so far; throw out the data for which the current rule set provides consistent predictions; repeat until no data is left. This algorithm runs in polynomial time, and we saw that it solves the consistency problem. Since there are only a polynomial number of possible decision lists, we can PAC-learn decision lists with this algorithm.

This algorithm and its analysis are easily generalized to handle k-decision lists. (Reminder: k-decision lists are just like "ordinary" decision lists, except their rules can test for the truth of k-literal conjunctions instead of just single literals.) As we'll see in Assignment 2, any decision tree of size s can be represented with a ($\log s$)-decision list. So, k-decision lists are fairly "expressive" — that is, they're capable of representing a fairly large and useful class of hypotheses.

One note of caution about the notion of "expressiveness": the important question is not, "What concepts can be represented?", so much as, "What concepts can be represented concisely, i.e. with polynomially sized concepts in the hypothesis class?" Decision trees can represent arbitrary boolean functions, but some functions require an exponential number of nodes in their decision tree representations — for example, the parity function requires a full decision tree with 2^n leaves. In cases like this, our algorithm isn't "learning" in any real sense; it's essentially memorizing the data.

Here's an interesting open problem: suppose the features in our domain are string-valued rather than binary — i.e., our instance space is now $(\Sigma^*)^n$. A decision list for such a domain might look something like

```
if (x1 = "Tuesday") then -
   else if (x5 = "Bovik") then +
      else if (x2 = "INC TG") then -
         ...
```

The same kind of algorithm we used for binary decision lists would allow us to find a consistent decision list in polynomial time. Unfortunately, there's no *a priori* bound on the number of values strings can take on, so the hypothesis space is unbounded. Nobody knows whether or not decision lists in this domain are PAC-learnable. We'll talk more about learning concept classes in this setting in the next lecture.

2 Occam Learning

Rehash from last lecture: suppose $size(c)$ is a "legal" size function, i.e., one such that no more than 2^s concepts have size less than s for any given s. Then, given m examples, with probability $\geq 1 - \delta$ there are no bad (error $> \epsilon$) hypotheses of size $\leq s$, where

$$s < \frac{1}{\ln 2}(\epsilon m - ln\frac{1}{\delta}).$$

Here is an interesting extension of this fact.

Theorem 1 *Suppose we have an algorithm that continues to request examples until it finds a hypothesis such that (1) the hypothesis is consistent with the data, and (2) the size s of the hypothesis satisfies*

$$s \leq \frac{1}{\ln 2}(\epsilon m - \ln\frac{2m^2}{\delta}),$$

where m is there number of examples it's seen so far. Then, with probability $1 - \delta$, this algorithm will never output a hypothesis with error rate $\geq \epsilon$.

Proof: Consider a fixed value of m. The chance of any small, "bad" hypotheses being consistent with the data so far is only $\delta' = \frac{\delta}{2m^2}$, by our previous analysis. Since

$$\sum_{m=1}^{\infty} \frac{\delta}{2m^2} \leq \delta,$$

the probability that we'll ever output a bad hypothesis is less than δ. ■

3 Are Complicated Hypotheses Necessarily Bad?

Our analysis so far has shown that if we have a small hypothesis space, then by taking a polynomial number of examples we can achieve "uniform convergence" — that is, the chance that any bad (error $> \epsilon$) hypothesis will still be consistent with the data can be forced below δ. Is the converse true? That is, with large hypotheses spaces, is it impossible to get uniform convergence with a polynomial number of examples? Answer: sometimes.

Let X be a random variable representing the number of "bad" hypotheses consistent with the data we've seen so far. One way of stating the analysis we've used to show that small hypothesis spaces guarantee uniform convergence is that

$$\mathbf{Prob}[X > 0] \leq E[X] \leq |H|(1-\epsilon)^m$$

So, if $|H|$ is small, $\mathbf{Prob}[X > 0]$ must also be small. What about the converse — if $|H|$ is large, must $\mathbf{Prob}[X > 0]$ also be large?

