5

The Bias-Complexity Trade-off

In Chapter 2 we saw that unless one is careful, the training data can mislead the learner, and result in overfitting. To overcome this problem, we restricted the search space to some hypothesis class $\mathcal{H}$. Such a hypothesis class can be viewed as reflecting some prior knowledge that the learner has about the task – a belief that one of the members of the class $\mathcal{H}$ is a low-error model for the task. For example, in our papayas taste problem, on the basis of our previous experience with other fruits, we may assume that some rectangle in the color-hardness plane predicts (at least approximately) the papaya's tastiness.

Is such prior knowledge really necessary for the success of learning? Maybe there exists some kind of universal learner, that is, a learner who has no prior knowledge about a certain task and is ready to be challenged by any task? Let us elaborate on this point. A specific learning task is defined by an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, where the goal of the learner is to find a predictor $h : \mathcal{X} \to \mathcal{Y}$, whose risk, $L_{\mathcal{D}}(h)$, is small enough. The question is therefore whether there exist a learning algorithm A and a training set size m, such that for every distribution $\mathcal{D}$, if A receives m i.i.d. examples from $\mathcal{D}$, there is a high chance it outputs a predictor h that has a low risk.

The first part of this chapter addresses this question formally. The No-Free-Lunch theorem states that no such universal learner exists. To be more precise, the theorem states that for binary classification prediction tasks, for every learner there exists a distribution on which it fails. We say that the learner fails if, upon receiving i.i.d. examples from that distribution, its output hypothesis is likely to have a large risk, say, ≥ 0.3, whereas for the same distribution, there exists another learner that will output a hypothesis with a small risk. In other words, the theorem states that no learner can succeed on all learnable tasks – every learner has tasks on which it fails while other learners succeed.

Therefore, when approaching a particular learning problem, defined by some distribution $\mathcal{D}$, we should have some prior knowledge on $\mathcal{D}$. One type of such prior knowledge is that $\mathcal{D}$ comes from some specific parametric family of distributions. We will study learning under such assumptions later on in Chapter 24. Another type of prior knowledge on $\mathcal{D}$, which we assumed when defining the PAC learning model,

https://doi.org/10.1017/CBO9781107298019.006 Published online by Cambridge University Press

is that there exists h in some predefined hypothesis class $\mathcal{H}$, such that $L_{\mathcal{D}}(h) = 0$. A softer type of prior knowledge on $\mathcal{D}$ is assuming that $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$ is small. In a sense, this weaker assumption on $\mathcal{D}$ is a prerequisite for using the agnostic PAC model, in which we require that the risk of the output hypothesis will not be much larger than $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$.

In the second part of this chapter we study the benefits and pitfalls of using a hypothesis class as a means of formalizing prior knowledge. We decompose the error of an ERM algorithm over a class $\mathcal{H}$ into two components. The first component reflects the quality of our prior knowledge, measured by the minimal risk of a hypothesis in our hypothesis class, $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$. This component is also called the *approximation error*, or the *bias* of the algorithm toward choosing a hypothesis from $\mathcal{H}$. The second component is the error due to overfitting, which depends on the size or the complexity of the class $\mathcal{H}$ and is called the *estimation error*. These two terms imply a tradeoff between choosing a more complex $\mathcal{H}$ (which can decrease the bias but increases the risk of overfitting) or a less complex $\mathcal{H}$ (which might increase the bias but decreases the potential overfitting).

5.1 THE NO-FREE-LUNCH THEOREM

In this part we prove that there is no universal learner. We do this by showing that no learner can succeed on all learning tasks, as formalized in the following theorem:

Theorem 5.1. (No-Free-Lunch) *Let A be any learning algorithm for the task of binary classification with respect to the 0–1 loss over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$, representing a training set size. Then, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that:*

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

This theorem states that for every learner, there exists a task on which it fails, even though that task can be successfully learned by another learner. Indeed, a trivial successful learner in this case would be an ERM learner with the hypothesis class $\mathcal{H} = \{f\}$, or more generally, ERM with respect to any finite hypothesis class that contains f and whose size satisfies the equation $m \geq 8\log(7|\mathcal{H}|/6)$ (see Corollary 2.3).

