The PAC Learning Model

These notes are slightly edited from scribe notes in previous years.

1 Scenario 2

The previous lecture analyzed a situation with a small enough finite set of concepts. The next example illustrates a case with an infinite set of hypotheses.

Examples An example is a real number.

Hypothesis A hypothesis is any interval \([a, b]\). We assume there is a true concept labeling the examples.

Situation Given 3000 IID examples, we want at most 1% error rate. Below is a visual representation of a subset of these examples, illustrating the domain:

\[
\begin{array}{ccccccccc}
  & & & & & & & & \\
\vdots & & & & & & & & \\
  & & & & & & & & \\
\end{array}
\]

The brackets mark the true concept, in relation to the sample data.

Picking a consistent hypothesis There is a wide range of hypotheses consistent with the sample data, ranging from the smallest interval including all +’s to the widest interval excluding all -’s. Because there is an infinite number of hypotheses, we cannot utilize the same proof from Game 1. In the following, consider \(\hat{h}\) = the tightest fit around positive examples.

What claims can we make?

- Suppose we place two points, \(a+\) and \(b-\), on the continuum such that:

\[
a+ = \arg\min_x \Pr[a, x] \geq \frac{0.01}{2} = \frac{1}{200}
\]

\[
b- = \arg\max_y \Pr[y, b] \geq \frac{0.01}{2} = \frac{1}{200}
\]

Note: both points are defined relative to distribution, not relative to the learner.

- What is the probability that an IID sample of 3000 examples does not hit \([a, a+]\) even once?

\[
(1 - \frac{1}{200})^{3000} \leq 2^{-21}
\]

The same holds for the interval \([b-, b]\).
Therefore, the probability that the sample has at least one example in \([a, a+]\) and at least one example in \([b-, b]\) is at least \(1 - (2^{-21} + 2^{-21}) = 1 - 2^{20}\).

- In other words, with probability \(\geq 1 - 2^{-20}\) the hypothesis has error smaller than 1% because the probability that the hypothesis is wrong is defined as the event that there is no example in that region:

\[
\text{err}(\text{hyp}) = Pr[a, x_{min}] + Pr[x_{max}, b] \leq Pr[a, a+] + Pr[b-, b] = \frac{1}{200} + \frac{1}{200} = \frac{1}{100}
\]

2 PAC Learning

We can now abstract the some of the details of the examples to capture the type of performance guarantees that was obtained.

**Definition:** Algorithm \(A\) PAC learns concept class \(C = \cup_n C_n\):

- if there exists a polynomial \(p(\ldots)\) such that
  - \(\forall n\)
  - \(\forall c \in C_n\)
  - \(\forall\) distributions \(D\) over \(X_n\),
  - \(\forall 0 < \epsilon < 1\) and \(\forall 0 < \delta < 1\)

If learning algorithm \(A\) is given \(\epsilon, \delta\) as input, and access to random training examples sampled from \(D\) and labeled by \(c\) (the true concept to be learned), then with probability at least \(1 - \delta\), \(A\) outputs a hypothesis \(h \in C\) such that:

\[
\text{Error}(h) = \text{Prob}_{D}[h(x) \neq c(x)] < \epsilon
\]

Moreover, \(A\)'s running time and sample size are bounded by \(p(n, \frac{1}{\epsilon}, \frac{1}{\delta}, \text{size}(c))\) (i.e. they are polynomial in those parameters.)

Note that we abstract the allowed error rate \(\epsilon\), and the failure probability \(\delta\). We also added two new parameters: a complexity parameter for the domain (typically number of features) and the size of the learned concept. All these can be used to quantify the complexity of the learning algorithm.

3 Learning a Finite Concept Class

We can now redo the analysis from scenario 1 in general terms. Let the sample size be \(m \geq \frac{1}{\epsilon} \ln \frac{|C|}{\delta}\).

Define: \(h\) is **good** when \(\text{Prob}[h(x) \text{ is correct}] \geq 1 - \epsilon\). Fixing any **bad** hypothesis \(h\) (one with error \(> \epsilon\)) we have

\[
\text{Prob}[h \text{ is correct on } m \text{ IID examples}] \leq (1- \epsilon)^m \leq (1- \epsilon)^{\frac{1}{\epsilon} \ln \frac{|C|}{\delta} } \leq e^{-\epsilon \frac{1}{\epsilon} \ln \frac{|C|}{\delta}} \leq \frac{\delta}{|C|}
\]

where we have used the fact that \(1 - x \leq e^{-x}\). Taking a union bound over all potential bad hypotheses shows that the total failure probability is at most \(|C| \frac{\delta}{|C|} = \delta\). Therefore, with probability at least \(1 - \delta\), the algorithm does not output a bad hypothesis (i.e. it outputs a good one). In other words the algorithm PAC learns \(C\).
The Tightest Fit Learning Algorithm on Interval [a,b]

The same abstraction can be done for scenario 2.

**Theorem:** The tightest fit learning algorithm PAC learns intervals when using \( m \geq \frac{2}{\epsilon} \ln \frac{2}{\delta} \) examples.

**Proof:** The true concept interval we are trying to learn is [a,b]. We define two points \( a^+ \) and \( b^- \) as follows:

\[
\begin{align*}
a^+ &= \text{argmin}_x (Pr[a, x] \geq \epsilon/2) \\
b^- &= \text{argmax}_y (Pr[y, b] \geq \epsilon/2)
\end{align*}
\]

*Note: both points are defined relative to distribution, not relative to the learner.*

Due to our definition of \( a^+ \) and \( b^- \), we have:

\[
Pr(\text{one example does not hit [a,a+]}) = 1 - \epsilon/2
\]

\[
Pr(\text{all m I.I.D. examples do not hit [a,a+]}) = (1 - \epsilon/2)^m
\]

\[
< e^{-m\epsilon/2} ; \quad \text{(using 1 - x < e^{-x})}
\]

\[
= \frac{\delta}{2}
\]

Also, \( Pr(\text{all m I.I.D. examples do not hit [b-,b]}) < \delta/2 \), by the same argument.

Then it follows that \( Pr(\text{sample hits both [a,a+] and [b-,b]}) \geq 1 - \delta \).

In this case, \( \text{err(hyp)} = Pr_D[h(x) \neq c(x)] < \epsilon \) is true since:

\[
\text{err(hyp)} = Pr_D[a, x^+_{min}] + Pr_D[x^+_{max}, b] ; \quad (x^+_{min} \text{ is the smallest } \oplus \text{ example, } x^+_{max} \text{ is the largest})
\]

\[
\leq Pr[a, a^+] + Pr[b^-, b] \\
= \epsilon/2 + \epsilon/2 \\
= \epsilon
\]

*NB:* The magnitude of \( \delta \) is usually exponentially small, whereas the magnitude of \( \epsilon \) is usually small only in a polynomial sense.