Not always. If the hypotheses mostly make mistakes on the same instances, then $E[X]$ may be much larger than $\mathbf{Prob}[X > 0]$. However, consider the case that H contains N hypotheses of error rate ϵ such that the events of these hypthoses being consistent with the sample are independent. Then,

$$\mathbf{Prob}[\text{All } h_i \text{ inconsistent}] \ = \prod_{i=1}^{N}[h_i \text{ is inconsistent}] \ = (1-(1-\epsilon)^m)^N \qquad (1)$$

If N is large — say, about $\frac{1}{(1-\epsilon)^m}$ — then this probability is about $\frac{1}{e}$. The probability of there being a consistent "bad" hypothesis is then $1 - \frac{1}{e}$. In other words, you can't say that consistency implies PAC-learning when there are lots of "independent" hypotheses.

3.1 An example

Let's consider the behavior of Crook County in presidential elections. As reported in the *Christian Science Monitor*, Oct 20 1992, Crook County had voted with the winning candidate for President of the United States in all 27 of the elections since its inception. How "surprising" a result is this? Must the people of Crook County have keen political insight, or might it just be dumb luck? Let's do a simple calculation.

There are 3141 counties in the U.S., and let's suppose that each county i has an independent probability p_i of correctly predicting the outcome of the election. Furthermore, suppose the p_i's are fixed over time.

For a county i with $p_i = \frac{3}{4}$, the probability of correctly predicting 27 elections in a row is $(\frac{3}{4})^{27} \approx \frac{1}{2362}$. If all 3141 counties i have $p_i = \frac{3}{4}$, then the probability that at least one of them correctly predicts 27 elections in a row is

$$1 - (1 - (\frac{3}{4})^{27})^{3141} \approx .735$$

So, if each of the counties had a .75 chance of voting with the winning candidate, it's not at all surprising that one of them voted with the winner in all 27 of the last elections.

On the other hand, suppose that all counties have probability $p_i = 0.7$ of predicting each outcome. Then, the chance of any county surviving 27 elections is at most:

$$3141(1-0.3)^{27} \approx 0.2.$$

(This is like our PAC bounds with $\epsilon = 1 - p = 0.3$ and $\delta = 0.2$.) This suggests it is reasonable to expect that Crook County is better than an 0.7 predictor, but probably not much better than an 0.75 predictor.

In fact, the model of each county predicting correctly with probability 0.75 fits historical data reasonably well. Eight years before, there had been two counties that had successfully predicted the last 25 elections, wheras the expected number predicted the model is 2.4; similarly, eight years before that, there had been four counties that had correctly predicted the last 23 elections, whereas the expected number predicted by our model is 4.2.

4 Dealing with Inconsistent Hypothesis Spaces

So far, we've concerned ourselves mostly with guarantees in situations where we only output hypotheses that are completely consistent with the data. What if there are no such hypotheses — what if all hypotheses in our class have nonzero observed error rates? What kinds of guarantees might we want to give in such situations? We might want to guarantee that any hypotheses that look good on a sample set would be at least "pretty good" if tested against the true problem distribution. Or, we might want to guarantee that all the hypotheses examined have true errors close to their observed errors.

It turns out that there are some convenient inequalities for calculating bounds on the probability of getting bad estimates of hypotheses' real error rates: Hoeffding and Chernoff bounds. We now examine these bounds, and describe results that can be derived from them.

What follows is the same as the handout "Probabilistic Inequalities".

5 Probabilistic inequalities

A common question that comes up in machine learning is: "Given some fixed hypothesis h, how much data do I need to see so that I can be confident that its observed error will be near to its true error?" We have already seen this as: "What's the chance that a given hypothesis of true error ϵ or more will have observed error of 0?"

In this case, from first principles we calculated the probability to be $(1-\epsilon)^m$, where m is the number of examples. We then noticed that if there are $|H|$ hypotheses under consideration, and this quantity is $\delta/|H|$, then there is at most a δ chance there exists *any* hypothesis that fools us. Solving for m, we got that $\frac{1}{\epsilon}\left[\ln(|H|) + \ln(\frac{1}{\delta})\right]$ examples suffice.

More generally, suppose we consider a hypothesis with true error p, and let $q = 1 - p$. If we see m examples, then the expected number of mistakes is pm. The *standard deviation* of this quantity is $\sqrt{pqm}$, or equivalently, the standard deviation of the *fraction* of mistakes made is $\sqrt{pq/m}$. We can now use a convenient rule for independent identically distributed Bernoulli trials, which in our terminology is:

$$\Pr[|\text{observed error} - \text{true error}| > 1.96\sigma] < 0.05,$$

where $\sigma = \sqrt{pq/m}$ is the standard deviation. For instance, if we want with 95% confidence for our true and observed errors to differ by only ϵ, then we need to see only $4p(1-p)/\epsilon^2$ examples (approximating 1.96^2 by 4).

