COS 511: Theoretical Machine Learning

Lecturer: Rob Schapire
Scribe: Yongxin (Taylor) Xi

Lecture #4
February 13, 2008

1 Theorem

1.1 Statement

Thm: Say algorithm A finds a hypothesis $h_A \in \mathcal{H}$ consistent with m examples, where $m \geq \frac{1}{\epsilon}(\ln|\mathcal{H}| + \ln\frac{1}{\delta})$. Then $\Pr[err_D(h_A) > \epsilon] \leq \delta$.

This theorem has an assumption that the space of hypotheses is finite, and that both the training examples and testing examples are generated independently according to distribution D. It gives an upper bound on the number of examples required relative to the error measure, confidence parameter and the logarithm of the size of hypothesis space. According to the theorem, simpler $\mathcal{H}$ leads to less data needed to learn.

1.2 Proof

Pf: We aim at upper-bounding $\Pr[h_A\ \epsilon\text{-bad}]$, which is the probability that the hypothesis generated by A has a performance worse than ϵ.

Define $\mathcal{B} = \{h \in \mathcal{H} : h\ \epsilon\text{-bad}\}$. Note that set $\mathcal{B}$ is not random; it is uniquely determined by the target concept c, the hypothesis class $\mathcal{H}$, the target distribution D, and the value of ϵ.

$$
\begin{aligned}
\Pr[h_A\ \epsilon\text{-bad}] &= \Pr[h_A\ \textit{consistent} \wedge h_A\ \epsilon\text{-bad}] && (1)\\
&\leq \Pr[\exists h \in \mathcal{H} : h\ \textit{consistent} \wedge \epsilon\text{-bad}] && (2)\\
&= \Pr[\exists h \in \mathcal{B} : h\ \textit{consistent}] && (3)\\
&= \Pr[\bigvee_{h\in\mathcal{B}} h\ \textit{consistent}] && (4)\\
&\leq \sum_{h\in\mathcal{B}} \Pr[h\ \textit{consistent}] && (5)\\
&= \sum_{h\in\mathcal{B}} \Pr[(h(x_1) = c(x_1) \wedge \ldots h(x_m) = c(x_m))] && (6)\\
&= \sum_{h\in\mathcal{B}} \prod_{i=1}^{m} \Pr[h(x_i) = c(x_i)] && (7)\\
&\leq \sum_{h\in\mathcal{B}} (1-\epsilon)^m && (8)\\
&= |\mathcal{B}|(1-\epsilon)^m && (9)\\
&\leq |\mathcal{H}|(1-\epsilon)^m && (10)\\
&\leq |\mathcal{H}|e^{-\epsilon m} && (11)\\
&\leq \delta && (12)
\end{aligned}
$$

Equation (2) uses the following fact: Let A, B be two events. If $A \Rightarrow B$ then $\Pr[A] \leq \Pr[B]$. in this case, the event appearing in the probability in Eq(1) implies the corresponding event in Eq(2). Also, note that in equation (2), whether or not h is consistent is dependent on the m training examples; however, whether h is ϵ-bad or not does not depend on the training set.

Equation (5) uses the union bound rule. Equation (6) is just the definition of consistency, while (7) follows the independence assumption of the m examples drawn from distribution. Because h is just one hypothesis from the set $\mathcal{B}$, i.e., h is ϵ-bad, we then obtain equation (8). Equation (10) is straightforward, since $\mathcal{B} \subseteq \mathcal{H}$.

1.3 Intuition

The theorem tells us that the number of examples needed for learning is related to the size of the hypothesis space. If we can compress this space, then learning could be easier. It is a theorem about how much data you need, and it is not about the efficiency of the learning algorithm, which will be addressed later. When we look at this upper bound of examples, it is a very loose bound thus quantitatively not good. However, it is qualitatively good, because it captures a general relation between learning performance and the size of the hypothesis space (complexity of the rules), and the number of training examples.

If we fix the number of training examples to be m, then with probability $(1 - \delta)$, $err_D(h_A) \leq \frac{\ln|\mathcal{H}| + \ln\frac{1}{\delta}}{m}$, which is inverse proportional to m.

This informs us that the more data we have, the lower an upper bound of error we achieve. Secondly, if $|\mathcal{H}|$ is large, then the algorithm will tend to find a poor yet consistent hypothesis from it. So, the more you know about the rules (thus smaller $|\mathcal{H}|$), the better the algorithm learns.