Proof. Let C be a subset of $\mathcal{X}$ of size $2m$. The intuition of the proof is that any learning algorithm that observes only half of the instances in C has no information on what should be the labels of the rest of the instances in C. Therefore, there exists a "reality," that is, some target function f, that would contradict the labels that $A(S)$ predicts on the unobserved instances in C.

Note that there are $T = 2^{2m}$ possible functions from C to $\{0,1\}$. Denote these functions by $f_1, \ldots, f_T$. For each such function, let $\mathcal{D}_i$ be a distribution over $C \times \{0,1\}$ defined by

$$\mathcal{D}_i(\{(x,y)\}) = \begin{cases} 1/|C| & \text{if } y = f_i(x) \\ 0 & \text{otherwise.} \end{cases}$$

https://doi.org/10.1017/CBO9781107298019.006 Published online by Cambridge University Press

That is, the probability to choose a pair (x, y) is $1/|C|$ if the label y is indeed the true label according to f_i, and the probability is 0 if $y \neq f_i(x)$. Clearly, $L_{\mathcal{D}_i}(f_i) = 0$.

We will show that for every algorithm, A, that receives a training set of m examples from $C \times \{0,1\}$ and returns a function $A(S) : C \to \{0,1\}$, it holds that

$$\max_{i \in [T]} \mathop{\mathbb{E}}_{S \sim \mathcal{D}_i^m} [L_{\mathcal{D}_i}(A(S))] \geq 1/4. \tag{5.1}$$

Clearly, this means that for every algorithm, A', that receives a training set of m examples from $\mathcal{X} \times \{0,1\}$ there exist a function $f : \mathcal{X} \to \{0,1\}$ and a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, such that $L_{\mathcal{D}}(f) = 0$ and

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m} [L_{\mathcal{D}}(A'(S))] \geq 1/4. \tag{5.2}$$

It is easy to verify that the preceding suffices for showing that $\mathbb{P}[L_{\mathcal{D}}(A'(S)) \geq 1/8] \geq 1/7$, which is what we need to prove (see Exercise 5.1).

We now turn to proving that Equation (5.1) holds. There are $k = (2m)^m$ possible sequences of m examples from C. Denote these sequences by $S_1, \ldots, S_k$. Also, if $S_j = (x_1, \ldots, x_m)$ we denote by S_j^i the sequence containing the instances in S_j labeled by the function f_i, namely, $S_j^i = ((x_1, f_i(x_1)), \ldots, (x_m, f_i(x_m)))$. If the distribution is $\mathcal{D}_i$ then the possible training sets A can receive are $S_1^i, \ldots, S_k^i$, and all these training sets have the same probability of being sampled. Therefore,

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}_i^m} [L_{\mathcal{D}_i}(A(S))] = \frac{1}{k} \sum_{j=1}^{k} L_{\mathcal{D}_i}(A(S_j^i)). \tag{5.3}$$

Using the facts that "maximum" is larger than "average" and that "average" is larger than "minimum," we have

$$\begin{aligned} \max_{i \in [T]} \frac{1}{k} \sum_{j=1}^{k} L_{\mathcal{D}_i}(A(S_j^i)) &\geq \frac{1}{T} \sum_{i=1}^{T} \frac{1}{k} \sum_{j=1}^{k} L_{\mathcal{D}_i}(A(S_j^i)) \\ &= \frac{1}{k} \sum_{j=1}^{k} \frac{1}{T} \sum_{i=1}^{T} L_{\mathcal{D}_i}(A(S_j^i)) \\ &\geq \min_{j \in [k]} \frac{1}{T} \sum_{i=1}^{T} L_{\mathcal{D}_i}(A(S_j^i)). \end{aligned} \tag{5.4}$$

Next, fix some $j \in [k]$. Denote $S_j = (x_1, \ldots, x_m)$ and let $v_1, \ldots, v_p$ be the examples in C that do not appear in S_j. Clearly, $p \geq m$. Therefore, for every function $h : C \to$

https://doi.org/10.1017/CBO9781107298019.006 Published online by Cambridge University Press

