1 Where we were last time

With probability $\geq 1 - \delta$, $\forall h \in \mathcal{H}$, if $h$ is consistent with a sample of size $m$ then

$$err(h) \leq \frac{2}{m}(\lg \Pi_{\mathcal{H}}(2m) + \lg \frac{1}{\delta}).$$

We also showed that $\Pi_{\mathcal{H}}(m) \leq \Phi_d(m)$ where $d = \text{VCdim}(\mathcal{H})$.

2 Finding the order of magnitude on $err(h)$

We will show that $\Phi_d(m) = \sum_{i=0}^{d} \binom{m}{i} \leq \left(\frac{em}{d}\right)^d$ for $m \geq d$. We have

$$\left(\frac{d}{m}\right)^d \sum_{i=0}^{d} \binom{m}{i} \leq \sum_{i=0}^{d} \left(\frac{d}{m}\right)^i \binom{m}{i},$$

since $0 < \frac{d}{m} \leq 1$

$$\leq \sum_{i=0}^{m} \left(\frac{d}{m}\right)^i \binom{m}{i} 1^{(m-i)},$$

since we’re adding $m - d$ positive terms, and $1^{(m-i)}$ doesn’t change anything. But this is the binomial function, so

$$= \left(1 + \frac{d}{m}\right)^m.$$ 

And from $(1 + x) \leq e^x$

$$\leq e^{\frac{d}{m}m} = e^d.$$ 

So returning to the original equation, if $h$ is consistent then

$$err(h) \leq O\left(\frac{d\ln\frac{m}{d} + \ln\frac{1}{\delta}}{m}\right).$$

Or equivalently, $err(h) \leq \epsilon$ for

$$m = O\left(\frac{d\ln(1/\epsilon) + \ln(1/\delta)}{\epsilon}\right).$$

3 How is $d$ useful?

The $\text{VCdim}$ of $\mathcal{H}$, $d$, gives us a bound on how many examples $m$ we need to achieve $\epsilon$ and $\delta$. But, $\mathcal{H}$ is arbitrarily chosen, so it would be meaningless to use it to provide a lower bound for $m$. However, a lower bound for $m$ using $\text{VCdim}(\mathcal{C})$ can be found.
4 Error for $m \leq \frac{d}{2}$

We will prove that...

\[ \forall \text{ algorithms } A \exists \text{ concept class } c \in C \text{ and a distribution } D \text{ such that if only } m \leq \frac{d}{2} \text{ examples are selected from } D \text{ then} \]

\[ Pr\left(err(h) > \frac{1}{8}\right) \geq \frac{1}{8} \]

That is, for $\epsilon < \frac{1}{8}$ and $\delta < \frac{1}{8}$, PAC learning is impossible with fewer than (or equal to) $\frac{d}{2}$ examples.

To do this, we will assume $c$ is chosen at random by an adversary.

Proof:

Assume $s_1 \cdots s_d$ are shattered.

If $d = VCdim(C)$, then there exists a set of such examples that are shattered.

Take $C'$, a subset of $C$ which contains one representative concept $c$ for each dichotomy of the shattered set such that $c$ produces that dichotomy.

$|C'| = 2^d$

The adversary chooses some random $c \in C'$, where all members of $C'$ are uniformly distributed. The distribution $D$ is uniform over the shattered set.

So far, we have outlined ”experiment 1,” which can be summarized as:

- $c$ chosen at random
- sample $S = \{x_1, \ldots, x_m\}$ chosen at random
- $h_A$ computed by $A$ using $S$ and labels on that set
- $x$, a test point, is randomly chosen, and we then test if $h_A(x) \neq c(x)$

But, we claim this experiment is equivalent to ”experiment 2,” as follows:

- $S$ chosen at random
- labels $c(x_i)$ chosen just for those $x_i \in S$
- $h_A$ computed by $A$ using $S$ and labels on that set
- $x$, a test point, is randomly chosen and labeled (unless already labeled)
- test if $h_A(x) \neq c(x)$

The label for $x$ might have already been chosen if $x \in S$, in which case the hypothesis (which we assume to be consistent) has zero probability of incorrectly labeling $x$. Otherwise, $h_A$ has a 50/50 chance of selecting the right label.

Furthermore, $x$ has at most a 50% chance of being in $S$ (since $m \leq d/2$). So, computing probability over $c, S, x$:

\[
Pr(h_A(x) \neq c(x)) = Pr(x \in S \text{ and } h_A(x) \neq c(x)) + Pr(x \notin S \text{ and } h_A(x) \neq c(x)) \\
\geq 0 + Pr(x \notin S)Pr(h_A(x) \neq c(x)|x \notin S) \\
\geq 0 + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}.
\]
So $\frac{1}{4} \leq E_{x}(Pr_{S,x}[h_{A}(x) \neq c(x)])$
therefore $\exists c \in C': Pr(h_{A}(x) \neq c(x)) \geq \frac{1}{4}$
so $E_{S}(Pr_{x}[h_{A}(x) \neq c(x)]) \geq \frac{1}{4}$

... $E_{S}(err(h_{A})) \geq \frac{1}{4}$
$\frac{1}{4} \leq E_{S}(err(h_{A})) \leq Pr(err(h_{A}) > \frac{1}{8}) + Pr(err(h_{A}) \leq \frac{1}{8}) \cdot \frac{1}{8}$

$\frac{1}{4} \leq Pr(err(h_{A}) > \frac{1}{8}) + \frac{1}{8}$, because $Pr(err(h_{A}) \leq \frac{1}{8})$ is at most 1.

\[ Pr(err(h_{A}) > \frac{1}{8}) \geq \frac{1}{8} \]

\section{Inconsistent Hypotheses}

What are the cases in which we would be unable to find a consistent hypothesis?