2 A plausible argument

Now let us try deriving an upper bound for m without using $|\mathcal{H}|$. Say h_s is the hypothesis output by the algorithm A given sample S, which contains m examples.

$$
\begin{aligned}
\Pr[err(h_s > \epsilon) \mid h_s \textit{ consistent}] &= \frac{\Pr[err(h_s) > \epsilon \wedge h_s \textit{ consistent}]}{\Pr[h_s \textit{ consistent}]} && (13)\\
&= \Pr[err(h_s) > \epsilon \wedge h_s \textit{ consistent}] && (14)\\
&= \Pr[h_s \textit{ consistent} \mid err(h_s) > \epsilon] \Pr[err(h_s) > \epsilon] && (15)\\
&\leq \Pr[h_s \textit{ consistent} \mid err(h_s) > \epsilon] && (16)\\
&= \Pr[(h_s(x_1) = c(x_1) \wedge \ldots \wedge h_s(x_m) = c(x_m)) \mid err(h_s) > \epsilon] && (17)\\
&= \prod_{i=1}^{m} \Pr[h_s(x_i) = c(x_i) \mid err(h_s) > \epsilon] && (18)\\
&\leq (1 - \epsilon)^m && (19)\\
&\leq e^{-\epsilon m} && (20)
\end{aligned}
$$

$$\leq \quad \delta, \text{ if } m \geq \frac{1}{\epsilon} \ln \frac{1}{\delta} \tag{21}$$

The argument goes as follows: Eq(13) uses definition of conditional probability; Eq(14) is because the denominator in Eq(13) is 1, since h_s is assumed to be consistent; Eq(16) is simply because $\Pr[err(h_s) > \epsilon] \leq 1$; Eq(18) is because the examples x_i are independent of one another; Eq(19) uses the fact that under the condition $err(h_s) > \epsilon$, the probability of $h_s(x_i) = c(x_i)$ shall be less than $1 - \epsilon$.

This argument appears plausible at first sight, but is wrong. The problem is that the hypothesis h_s in the proof is generated from the algorithm using sample S. Since S is random, h_s is also random. If we look back to the proof of the theorem, h is selected from the ϵ-bad set $\mathcal{B}$ (which is not random). In fact, h_s should be consistent with all the m examples (because it is generated by algorithm A), so $\Pr[h_s(x_i) = c(x_i)] = 1$ for all i. Also, the independence assumption is no longer valid. Therefore the bound is incorrect.

3 Consistency via PAC

Now we ask the following question: given a PAC algorithm A which learns $\mathcal{C}$ by $\mathcal{C}$, and m examples, can A find a rule $c \in \mathbb{C}$ consistent with all the examples?

The answer is yes, with probability $(1 - \delta)$. Here is how we obtain c. First generate D from uniform probability on the m examples. Choose $\epsilon < \frac{1}{m}$. Draw random examples from D, then the algorithm A would output hypothesis h, such that $err_D(h) \leq \epsilon < \frac{1}{m}$ with probability higher than $(1 - \delta)$, since each example has weight $\frac{1}{m}$ under D , h cannot make even a single error on the given examples. So h is consistent with all m examples.

4 Behaviors of training samples

Generally speaking, $|\mathcal{H}|$ is usually very big. For a sample set $S = \langle x_1, x_2, \ldots, x_m \rangle$, define the behavior set $\Pi_{\mathcal{H}}(S) = \{\langle h(x_1), \ldots, h(x_m) \rangle : h \in \mathcal{H}\}$ consisting of all possible behavior combinations of the m examples.

Define $\Pi_{\mathcal{H}}(m) = \max_{S:|S|=m} |\Pi_{\mathcal{H}}(S)|$. Obviously $\Pi_{\mathcal{H}}(m) \leq 2^m$.

4.1 Example1: positive half lines

Possible behaviors of positive half lines on four training examples are:

$$\begin{array}{cccc} - & - & - & - \\ - & - & - & + \\ - & - & + & + \\ - & + & + & + \\ + & + & + & + \end{array}$$

So there are five possible behaviors. If there are m training examples, then $m + 1$ possible behaviors/dichotomies exist, i.e., $\Pi_{\mathcal{H}}(m) = m + 1$.

4.2 Example2: half lines (either positive or negative)

Counting twice as much as the positive case, then subtracting the all –'s and all +'s which are counted twice, renders $\Pi_{\mathcal{H}}(m) = 2(m+1) - 2 = 2m$.

4.3 Example3: intervals

Select a range on the real axis, classify points within the range as positive, and outside the range as negative. If there are m points, there will be $m+1$ regions to locate the two borders of the range. We can either choose two different regions $\binom{m+1}{2}$, or the same region (1). So $\Pi_{\mathcal{H}}(m) = \frac{m(m+1)}{2} + 1$.

In these three examples, the numbers of all possible behaviors are much smaller than 2^m. In the first two cases they are linear in m, and the last case quadratic in m. This motivates us to try to substitute $\ln \Pi_{\mathcal{H}}(m)$ for $\ln |\mathcal{H}|$ in the theorem.

Also, for any $\mathcal{H}$, we can show that $\Pi_{\mathcal{H}}(m)$ is either 2^m or $O(m^d)$ polynomial, where d is called the VC-dimension. Then $\ln \Pi_{\mathcal{H}}(m) = d \ln m$, which will be determined by the VC-dimension. This is a strong argument, and we will show it next time in class.