$\{0,1\}$ and every i we have

$$\begin{aligned} L_{\mathcal{D}_i}(h) &= \frac{1}{2m}\sum_{x\in C} \mathbb{1}_{[h(x)\neq f_i(x)]} \\ &\geq \frac{1}{2m}\sum_{r=1}^{p} \mathbb{1}_{[h(v_r)\neq f_i(v_r)]} \\ &\geq \frac{1}{2p}\sum_{r=1}^{p} \mathbb{1}_{[h(v_r)\neq f_i(v_r)]}. \end{aligned} \tag{5.5}$$

Hence,

$$\begin{aligned} \frac{1}{T}\sum_{i=1}^{T} L_{\mathcal{D}_i}(A(S_j^i)) &\geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{2p}\sum_{r=1}^{p} \mathbb{1}_{[A(S_j^i)(v_r)\neq f_i(v_r)]} \\ &= \frac{1}{2p}\sum_{r=1}^{p}\frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{[A(S_j^i)(v_r)\neq f_i(v_r)]} \\ &\geq \frac{1}{2}\cdot \min_{r\in[p]}\frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{[A(S_j^i)(v_r)\neq f_i(v_r)]}. \end{aligned} \tag{5.6}$$

Next, fix some $r \in [p]$. We can partition all the functions in $f_1, \ldots, f_T$ into $T/2$ disjoint pairs, where for a pair $(f_i, f_{i'})$ we have that for every $c \in C$, $f_i(c) \neq f_{i'}(c)$ if and only if $c = v_r$. Since for such a pair we must have $S_j^i = S_j^{i'}$, it follows that

$$\mathbb{1}_{[A(S_j^i)(v_r)\neq f_i(v_r)]} + \mathbb{1}_{[A(S_j^{i'})(v_r)\neq f_{i'}(v_r)]} = 1,$$

which yields

$$\frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{[A(S_j^i)(v_r)\neq f_i(v_r)]} = \frac{1}{2}.$$

Combining this with Equation (5.6), Equation (5.4), and Equation (5.3), we obtain that Equation (5.1) holds, which concludes our proof. □

5.1.1 No-Free-Lunch and Prior Knowledge

How does the No-Free-Lunch result relate to the need for prior knowledge? Let us consider an ERM predictor over the hypothesis class $\mathcal{H}$ of all the functions f from X to $\{0,1\}$. This class represents lack of prior knowledge: Every possible function from the domain to the label set is considered a good candidate. According to the No-Free-Lunch theorem, any algorithm that chooses its output from hypotheses in $\mathcal{H}$, and in particular the ERM predictor, will fail on some learning task. Therefore, this class is not PAC learnable, as formalized in the following corollary:

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

Proof. Assume, by way of contradiction, that the class is learnable. Choose some $\epsilon < 1/8$ and $\delta < 1/7$. By the definition of PAC learnability, there must be some

https://doi.org/10.1017/CBO9781107298019.006 Published online by Cambridge University Press

learning algorithm A and an integer $m = m(\epsilon, \delta)$, such that for any data-generating distribution over $\mathcal{X} \times \{0, 1\}$, if for some function $f : \mathcal{X} \to \{0, 1\}$, $L_{\mathcal{D}}(f) = 0$, then with probability greater than $1 - \delta$ when A is applied to samples S of size m, generated i.i.d. by $\mathcal{D}$, $L_{\mathcal{D}}(A(S)) \leq \epsilon$. However, applying the No-Free-Lunch theorem, since $|\mathcal{X}| > 2m$, for every learning algorithm (and in particular for the algorithm A), there exists a distribution $\mathcal{D}$ such that with probability greater than $1/7 > \delta$, $L_{\mathcal{D}}(A(S)) > 1/8 > \epsilon$, which leads to the desired contradiction. □

How can we prevent such failures? We can escape the hazards foreseen by the No-Free-Lunch theorem by using our prior knowledge about a specific learning task, to avoid the distributions that will cause us to fail when learning that task. Such prior knowledge can be expressed by restricting our hypothesis class.

