COS 511: Foundations of Machine Learning

Rob Schapire
Scribe: Jeffrey Bigham

Lecture #3
February 11, 2003

1 Comment on Union Bound

In the previous lecture in which probabilities were discussed, the Union Bound property of random variables was discussed. In its basic form, this rule in which A and B are random variables states the following:

$$Pr[A \vee B] \leq Pr[A] + Pr[B]$$

If A and B are disjoint then this will result in equality. The idea extends to more variables in the logical way:

$$Pr[A_1 \vee A_2 \vee \ldots] \leq \sum_i Pr[A_i]$$

2 Review of PAC Model

2.1 The PAC Model

A class C is PAC learnable by H if:
$\exists$ algorithm A
$\forall$ $c \in C$
$\forall$ distributions D on X
$\forall$ $\epsilon > 0$
$\forall$ $\delta > 0$
A takes $m = poly(1/\epsilon, 1/\delta)$ examples from D and produces hypothesis h $\in$ H subject to $Pr[err_D(h) \leq \epsilon] \geq$ 1 - δ

2.2 Terminology Related to PAC

Definitions:
C = target class or target space
H = hypothesis class or hypothesis space
c = target concept
D = target distribution
ϵ = accuracy parameter
δ = confidence parameter
S = training sample

2.3 Concepts of Error

2.3.1 True or Generalization Error

$err_D(h) = Pr_{x \sim D}[h(x) \neq c(x)]$, where h is fixed, x is random

2.3.2 Training Error

training error (empirical error) $= \frac{1}{m}|\{i: c(x_i) \neq h(x_i)\}|$, where h = output hypothesis

3 Example Revisited

In the previous lecture the example of positive half lines on the Real number line was offered as an example of a PAC learnable concept. This example operates in the domain of the Real $\Re$ numbers. Classes and hypotheses can be represented in this domain by a single point, with all numbers to the right of that point being a positive example and all numbers to the left of that point being a negative example. c represents the target class and h represents a given hypothesis.

$$X = \Re$$

$$H = C = \{positive\ half\ lines\ \}$$

The region of the number line that exists between c and h is the region representing the error of the hypothesis. During learning the algorithm does not know the distribution defined by the class that its goal is to learn, and, therefore, the learned hypothesis h can be off in one of two ways: Its representative value can either be too large (to the right of c on the number line) or it can be too small (to the left of c on the number line). These two bad cases will be labelled B^+ and B^- respectively.

B^+: h is more than ϵ to the right of c, where ϵ is in terms of distribution weight.
B^-: h is more than ϵ to the left of c, where ϵ is in terms of the distribution weight.

The bad case, B^+, can only occur if no training examples occur in the region, denoted R^+, between c and an h that is to the right of c. Symmetrically, the bad case B^-, can only occur if no training examples occur in the region, denoted R^-, between c and an h that is to the left of c. The reason for this is that if such a training example occurred, it would shrink the corresponding region to the region between c and that example. Because of the setup of this problem, it follows directly that in this problem errors can only be found on one side of the true class c.

The regions R^+ and R^- are defined as the region formed by starting at c on the number line and then moving right or left, respectively, until ϵ of the distribution is covered. Formally that is:

$$Pr_{x\sim D}[x \,\epsilon\, R^+] = \epsilon$$

To bound the probability of the bad case B^+ happening we can therefore calculate the probability of none of the training examples occurring in R^+.

$$Pr[B^+] \leq Pr[x_1 \notin R^+ \wedge \cdots \wedge x_m \notin R^+]$$

Because the x_i's are independent, this reduces to:

$$Pr[B^+] \leq \prod_{i=1}^{m} Pr[x_i \notin R^+]$$

Because each probability in the above product is at most $(1-\epsilon)$ the above formula can be simplified to:

$$Pr[B^+] \leq (1-\epsilon)^m$$

The same bound on the probability of B^- is obtained by a symmetric argument.

The probability that a given hypothesis h is ϵ-bad is then bounded by the following:

$$Pr[err_D(h) > \epsilon] \leq Pr[B^+ \vee B^-]$$

$$Pr[err_D(h) > \epsilon] \leq Pr[B^+] + [B^-]$$

$$Pr[err_D(h) > \epsilon] \leq 2(1-\epsilon)^m$$

then using the following useful inequality:

$$\forall x \in \Re : 1 + x \leq e^x$$

this can be simplified to:

$$Pr[err_D(h) > \epsilon] \leq 2e^{-\epsilon m}$$

Thus, rearranging things slightly we obtain the result that

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

if

$$m \geq \frac{1}{\epsilon} \ln(\frac{2}{\delta})$$

If m is at least that large then $Pr[h$ is ϵ-bad$]$ is less than δ. In practice, the data is usually given, so considering the error-rate is often useful. If we let ϵ be such that $2e^{-\epsilon m} = \delta$, then ϵ will be an upper bound on the error rate of h, so our argument shows that, with probability at least $1-\delta$,

$$err_D(h) \leq \frac{1}{m} \ln(\frac{2}{\delta})$$

for any h consistent with a sample of size m.

