COS 511: Theoretical Machine Learning

Lecturer: Rob Schapire
Scribe: Po-Hsuan (Cameron) Chen

Lecture # 4
February 14, 2013

1 Proof of learning bounds

1.1 Theorem

Theorem 1. *Suppose 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 $Pr[err_D(h) \geq \epsilon] \leq \delta$.*

We have made two assumptions for this theorem to apply. We assume (1) finite hypotheses space (2) both training and testing examples are independent and identically distributed (i.i.d.) according to some distribution D. This theorem provides an upper bound on the amount of training data m needed to achieve a certain error rate given ϵ, δ, and size of the hypotheses space $|\mathcal{H}|$.

1.2 Proof

We define $\mathcal{B} = \{h \in \mathcal{H} : h\ \epsilon\text{-bad}\}$ as the set of all $h \in \mathcal{H}$ that is ϵ-bad. ϵ-bad is defined as $err_D(h) > \epsilon$, which means that the hypothesis h has a performance worse than ϵ.

Proof. Assume h_A is consistent. We aim at providing an upper-bound to $Pr[err_D(h) \geq \epsilon]$, which is the probability that the hypothesis generated by A has a performance worse than ϵ.

$$
\begin{aligned}
& Pr[h_A\ \epsilon\text{-bad}] && (1.1)\\
&= Pr[h_A\ consistent \wedge h_A\ \epsilon\text{-bad}] && (1.2)\\
&\leq Pr[\exists h \in \mathcal{H} : h\ consistent \wedge \epsilon\text{-bad}] && (1.3)\\
&= Pr[\exists h \in \mathcal{B} : h\ consistent] && (1.4)\\
&= Pr[\bigvee_{h\in\mathcal{B}} h\ consistent] && (1.5)\\
&\leq \sum_{h\in\mathcal{B}} Pr[h\ consistent] && (1.6)\\
&= \sum_{h\in\mathcal{B}} Pr[(h(x_1) = c(x_1) \wedge \cdots \wedge h(x_m) = c(x_m))] && (1.7)
\end{aligned}
$$

$$= \sum_{h \in \mathcal{B}} \prod_{i=1}^{m} Pr[h(x_i) = c(x_i)] \tag{1.8}$$

$$\leq \sum_{h \in \mathcal{B}} (1-\epsilon)^m \tag{1.9}$$

$$= |\mathcal{B}|(1-\epsilon)^m \tag{1.10}$$

$$\leq |\mathcal{H}|(1-\epsilon)^m \tag{1.11}$$

$$\leq |\mathcal{H}|e^{-\epsilon m} \tag{1.12}$$

$$\leq \delta \tag{1.13}$$

□

Since we assume h_A is consistent, Equation(1.2) is directly implied by Equation(1.1). Equation (1.3) use the fact that if A, B are two events and A implies B, then $Pr[A] \leq Pr[B]$. We apply the definition of $\mathcal{B}$ to get Equation(1.4). Equation (1.6) uses union bound: $Pr[A \vee B] \leq Pr[A] + Pr[B]$. Equation (1.7) is just the definition of consistency. By applying the independence assumption that the m examples are sampled from distribution D, we get Equation (1.8). For Equation (1.9), since $h \in \mathcal{B}$, therefore h is ϵ-bad, which means that with probability $(1-\epsilon)$ that h is consistent to data x_i. Equation (1.11) is straightforward, since $\mathcal{B} \subseteq \mathcal{H}$. Equation(1.12) is based on the inequality $1 + x \leq e^x$.

1.3 Intuition

This theorem shows that when hypothesis space $\mathcal{H}$ is finite, a consistent algorithm A is a PAC-learning algorithm. This theorem provides an upper bound on how much data we need to achieve a certain general error rate. The bound captures a general relation between learning performance and the size of the hypothesis space, and the number of training examples. From the result we can see that the more data we have, the lower the upper bound of error we can achieve. Furthermore, the smaller the hypothesis size $|\mathcal{H}|$ is (by knowing more about the concept space), the less data we need to achieve a certain general error rate.

