COS 511: Foundations of Machine Learning

Rob Schapire
Scribe: David Weiss

Lecture #8
March 2, 2006

1 A More General Set of Tools

In the previous lecture, we introduced a generalized form of the PAC learning model that allows for the possibility that we are not able or do not wish to find a consistent hypothesis during learning. Before we delve further into analysis of this model, we will first develop a set of general tools that allow us to describe more precisely the convergence of a random variable to its expectation.

1.1 Abstract Case & Hoeffding's Inequality

Given a set of *i.i.d* random variables $X_1, \ldots, X_m$, where X_i is always bounded ($X_i \in [0, 1]$) and all X_i are drawn from the same distribution, let $p = E[X_i]$ be the common expectation of all X_i. Now, we define the *empirical average* $\hat{p}$ as follows:

$$\hat{p} = \frac{1}{m} \sum X_i.$$

Thus, in the development of our general techniques, we wish to answer the question: *how quickly does $\hat{p} \to p$?*

As it turns out, an easy first stab at an answer to this question is given by the theorem we discussed in the last class, *Hoeffding's Inequality:*

Hoeffding's Inequality. *Assume random variables $X_1, \ldots, X_m$ are i.i.d. Let*

$$p = E[X_i], \qquad X_i \in [0, 1], \qquad \hat{p} = \frac{1}{m} \sum X_i, \qquad \epsilon \geq 0$$

Then

$$Pr[|\hat{p} - p| \geq \epsilon] \leq 2e^{-2\epsilon^2 m}.$$

From this inequality we can see that $\hat{p}$ converges to p exponentially fast in our sample size, m.

Intuitively, we can imagine a graph of the r.v. $\hat{p}$ and consider the areas of the distribution where $|\hat{p} - p| \geq \epsilon$; we can then interpret Hoeffding's inequality as a bound on the probability that $\hat{p}$ occurs within these regions on the outskirts of the distribution. Thus, Hoeffding's inequality is also referred to as a *tail bound* or a *concentration inequality* of the distribution over $\hat{p}$.

However, as we shall see in the next section, we can prove a stronger, more powerful, yet less well-known result, of which Hoeffding's Inequality is a special case.

1.2 Relative Entropy

In this section, we will prove the following stronger bound on the convergence of a random variable to its expectation. Bounds of this general form, including Hoeffding's inequality, are generically known as *Chernoff Bounds.*

Theorem 1.2.1. *Assume random variables $X_1, \ldots, X_m$ are i.i.d. Let*

$$p = E[X_i], \qquad X_i \in [0,1], \qquad \hat{p} = \frac{1}{m}\sum X_i, \qquad \epsilon \geq 0.$$

Then

$$Pr[\hat{p} \geq p + \epsilon] \leq e^{-RE(p+\epsilon \,||\, p)m}$$
$$Pr[\hat{p} \leq p - \epsilon] \leq e^{-RE(p-\epsilon \,||\, p)m}$$

where $RE(p + \epsilon \,||p)$ is defined as the relative entropy *between $p + \epsilon$ and p.*

Relative Entropy is a means of measuring the distance between two distributions from an information theory perspective. In other circles, it is also known as *Kullbach-Liebler Divergence* or *I-divergence.* However, it is an extremely prevalent and useful concept, so we will now take a detour into information theory to define and understand exactly what it is we are trying to prove.

1.2.1 Entropy

In information theory, a common setup is as follows. One subject, *Alice*, wishes to send a message across a wire or *channel* to a recipient, *Bob.*

$$A \xrightarrow{channel} B$$

Alice can only send one bit at a time, but she wishes to send one of N possible *messages* to Bob without error: say, the 26 letters of the alphabet. We must therefore assign each letter a unique binary representation. A naive implementation (Figure 1) would be to see that bit strings of length 5 will provide enough unique representations to code each letter uniquely:

letter	**encoding**
a	00000
b	00001
$\vdots$	$\vdots$
z	01110

Figure 1: A naive, inefficient encoding of the alphabet.

The problem with the naive encoding of the alphabet is that it is terribly inefficient: we know that the message e is likely to appear many more times than the message z, so we can reduce the expected message length by using fewer than 5 bits to encode e and (if necessary) more than 5 bits to encode z (Figure 2).