The above rule is convenient for testing out a single hypothesis on test data. More generally, we'd like a rule that can be applied for any desired confidence level (like the first case above) but holds even when the observed error is not 0 (like the second case above). For instance, we might want to prove uniform convergence results, or we might just want to test out several hypotheses.[1] Luckily, there are some convenient inequalities for doing this, known as Hoeffding and Chernoff bounds. These bounds state the following. (Actually, they hold for a variety of situations we won't describe here; a good book that discusses how these are derived and many other things is Alon and Spencer's *The Probabilistic Method.*)

Consider a hypothesis with true error rate p (or a coin of bias p) observed on m examples (the coin is flipped m times). Let S be the number of observed errors (the number of heads seen) so S/m is the observed error rate.

Hoeffding bounds state that for any $\epsilon \in [0,1]$,

1. $\Pr[\frac{S}{m} > p + \epsilon] \leq e^{-2m\epsilon^2}$, and
2. $\Pr[\frac{S}{m} < p - \epsilon] \leq e^{-2m\epsilon^2}$.

Chernoff bounds state that under the same conditions,

1. $\Pr[\frac{S}{m} > p(1+\epsilon)] \leq e^{-mp\epsilon^2/3}$, and
2. $\Pr[\frac{S}{m} < p(1-\epsilon)] \leq e^{-mp\epsilon^2/2}$.

Let's do a quick example of using Hoeffding bounds. Suppose we'd like to say that with high probability $(1-\delta)$, all hypotheses in H have their observed error within ϵ of their true error. To achieve this, we just need for each hypothesis individually to have confidence parameter $\delta/|H|$. So, we set $2e^{-2m\epsilon^2} \leq \delta/|H|$, or:

$$2m\epsilon^2 \geq \ln(2|H|/\delta).$$

We can now either solve for ϵ as a function of m and δ or solve for m as a function of ϵ and δ. Solving for ϵ, we find that our goal is achieved for ϵ satisfying:

$$\epsilon \geq \sqrt{\frac{\ln(2|H|/\delta)}{2m}}.$$

This says roughly that it suffices to go $\sqrt{2\ln(2|H|/\delta)}$ standard deviations to get our desired confidence. Actually, this is a nice way of thinking about the meaning of Hoeffding bounds:

[1] Say we test out one hypothesis and find its observed error to be too high using the 0.05 confidence measure. Then we run our algorithm some more, get another hypothesis, test that one with 0.05 confidence, and find it also is too high. Eventually we get one that passes our test. Are we in good shape?

they say that if we want to replace the "0.05" in the earlier equation with some arbitrary small δ', then it suffices to go about $\sqrt{2\ln(2/\delta')}$ standard deviations.

Solving for m we get:

$$m \geq \frac{1}{2\epsilon^2}\ln(2|H|/\delta) = \frac{1}{2\epsilon^2}[\ln(|H|) + \ln(2/\delta)].$$

This is a lot like the first PAC bound, but we are now quadratic in $1/\epsilon$.

How can we get rid of that pesky ϵ^2? We can do that by relaxing our goal, to say that for hypotheses whose true error is greater than ϵ, we are satisfied if their observed error comes just within a *factor of 2*. If you think about it, this is often all we need since we don't care so much about the high-error hypotheses. Now we can use Chernoff bounds.

If $p \geq \epsilon$ and we only need to be within a factor of 2 with confidence δ', we can solve that it suffices to have $m \geq \frac{8}{p}\ln(2/\delta')$ (and we can replace p with ϵ for a weaker bound). On the other hand, if $p < \epsilon$, we can rewrite our additive error goal by saying that we want the observed error to be no more than $p(1 + \epsilon/p)$, which by Chernoff bounds implies that $m \geq \frac{3p}{\epsilon^2}\ln(1/\delta')$ examples suffice (and we can replace $\frac{3p}{\epsilon^2}$ with $\frac{3}{\epsilon}$ for a weaker bound). So, setting $\delta' = \delta/|H|$ we get that:

$$m \geq \frac{8}{\epsilon}[\ln(|H|) + \ln(2/\delta)]$$

examples suffice.