The Bias-Complexity Trade-off

Having misleading training data can lead to unsuccessful learning. To overcome this, the hypothesis set was restricted to some hypothesis class $\mathcal{H}$. This set reflects the prior knowledge that we have on the problem – bias. Now, we wish to know if such prior knowledge is required for the learner in the more general case; can we find an universal learner not restricted by some hypothesis class to take on any learning task. Formally defined, we wish to see if there exists a learner that given algorithm A, training set of size m and any distribution $D : \mathcal{X} \times \mathcal{Y}$ can find a way to always find a mapping $h : \mathcal{X} \to \mathcal{Y}$ whose risk is small enough.

Spoiler alert. This chapter is about proving that such a learner does not exist. A learner fails if it outputs risk such that there exists another learner with a smaller risk. Therefore, one should have some prior knowledge when learning. This can come from some specific parametric distribution or that there exists a hypothesis class $\mathcal{H}$ such that $h \in \mathcal{H}$ has zero or small loss.

But, restricting oneself to a hypothesis class can also be problematic. The error of the ERM algorithm is decomposed into two components. The first component reflects the quality of prior knowledge, measured by the minimal risk hypothesis (*approximation error*, *bias*) and the second component is about the complexity of class $\mathcal{H}$ – *estimation error*.

5.1 The No-Free-Lunch Theorem

Here, we prove there is no universal learner.

Theorem 5.1. A is any learning algorithm for a supervised binary classification problem over a domain $\mathcal{X}$. m is the training size upper bounded by $|\mathcal{X}|/2$. Then there exists a distribution D such that

1. There exists a function $f : \mathcal{X} \to 0, 1$ with $L_D(f) = 0$.
2. With probability of at least $\frac{1}{7}$ over the choice of $S \sim D^m$, the loss $L_D(A(S)) \geq 1/8$.

- **TODO**: should it be smaller or equal (see end of page 38 for $p \geq m$)

For every learner, there exists a task for which there is a better learner.

Proof Take C as the a subset of $\mathcal{X}$ of size $2m$. The idea of the proof is that a learning algorithm seeing half of the instances in C cannot generalize to the other half as good as some other "reality" function

There are $T = 2^{2m}$ possible mappings from $C \to 0, 1$. Label every one with $f_1, ..., f_T$. Now we define a distribution for each f_i such that $D_i(x, y) = 1/|C|$, if $y = f_i(x)$. This means the loss for function f_i on dataset D_i is always zero.

The idea is to show that the expectation of the loss from algorithm $A(S)$ on m examples from $C \times 0, 1$ is larger than that, more specifically

$$\max_{i \in |T|} \mathbb{E}[L_{D_i} A(S)] \geq 1/4$$

This condition then implies that the probablity of having loss larger than $1/8$ is at least $1/7$. (solve in exercise)

There are $k = (2m)^m$ m size sequences in C, denoting them as $S_1, ..., S_k$. An instance consists of samples $S_j = (x_1, ..., x_m)$, which can be labeled by any function f_i pairs $S_j^i = ((x_1, f_i(x_1)), ..., (x_m, f_i(x_m)))$. Now, we calculate the expected loss of algorithm getting dataset S for distribution D_i:

$$\mathbb{E}_{S \sim D_i^m}[L_{D_i}(A(S))] = \frac{1}{k} \sum_{j=1}^{k} L_{D_i}(A(S_j^i))$$

Searching through all possible distributions D_i and using the fact that maximum is always larger or equal to maximum, we fix the dataset for some $j \in [k]$. Let $v_1, ..., v_p$ be examples not in S_j ($p \geq m$). Now we want to find examples where labeling function is wrong.

$$L_{D_i}(h) = \frac{1}{2m} \sum_{x \in C} 1_{[h(x) \neq f_i(x)]}$$

$$L_{D_i}(h) \geq \frac{1}{2p} \sum_{r=1}^{p} 1_{[h(v_r) \neq f_i(v_r)]}$$

Also iterating through all possible labelings of $C \times 0, 1$ and taking the min:

$$\frac{1}{T} \sum_{i=1}^{T} L_{D_i}(A(S_j^i)) \geq \frac{1}{T} \sum_{i=1}^{T} \frac{1}{2p} \sum_{r=1}^{p} 1_{[A(S_j^i)(v_r) \neq f_i(v_r)]}$$

$$\frac{1}{T} \sum_{i=1}^{T} L_{D_i}(A(S_j^i)) \geq \frac{1}{2} \min_{r \in [p]} \frac{1}{T} \sum_{i=1}^{T} 1_{[A(S_j^i)(v_r) \neq f_i(v_r)]}$$

Next, we fix some example v_r, and partition all functions f_i into two disjoint $T/2$ sets $(f_i, f_i^{'})$ such that $f_i(v_r) \neq f_i^{'}(v_r)$, which makes the total of equal 1, implying that half of that is $\frac{1}{2}$.

$$\frac{1}{T} \sum_{i=1}^{T} L_{D_i}(A(S_j^i)) \geq \frac{1}{4}$$

$$\mathbb{E}_{S \sim D_i^m}[L_{D_i}(A(S))] \geq \frac{1}{4}$$

5.1.1 No-Free-Lunch and Prior Knowledge

If we take the hypothesis set $\mathcal{H}$ to be infinite, the previously proved no-free-lunch theorem says the learner will fail on some task, formally said

Corollary 5.2. Let $\mathcal{X}$ be an infinite domain set and let $\mathcal{H}$ be set of all possible functions from $\mathbb{X}$ to $0, 1$. $\mathcal{H}$ is not PAC learnable.

The proof is to apply the previously proved no-free-lunch theorem. Since we need to choose some hypothesis class, the idea is to find a class that involves the optimal labeling functions, but one can't have too big of a hypothesis class set (all functions over a domain) due to the no-free-lunch theorem. Therefore, a tradeoff is necessary.

5.2 Error decomposition

The error of an $ERM_{\mathcal{H}}$ predictor is decomposed in two:

$$L_D(h_S) = \epsilon_{app} + \epsilon_{est}$$

$$\epsilon_{app} = \min_{h \in \mathcal{H}} L_D(h)$$

$$\epsilon_{est} = L_D(h_S) - \epsilon_{app}$$

The approximation error reflects the inductive bias, the minimal risk achievable by a predictor by restricting ourselves to a specific hypothesis class. Enlarging the hypothesis class tends to decrease this error.

The estimation error – the difference between the approximation error and the error achieved by the $ERM_{\mathcal{H}}$ predictor.