In general, if $P(x)$ = probability of sending message x, then it can be shown that the optimal number of bits to encode message x is given by

$$\text{optimal \# of bits} = \lg \frac{1}{P(x)}.$$

Thus the expected number of bits to encode any message, called the *entropy* of the distribution over messages P, is given by

$$E[\text{message length}] = \sum_x P(x) \lg \frac{1}{P(x)} \qquad \leftarrow \text{“entropy”}$$

letter	**encoding**
a	01
b	0010
$\vdots$	$\vdots$
z	11111

Figure 2: A more efficient encoding of the alphabet.

From this definition, we can see that the entropy will be largest as P approaches the uniform distribution. (By convention, $0 \lg 0$ is defined to be 0.)

Suppose now that we don't actually know P, and instead have a "mistaken" belief that $P(X) = Q(X)$, where $Q(X)$ is the *believed* P. Instead of assigning our messages $\lg 1/P(X)$ bits, we are going to assign $\lg 1/Q(X)$ bits, which can only be less efficient than an encoding according to P. Our new expected message length will be

$$E[\text{message length}] = \sum_x P(X) \lg \frac{1}{Q(X)} \geq \mathit{entropy}(P).$$

To get an idea of how much our poor estimate Q is hurting us, we look at the *difference* between the entropy of P and the expected message length according to Q:

$$\begin{aligned} \sum_x P(X) \lg \frac{1}{Q(X)} - \sum_x P(X) \lg \frac{1}{P(X)} &= \sum_x P(X) \lg \frac{P(X)}{Q(X)} \\ &= \mathit{RE}(P||Q) \end{aligned} \tag{1}$$

where $\mathit{RE}(P||Q)$ denotes the *relative entropy* between distributions P and Q. This will be our measure of the "distance" between two distributions. Note that this value is always positive, since the expected message length using P is always less than the expected message length using Q. Also note that although information theorists use lg as a standard logarithm, we will use for convenience ln that is simply off by a constant factor.

As a final note, we introduce a shorthand so that we may write $\mathit{RE}(p||q)$ even though p and q are fixed scalars and not probability distributions. Implicitly, we will take $\mathit{RE}(p||q)$ to refer to the distributions P and Q defined by two events with probabilities $(p, 1-p)$ and $(q, 1-q)$. Thus, equation (1) above becomes

$$\mathit{RE}(p||q) = \mathit{RE}\left((p, 1-p) \,||\, (q, 1-q)\right) = p\left(\ln \frac{p}{q}\right) + (1-p)\left(\ln \frac{1-p}{1-q}\right)$$

which we abbreviate using the shorthand on the leftmost side of the equation. This allows us to write in Theorem 1.2.1 the construct $\mathit{RE}(p+\epsilon||p)$ with clarity.

1.2.2 A False Start Proof of RE Theorem

Now that we have defined all of our terms, we shall prove Theorem 1.2.1. To recap, we aim to derive the following inequality:

$$Pr[\hat{p} \geq p + \epsilon] \leq e^{-\mathit{RE}(p+\epsilon||p)m}. \tag{2}$$

One simple approach to this problem is to apply the Markov Inequality and use the fact that $E[\hat{p}] = p$. The Markov Inequality states, for any r.v. X,

$$Pr[X \geq \delta] \leq \frac{E[X]}{\delta}.$$

We can then rewrite (2):

$$Pr[\hat{p} \geq p + \epsilon] \leq \frac{E[\hat{p}]}{p + \epsilon} = \frac{p}{p + \epsilon}$$

And we have hit a dead end. Unfortunately, this weak bound is useless; has the Markov Inequality forsaken us? Thankfully, it has not. We can use a "monotonic trick" to allow us to use the Markov inequality to produce better results.