1.4 A seemingly plausible argument

We claim that the bound provided in Theorem 1 is not tight enough, since we are trying to bound it based on the size of the whole hypothesis space. Now let's try deriving an upper bound without using $|\mathcal{H}|$. Say h_A is the hypothesis output by the algorithm A given sample $\mathcal{S}$, which contains m examples.

$$Pr[h_A\,\epsilon\text{-bad}|h_A\,consistent] \tag{1.14}$$
$$= \frac{Pr[h_A\,consistent|h_A\,\epsilon\text{-bad}]Pr[h_A\,\epsilon\text{-bad}]}{Pr[h_A\,consistent]} \tag{1.15}$$
$$\leq Pr[h_A\,consistent|h_A\,\epsilon\text{-bad}] \tag{1.16}$$
$$= Pr[h_A(x_1) = c(x_1) \wedge \cdots \wedge h_A(x_m) = c(x_m)|h_A\,\epsilon\text{-bad}] \tag{1.17}$$
$$= \prod_{i=1}^{m} Pr[h_A(x_i) = c(x_i)|h_A\,\epsilon\text{-bad}] \tag{1.18}$$
$$\leq (1-\epsilon)^m \tag{1.19}$$
$$\leq e^{-\epsilon m} \tag{1.20}$$
$$\leq \delta \tag{1.21}$$

Therefore we get a tighter bound on m, such that as long as $m \geq \frac{1}{\epsilon} ln \frac{1}{\delta}$ we can get an upper bound δ on $Pr[h_A\,\epsilon\text{-bad}|h_A\,consistent]$.

We derive Equation (1.15) based on bayes rule. Since we know $Pr[h_A\,\epsilon - bad] \leq 1$ and $Pr[h_A\,consistent] = 1$, we get Equation (1.16). Equation (1.17) is derived by applying the definition of consistency. By applying the independence assumption that the m examples are sampled from distribution D, we get Equation (1.18). For Equation (1.19), given h_A is ϵ-bad, we know $err(h_A) > \epsilon$, and the probability of $h_A(x_i) = c(x_i)$ shall be less than $1 - \epsilon$.

The argument seems plausible, but it is actually incorrect. The problem is that the hypothesis h_A is generated from the sample $\mathcal{S}$, therefore it's actually a random variable that depends on the sample $\mathcal{S}$. Since h_A depends on the sample $\mathcal{S}$, given h_A is ϵ-bad, those samples are no longer i.i.d. Equation (1.18) is incorrect. Besides, $Pr[h_A(x_i) = c(x_i)|h_A\,\epsilon\text{-bad}]$ should be 1 for all i, because $h_A\,\epsilon$-bad already implies consistency of h_A for all samples m.

2 Consistency via PAC

Previously we were discussing the derivation of PAC learning given consistency on training samples. However, given a PAC algorithm A, can A find a target concept c such that it's consistent with all the examples?

Proposition. *Say $\mathcal{C}$ is PAC-learnable by $\mathcal{C}$ using algorithm A. Then A can be used as a subroutine to find, given n examples $\mathcal{S}$, a concept c in $\mathcal{C}$ that is consistent with all n examples (if one exists).*

Proof. Given n examples, we construct a distribution D that is uniform over the n examples in $\mathcal{S}$. We choose $\epsilon < \frac{1}{n}$, and any desired value of $\delta > 0$. We run algorithm A by sampling m examples from distribution D (where m is the number of examples required by A to attain the desired accuracy ϵ and confidence δ) to get hypothesis $h \in \mathcal{C}$, such that $err_D(h) \leq \epsilon < \frac{1}{n}$ with probability $\geq (1 - \delta)$. Given that D is uniform over $\mathcal{S}$, if h makes any error on any example, the probability of $err_D(h)$ will be at least $\frac{1}{n}$, which contradicts the ϵ we picked. Therefore, h is consistent with all n examples. □

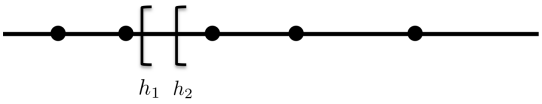
3 Learnability of Infinite Hypothesis Spaces

The result shown above only holds for finite hypothesis spaces. There are still various methods, such as for half-lines, rectangles, etc. that allow us to learn even though they have infinite hypothesis spaces. We tried to discuss what is the characteristic to make some $\mathcal{C}$ PAC-learnable. Before looking at some example, we first define growth function.

