Machine Learning *Alex Clemmer, u0458675*

Notes

Hypothesis Class

A Hypothesis Class $\mathcal{H}$ is a non-finite-dimensional index into the space of hypothesis functions we can learn from our data.

Our goal is usually to learn some $h \in \mathcal{H}$ which outputs a result that we "hypothesize" given some bit of data. This h is typically learned using some N i.i.d. training examples (which, more technically, we can express as $\mathcal{D} \in (\boldsymbol{x}, y)^N$). Each training example $(\boldsymbol{x}_i, y_i)$ is drawn from $P(\boldsymbol{x}_i, y_i)$, which in other words is $\mathcal{D} \sim P^N$.

0-1 Training ("Empirical") Error

One way of measuring the "error" of some hypothesis funciton $h \in \mathcal{H}$ is to just look at the number of wrong answers it produces over all possible input:

$$L_{\mathcal{D}}(h) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}(h(\boldsymbol{x}_n \neq y_n))$$

0-1 Expected Error aka "True Error" aka "Misclassification Probability"

Another way to measure error is to look at the probability of misclassification. We can compute this, intuitively by looking at the expected error of misclassification:

$$L_P(h) = \mathbb{E}_{(\boldsymbol{x},y)\sim P}[\mathbb{I}(h(\boldsymbol{x}_n) \neq y_n)]$$

Zero Training Error

Say we have some hypothesis function $h \in \mathcal{H}$ with zero training error and a true error $L_P(h) > \epsilon$. The probability of h having zero error on any training example is $\leq 1 - \epsilon$. If we take the term $L_{\mathcal{D}}(h) = 0 \cap L_P(h) > \epsilon$ to mean roughly "h is bad", then the probability of h having zero on any training set $\mathcal{D} \in (\boldsymbol{x}, y)^N$ is:

$$P_{\mathcal{D}\sim P^N}(L_{\mathcal{D}}(h) = 0 \cap L_P(h) > \epsilon) \leq (1 - \epsilon)^N$$

Given that the hypothesis class $\mathcal{H}$ has k such hypothesis $\{h_1 \dots h_k\}$, the probability that *at least one* of them has zero training error is

$$P_{\mathcal{D}\sim P^N}(\text{"}h_1 \text{ is bad"} \cup \dots \cup \text{"}h_k \text{ is bad"}) \leq k(1 - \epsilon)^N$$

Of course $k \leq |\mathcal{H}|$, so k can actually be replaced by $|\mathcal{H}|$:

$$P_{\mathcal{D}\sim P^N}(\exists h : \text{“}h \text{ is bad”}) \leq |\mathcal{H}(1-\epsilon)^N$$

Further, since $(1-\epsilon) < e^{-e}$, we can reduce even more:

$$P_{\mathcal{D}} \sim P^N(\exists h : \text{“}h \text{ is bad} \leq |\mathcal{H}|e^{-Ne}$$

What we gain from this is that *the probability of h being bad decreases exponentially as N increases.*

Non-zero Training Error

An *empirical mean* produces a vector $\bar{\boldsymbol{x}}$, each of whose elements corresponds to the mean of all corresponding elements from every random variable in a set $\{z_1 \dots z_N\}$. Given any N-length set of random variables, this means basically that:

$$\bar{\boldsymbol{z}} = \frac{1}{N}\sum_{n=1}^{N} z_n$$

We will say also that the *true mean* is μ_z. Then the *Chernoff Bound* is given by:

$$P(|\mu_z - \bar{\boldsymbol{z}}| \geq \epsilon) \leq e^{-2N\epsilon^2} \tag{1}$$

Generalized, for any single hypothesis $h \in \mathcal{H}$, we can view the training error $L_P(h)$ as the true mean and the expected error $L_{\mathcal{D}}(h)$ have:

$$P(L_P(h) - L_{\mathcal{D}}(h) \geq \epsilon) \leq e{-}2N\epsilon^2 \tag{2}$$

Infinite Hypothesis Classes

When $|\mathcal{H}|$ is finite, then

$$L_P(h) \leq L_{\mathcal{D}}(h) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{1}{\delta}}{2N}} \tag{3}$$

Clearly, though, this breaks down when $|\mathcal{H}|$ is infinite. Might replace this with some notion of the complexity of $|\mathcal{H}|$. It turns out that we can, using the *Vapnik-Chervonenkis dimension*, which is a measure on the complexity of a hypothesis class.

First, some terminology: a set of points is said to be *shattered* by some $\mathcal{H}$ if for all possible labellings of the points, $\exists h \in \mathcal{H}$ that can represent the corresponding labeling function. One example of this is the fact that 3 points can always be shattered by any linear separator—there is no choice of labels you can pick for three points such that a linear separator fails to split them. Unfortunately, for 4 points in two dimensions, this does not work.