1.2.3 A Real Proof of RE Theorem

Proof. Let $q = p + \epsilon$ and choose an arbitrary $\lambda > 0$. Now,

$$Pr[\hat{p} \geq q] = Pr[e^{\lambda m \hat{p}} \geq e^{\lambda m q}]$$

where m is the number of sample points in the definition of $\hat{p}$. Note that this step is valid because $e^{\lambda m x}$ is a monotonically increasing function of x. Now, if we apply Markov, we get

$$Pr[\hat{p} \geq q] \leq \frac{E[e^{\lambda m \hat{p}}]}{e^{\lambda m q}} = e^{-\lambda m q} E[e^{\lambda m \hat{p}}]. \tag{3}$$

Then

$$\begin{aligned} E[e^{\lambda m \hat{p}}] &= E\left[\exp\left(\lambda m (1/m) \sum_i X_i\right)\right] = E\left[\exp\left(\lambda \sum_i X_i\right)\right] \\ &= E\left[\prod_i \exp\left(\lambda X_i\right)\right] \\ &= \prod_i E[e^{\lambda X_i}] \end{aligned}$$

because all X_i are independent. At this point, we know that $X_i \in [0,1]$, so we can define a line $y(x) = (1 - x) + x e^{\lambda}$ such that $y(x) \geq e^{\lambda x}$ for $x \in [0,1]$. Thus we can rewrite the last equation as an inequality, and then use linearity of expectation to simplify since $y(x)$ is a linear function of x:

$$\begin{aligned} E[e^{\lambda m \hat{p}}] &\leq \prod_i E\left[(1 - X_i) + X_i e^{\lambda}\right] \\ &= \prod_i \left[(1 - p) + p e^{\lambda}\right] \\ &= (1 - p + p e^{\lambda})^m \end{aligned}$$

where the last steps use linearity of expectation and the fact that $E[X_i] = p$. Finally, we can substitute the previous equation back into (3):

$$\begin{aligned} Pr[\hat{p} \geq q] &\leq e^{-\lambda m q} (1 - p + p e^{\lambda})^m \\ &\leq \left[e^{-\lambda q}\left(1 - p + p e^{\lambda}\right)\right]^m. \end{aligned} \tag{4}$$

We now have an exponential bound on $Pr[\hat{p} \geq q]$, but it depends on an arbitrary $\lambda > 0$. If we let

$$\phi(\lambda) = \ln \left[e^{-\lambda q}(1 - p + pe^{\lambda}) \right]$$

then we can solve for the maximum of $\phi(\lambda)$ by setting $\frac{d\phi(\lambda)}{d\lambda} = 0$ and solving for λ. In this case, the solution is given by

$$\lambda = \ln \left(\frac{q(1-p)}{p(q-1)} \right), \qquad \phi(\lambda) = -RE(q \mid\mid p).$$

Finally, we conclude:

$$Pr[\hat{p} \geq q] \leq e^{\phi(\lambda)m} = e^{-RE(q \mid\mid p)m}$$

And we have proved the first half of Theorem 1.2.1. To prove the latter half, we simply substitute X_i with a new r.v. $1 - X_i$ and plug it into the first bound. □

What do we make of this new theorem? First of all, Hoeffding's inequality now follows from the fact that $RE(p + \epsilon || p) \geq 2\epsilon^2$ for all p and ϵ. It can be shown that, as p goes towards either 0 or 1 (away from 1/2), $RE(p + \epsilon || p) \rightarrow \epsilon$, rather than the ϵ^2 growth factor derived from Hoeffding's Theorem. Also, keep in mind once again that although these are general probabilistic techniques, we will ultimately be applying them to our inconsistent PAC learning model.

1.3 McDiarmid's Inequality

Although we don't have time to prove it, McDiarmid's inequality provides a very useful generalization of Hoeffding's Inequality.

McDiarmid's Inequality. *Let $X_1, \ldots, X_m$ be independent but* not necessarily *identically distributed random variables. Let $f(x_1, \ldots, x_m)$ be a function such that if we change only one of the parameters x_i to a new value x'_i, then the function changes by at most c_i:*

$$|f(x_1, \ldots, x_i, \ldots, x_m) - f(x_1, \ldots, x'_i, \ldots, x_m)| \leq c_i \qquad \forall x_1, \ldots, x_m, x'_i$$

Then

$$Pr\left[f(X_1, \ldots, X_m) \geq E[f(X_1, \ldots, X_m)] + \epsilon\right] \leq \exp\left(\frac{-2\epsilon^2}{\sum_i c_i^2} \right).$$

As a special case, we can use McDiarmid's Inequality to derive a proof of Hoeffding's Inequality in a quick manner:

Proof. Let us define $\hat{p}$ such that

$$\hat{p} = \frac{1}{m} \sum_{i=1}^{m} X_i = f(X_1, \ldots, X_m)$$

Because $X_i \in [0, 1]$, the most that X_i can be changed is from 0 to 1, and the maximum change in f is bounded by $c_i = 1/m$. Thus $\hat{p}$ satisfies the requirements for f in Theorem

1.3. Now, we simply apply the theorem:

$$\begin{aligned} Pr[f \geq E[f] + \epsilon] &\leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^m c_i^2}\right) \\ Pr[\hat{p} \geq p + \epsilon] &\leq \exp\left(\frac{-2\epsilon^2}{m(1/m)^2}\right) \\ &= \exp\left(\frac{-2\epsilon^2}{1/m}\right) \\ &= e^{-2\epsilon^2 m} \end{aligned}$$

□

2 Back to PAC

In the previous class, we showed that if we could prove that $\forall h \in H$,

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

then learning will be possible in our inconsistent PAC model. At this point, however, we have all the tools we need to prove this result.

Theorem 2.1. *Given m examples and a finite hypothesis space H, then, with probability $1 - \delta$, if*

$$m = O\left(\frac{\ln|H| + \ln(1/\delta)}{\epsilon^2}\right)$$

then

$$\forall h \in H : \quad |err(h) - \hat{err}(h)| \leq \epsilon.$$

Proof. Fix $h \in H$. Now we define X_i such that

$$X_i = \begin{cases} 1 \text{ if } h \text{ is correct on example } i \\ 0 \text{ otherwise.} \end{cases}$$

Furthermore, $E[X_i] = err(h)$, the probability of making an incorrect guess, and $\hat{err}(h) = \frac{1}{m}\sum X_i$ is the empirical error. We now apply Hoeffding's Inequality:

$$Pr\left[|\hat{err}(h) - err(h)| > \epsilon\right] \leq e^{-2\epsilon^2 m}. \tag{5}$$

Since (5) is true for any fixed h, we can use the Union Bound to bound the probably of *any* h:

$$Pr\left[\exists h \in H : |\hat{err}(h) - err(h)| > \epsilon\right] \leq 2|H|e^{-2\epsilon^2 m} = \delta$$

Setting this result equal to δ and solving for m, we then have

$$m = O\left(\frac{\ln|H| + \ln(1/\delta)}{\epsilon^2}\right)$$

□

2.1 Important Implications of New Bounds

We discover an important difference between our new result and the result for a purely consistent h when we solve for the error ϵ:

$$|err(h) - \hat{err}(h)| \leq \sqrt{\frac{\ln 2|H| + \ln(1/\delta)}{2m}}$$

which implies

$$err(h) \leq \hat{err}(h) + \sqrt{\frac{\ln 2|H| + \ln(1/\delta)}{2m}}.$$

Ignoring all other parameters besides the size of the sample m, we find that whereas our error was previously bounded roughly by $1/m$, we are now bounded only by $1/\sqrt{m}$ - this is quite a significant factor. Intuitively, to perform twice as well, we previously needed twice as many examples, but we now need four times as many examples!

Our new weakness in the bound derives from the ϵ^2 term in Hoeffding's inequality. As was hinted at earlier, the problem is that Hoeffding's inequality becomes weaker as p approaches the extremities of 0 and 1, since one would expect less "fluctuations" in the value of $\hat{p}$ in these cases. This was our justification to use the stronger RE bounds.

However, there is another significant intuitive result to be gained from our bound on h in the inconsistent PAC model: our true error $err(h)$ is dependent on both the empirical error $\hat{err}(h)$ *as well as* the $O(\sqrt{(|h| + \ln(1/\delta))/m})$ term. Once again, we have a trade-off between hypothesis complexity and the true error of our hypothesis, but this time we are only able to measure and train our hypothesis on the empirical term $\hat{err}(h)$. When $|h|$ is very small, $err(h)$ will be dominated by $\hat{err}(h)$, but at some point when $|h|$ increases greatly the O term will dominate and cause true error $err(h)$ to increase, even if $\hat{err}(h)$ is decreasing during training!