4 Sketched Examples

To further the understanding of how the PAC learning model is applied in practice, two other examples were sketched.

4.1 Intervals

This example was very much like the positive half-line example from above, but, instead of merely being able to divide the number line into a negative region on the left and a positive region on the right, classes and hypotheses in this space were represented by intervals with examples in the interval classified as positive and those outside of the interval classified as negative.

$X = \Re$

$H = C = Intervals$

Someone noticed that if the intervals were cut in half, this problem could be viewed as two problems very similar to the positive half-line example presented above. The argument would always begin at the endpoint formed when the interval was divided in half and from there an argument symmetric to the one above could be presented, with the errors at the regions approaching the endpoints required to be less than $\frac{\epsilon}{2}$.

4.2 Rectangles

Extending the intervals example to two dimensions results in a problem whose classes and hypotheses are defined by rectangles. In this example, the class in this problem is represented by an unknown rectangle whose axes are parallel to the axes of the coordinate system. A given hypothesis will be accurate in the areas in which its area overlaps the area of the true class' rectangle. The observation presented by someone in the class is that in this case you can once again provide a symmetric argument to that proposed above for the intervals and positive half lines, by splitting the area that defines the error into four rectangles and requiring that each be less than $\frac{\epsilon}{4}$.

This example is discussed on pages 1-6 of the course textbook, An Introduction to Computational Learning Theory, by Kearns and Vasirani.

5 Moving Toward A General Result

In these last examples, we had to give arguments that were peculiar to the particular classes we were trying to learn. In this section, a more general approach is developed in which we judge the hypothesis by how well it explains the examples that have already been seen.

5.1 A Result for Finite Hypothesis Spaces

We assume now that the set of all hypotheses H is finite. The goal is to choose any consistent $h \in H$ and show that it is PAC compliant for a sample size of m. Let

$$B = \{h \in H : err_D(h) > \epsilon\}$$

Algorithm A outputs any hypothesis $h_A \in H$ consistent with the sample.

$$Pr[h_A \in B] \leq Pr[\exists h \in B : h \text{ consistent with sample}]$$

If we fix $h \in B$

$$Pr[h \text{ consistent}] = Pr[h(x_1) = c(x_1) \wedge \cdots \wedge h(x_m) = c(x_m)]$$

$$= \Pi_{i=1}^{m} Pr[h(x_i) = c(x_i)]$$

$$< (1 - \epsilon)^m$$

Using this we can obtain the following central result as follows:

$$Pr[\exists h \in B: \ h\ consistent] \le |B|(1-\epsilon)^m \le |H|(1-\epsilon)^m$$

$$\le |H|e^{-\epsilon m}$$

$$\le \delta$$

$$m \ge \frac{1}{\epsilon}(ln|H| + ln\frac{1}{\delta})$$

$$err(h) \le \frac{1}{m}(ln|H| + ln\frac{1}{\delta})$$

NOTES:

- bound only logarithmic in $\frac{1}{\delta}$
- If we were to represent the h's in terms of bits, we would need $lg|H|$ bits, so $ln|H|$ is just represented by a number of bits.

5.2 Occam's Razor

Occam's Razor: Simply stated, Occam's Razor is the general principle that using a hypothesis space that is more complicated than required is not as good as using a simpler one.

The bound above gets at the idea of Occam's Razor.
To illustrate this we revisited the Monotone Conjunction Example from the first lecture:

011010	+
101010	+
− − − − −	
001010	

In this case n is equal to the number of bits in the binary string:

$$ln|H| = n(ln2)$$

In the case of DNF we get:

$$|H| = 2^{2^n}$$

and end up with an exponential after $ln|2^{2^n}|$.

5.3 Intuitive Results of Occam's Razor Analysis

This suggests that PAC-learning is possible whenever you are able to write down a hypothesis in a reasonable number of bits and is only hard because of computational reasons instead of statistical reasons or because of a lack of information.

This model doesn't handle infinite hypothesis spaces and one approach may be to discretize but it is unclear if this approach will always work. This applies directly to the positive half line example from the beginning of lecture.

Intuitions:

- more data means the error goes down
- more hypotheses that you consider, the more likely you'll choose one that seems good but isn't
- the more you know ahead of time about the data, the smaller you can make H

6 False Argument for Getting Rid of H Dependence

6.1 The Faulty Analysis

if h_A $\epsilon - bad$ then:

$$Pr[h_A\, consistent] = Pr[h_A(x_1) = c(x_1) \wedge \cdots \wedge h_A(x_m) = c(x_m)]$$

$$= \Pi_{i=1}^{m} Pr[h_A(x_i) = c(x_i)]$$

$$\leq (1-\epsilon)^m \leq e^{-\epsilon m} \leq \delta$$

$$m \geq \frac{1}{\epsilon} ln \frac{1}{\delta}$$

6.2 Mistake

The main mistake here is that h_A is now a random variable, so the reduction to a product in step two does not follow since the events $h_A(x_i) \neq c(x_i)$ are no longer independent of one another.