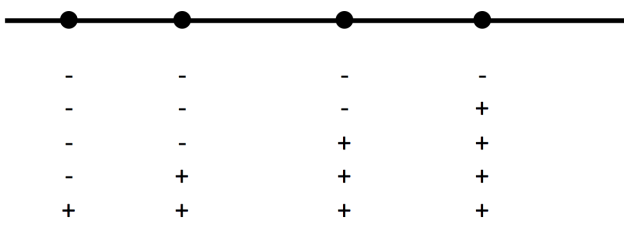
Definition. Growth function

For an unlabeled sample set $\mathcal{S} = \langle x_1, x_2, \ldots, x_m \rangle$, define a behavior set $\Pi_{\mathcal{H}}(\mathcal{S}) = \{\langle h(x_1), \ldots, h(x_m)\rangle : h \in \mathcal{H}\}$ and a growth function $\Pi_{\mathcal{H}}(m) = max_{|\mathcal{S}|=m}|\Pi_{\mathcal{H}}(\mathcal{S})|$. Notice that $\Pi_{\mathcal{H}}(\mathcal{S})$ is a set while $\Pi_{\mathcal{H}}(m)$ is a number.

3.1 Example 1: positive half-lines

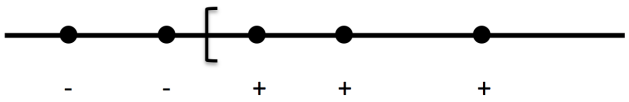


Given any unlabeled data set, h_1, h_2 behave exactly the same on the data set we provided. Although there are infinitely many hypotheses, with respect to a finite set of training points, there are only finitely many possible behaviors/labelings/dichotomies. For a set of four unlabeled data as shown in following figure, there are only 5 possible behaviors/labeling/dichotomies.



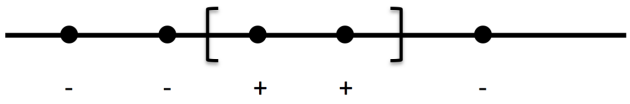
Generally speaking, given m points of data, there are only $m + 1$ possible behavior/dichotomies/labelings. $\prod_{\mathcal{H}}(m) = m + 1$. And these behavior/dichotomies/labelings kind of define the effective hypothesis spaces.

3.2 Example 2: half-lines



It's similar to Example 1, however, for half-line we can label data on the right as both + and −, while positive half-lines we can only label the data on the right as +. We double the result of example 1 (since we can interchange + and −) while subtracting the all −'s case and all +'s case which we counted twice. In the end we get $\Pi_{\mathcal{H}}(m) = 2(m + 1) - 2 = 2m$.

3.3 Example 3: intervals



Given 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 $m+1$ intervals to place the two borders of the range. Therefore the number of range will be $\binom{m+1}{2} + 1$, where 1 is the case that both borders are placed within the same interval, and it turns out identical no matter which interval you pick to place those borders. $\Pi_{\mathcal{H}}(m) = \binom{m+1}{2} + 1$.

3.4 Example 4: worst case

The worst case is that we need a hypothesis space with size equal to all possible functions on m points, which is 2^m.

Since the growth function is way smaller than the size of the original hypothesis space, it motivates us to replace $|\mathcal{H}|$ by the new space $\Pi_{\mathcal{H}}(m)$ in the theorem.

Also, for any $\mathcal{H}$, we can show that there are only two possible cases for $\Pi_{\mathcal{H}}(m)$. It's either $\Pi_{\mathcal{H}}(m) = 2^m$ (learning is hard) or $\Pi_{\mathcal{H}}(m) = O(m^d)$ (learning is possible), where d is VC-dimension of $\mathcal{H}$.

We state a theorem here, and it will be discussed in the next class.

Theorem. *With probability* $\geq 1 - \delta$. $\forall h \in \mathcal{H}$, *if* h *is consistent, then*

$$err_D(h) \leq O(\frac{ln(\Pi_{\mathcal{H}}(2m)) + ln\frac{1}{\delta}}{m})$$