This fundamental trade-off in theory describes a very real phenomenon in the practice of machine learning called *overfitting*. Overfitting is demonstrated with real-world data in Figure 3. As the complexity of the hypotheses increases, the probability of finding a consistent hypothesis increases and so $\hat{err}(h)$ approaches 0. However, at some point, the O term begins to dominate and $err(h)$ reaches a minimum after which it begins to rise again.

Practically, overfitting is a difficult problem because in many cases only $\hat{err}(h)$ can be observed directly. There are historically at least three main approaches to solving this problem that are common in machine learning:

- "Structural Risk Minimization" - this solution attempts to figure out the exact value of the theoretical bounds so that error can be minimized directly. However, although theory predicts the existence of the bound, due to a prevalence of unknown constants the O term must be estimated empirically, which is prone to a multitude of inaccuracies and difficulties.

- "Cross-Validation" - this solution involves separating a segment of the training data for the algorithm to be used as a "test" sample; especially if multiple subsets of the data are used as both testing and training data, a fairly effective estimate of $err(h)$ in addition to $\hat{err}(h)$ can be found. However, cross-validation requires many training examples and is often demanding on time, data, and computational resources.

- New Algorithms - this approach does not involve a specific solution, but generally seeks to find algorithms that can learn hypotheses that are robust and less vulnerable to the dangers of overfitting.

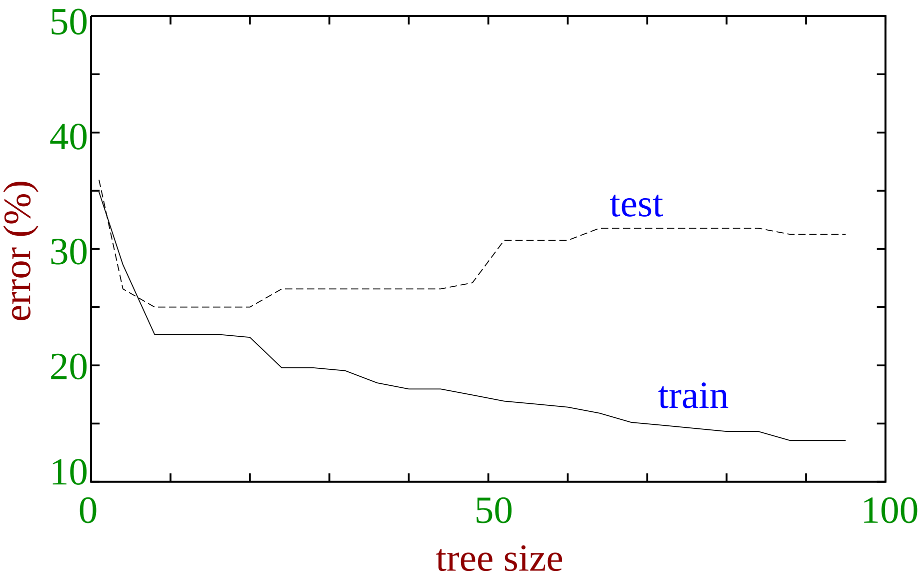

Figure 3: An example of over-fitting from real world data. As the complexity of the hypothesis (in this case, a decision tree) increases, $\hat{err}(h)$ approaches 0, but $err(h)$ reaches a minimum and begins to rise.

3 Boosting

This largely marks the end of the general sample complexity portion of the class. In the next class and coming weeks, we will examine various learning algorithms in rigorous mathematical detail.

The first such algorithm we will be looking at, Boosting, grew out of the PAC learning model. The question Boosting tries to answer is: PAC learning depends on making ϵ arbitrarily small, but suppose this is impossible? Is there even so a way to *force* error to 0, despite this limitation? The answer is that it is possible – this is the basis of Boosting algorithms.

To start off our Boosting analysis, we say that a concept class C is *weakly learnable* if $\exists$ an algorithm A and $\exists \gamma > 0$ such that, given m training examples, A finds a hypothesis $h \in H$ such that

$$\forall c \in C, \quad \forall D, \quad \delta > 0 :$$

$$\epsilon \geq 1/2 - \gamma$$

$$Pr[err(h) > \epsilon] \leq \delta$$

Next week, we will answer the questions: is weak learning equivalent to strong learning? i.e., Can a weak learner A be converted to a strong learner?