- The true concept is not in $\mathcal{H}$
- The true concept is computationally hard to find
- There is no functional relationship between examples and labels

What if labels are probabilistically related to examples?

For a distribution $D$ on $X$ which takes values 0 or 1,
Replace $c(x)$ by $y$, no longer a function of $x$.
$Pr_{D}[x,y] = Pr(x)Pr(y|x)$
Before, we assumed $Pr(y|x)$ was either 0 or 1.
And we redefine error as $err(h) = Pr_{(x,y)\sim D}[h(x) \neq y]$
The best $h$ is one for which $h(x)$ is the more probable of 0 or 1:
$h_{\text{opt}}(x) = \{1 \text{ if } E(y|x) \geq \frac{1}{2} ; 0 \text{ else} \}$
$h_{\text{opt}}(x)$ is "Bayes’ optimal decision rule" and $err(h_{\text{opt}})$ is "Bayes’ error”

Let’s find an $h$ that minimizes $err(h)$.
We need an $\mathcal{H}$ rich enough so that $h_{\text{opt}}$ can be approximated. This is a possible source of error.

Idea: Minimize the number of errors on $S = \{(x_{i}, y_{i})\}$, "empirical risk minimization”.
Empirical errors $\hat{err}(h) = \frac{1}{m}|\{i : h(x_{i}) \neq y_{i}\}|$. We need the empirical error to be close
to the true error for every $h \in \mathcal{H}$. This is called uniform convergence. If we can do this,
then minimizing $\hat{err}(h)$ also means approximately minimizing $err(h)$:

Suppose we can show that $\forall h \in \mathcal{H}$
$|err(h) - \hat{err}(h)| \leq \epsilon$
Then let $\hat{h}$ be the hypothesis that minimizes $\hat{err}(h)$.
$err(\hat{h}) \leq \hat{err}(\hat{h}) + \epsilon$, by rewriting the above
$\leq \hat{err}(h) + \epsilon$ for any $h$, including the best one
$\leq err(h) + 2\epsilon$ by substituting from the original equation

So the true error of $\hat{h}$, the most consistent hypothesis, is within $2\epsilon$ of the error of the
best $h$ in the entire class, provided we can prove uniform convergence.

To prove uniform convergence results, we will need a powerful tool, called Chernoff bounds.
6 Chernoff Bounds, Part 1

For some set of random variables $X_1 \cdots X_m$, independently identically distributed, where $X_i \in [0, 1]$, let

\[ p = \text{E}(X_i) \]

\[ \hat{p} = \frac{1}{m} \sum X_i \]

which we will prove converges on $p$ quickly.

In the setting above, $X_i = \{1 \text{ if } h(x_i) \neq y_i, 0 \text{ else} \}$, $p = err(h)$ and $\hat{p} = \hat{err}(h)$.

Hoeffding’s Inequality states that:

\[ \Pr(\hat{p} \geq p + \epsilon) \leq e^{-2\epsilon^2 m} \]

\[ \Pr(\hat{p} \leq p - \epsilon) \leq e^{-2\epsilon^2 m} \]

So \( |\hat{p} - p| \leq \sqrt{\frac{\ln \frac{2}{\delta}}{2m}} \) with prob. \( \geq 1 - \delta \)

We will prove a stronger form:

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

where $RE$ is the relative entropy function, described below

7 Relative Entropy

$RE = \text{Relative Entropy}$ also known as Kullback-Liebler (KL) divergence

$RE(\cdot || \cdot)$ measures the distance between two distributions

Let’s say we’re sending a message $x$ which is selected from a distribution defined by probability $P(x)$.

The best way to encode $x$ is to use $\lg \frac{1}{P(x)}$ bits for $x$.

The entropy of $P$ is the expected code length: $\sum P(x) \lg \frac{1}{P(x)}$

But let’s say we ”think” the distribution of $x$ is $Q$.

The cross entropy of $P$ and $Q = \sum P(x) \lg \frac{1}{Q(x)}$, which would be the average code length, and is always at least the entropy of $P$.

The difference between the cross entropy and the entropy is $\sum P(x) \log \frac{P(x)}{Q(x)}$

which we call $RE(P||Q)$

If $x$ can take on only the values 0 and 1 with probability $p$ and $1 - p$, respectively, from $P$, and $q$ and $1 - q$, respectively, from $Q$,

then we may use the shorthand $RE(p||q) = p \lg \frac{p}{q} + (1 - p) \lg \frac{1 - p}{1 - q}$.

Although we used base 2 logarithm above in the definition of relative entropy, from now, we will use natural logarithm.