But how should we choose a good hypothesis class? On the one hand, we want to believe that this class includes the hypothesis that has no error at all (in the PAC setting), or at least that the smallest error achievable by a hypothesis from this class is indeed rather small (in the agnostic setting). On the other hand, we have just seen that we cannot simply choose the richest class – the class of all functions over the given domain. This tradeoff is discussed in the following section.

5.2 ERROR DECOMPOSITION

To answer this question we decompose the error of an $\text{ERM}_{\mathcal{H}}$ predictor into two components as follows. Let h_S be an $\text{ERM}_{\mathcal{H}}$ hypothesis. Then, we can write

$$L_{\mathcal{D}}(h_S) = \epsilon_{\text{app}} + \epsilon_{\text{est}} \quad \text{where}: \ \epsilon_{\text{app}} = \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h), \quad \epsilon_{\text{est}} = L_{\mathcal{D}}(h_S) - \epsilon_{\text{app}}. \tag{5.7}$$

- **The Approximation Error** – the minimum risk achievable by a predictor in the hypothesis class. This term measures how much risk we have because we restrict ourselves to a specific class, namely, how much *inductive bias* we have. The approximation error does not depend on the sample size and is determined by the hypothesis class chosen. Enlarging the hypothesis class can decrease the approximation error.

  Under the realizability assumption, the approximation error is zero. In the agnostic case, however, the approximation error can be large.[1]
- **The Estimation Error** – the difference between the approximation error and the error achieved by the ERM predictor. The estimation error results because the empirical risk (i.e., training error) is only an estimate of the true risk, and so the predictor minimizing the empirical risk is only an estimate of the predictor minimizing the true risk.

  The quality of this estimation depends on the training set size and on the size, or complexity, of the hypothesis class. As we have shown, for a finite hypothesis class, ϵ_{est} increases (logarithmically) with $|\mathcal{H}|$ and decreases with m. We can

[1] In fact, it always includes the error of the Bayes optimal predictor (see Chapter 3), the minimal yet inevitable error, because of the possible nondeterminism of the world in this model. Sometimes in the literature the term *approximation error* refers not to $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$, but rather to the excess error over that of the Bayes optimal predictor, namely, $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - \epsilon_{\text{Bayes}}$.

https://doi.org/10.1017/CBO9781107298019.006 Published online by Cambridge University Press

think of the size of $\mathcal{H}$ as a measure of its complexity. In future chapters we will define other complexity measures of hypothesis classes.

Since our goal is to minimize the total risk, we face a tradeoff, called the *bias-complexity tradeoff*. On one hand, choosing $\mathcal{H}$ to be a very rich class decreases the approximation error but at the same time might increase the estimation error, as a rich $\mathcal{H}$ might lead to *overfitting*. On the other hand, choosing $\mathcal{H}$ to be a very small set reduces the estimation error but might increase the approximation error or, in other words, might lead to *underfitting*. Of course, a great choice for $\mathcal{H}$ is the class that contains only one classifier – the Bayes optimal classifier. But the Bayes optimal classifier depends on the underlying distribution $\mathcal{D}$, which we do not know (indeed, learning would have been unnecessary had we known $\mathcal{D}$).

Learning theory studies how rich we can make $\mathcal{H}$ while still maintaining reasonable estimation error. In many cases, empirical research focuses on designing good hypothesis classes for a certain domain. Here, "good" means classes for which the approximation error would not be excessively high. The idea is that although we are not experts and do not know how to construct the optimal classifier, we still have some prior knowledge of the specific problem at hand, which enables us to design hypothesis classes for which both the approximation error and the estimation error are not too large. Getting back to our papayas example, we do not know how exactly the color and hardness of a papaya predict its taste, but we do know that papaya is a fruit and on the basis of previous experience with other fruit we conjecture that a rectangle in the color-hardness space may be a good predictor.

5.3 SUMMARY

The No-Free-Lunch theorem states that there is no universal learner. Every learner has to be specified to some task, and use some prior knowledge about that task, in order to succeed. So far we have modeled our prior knowledge by restricting our output hypothesis to be a member of a chosen hypothesis class. When choosing this hypothesis class, we face a tradeoff, between a larger, or more complex, class that is more likely to have a small approximation error, and a more restricted class that would guarantee that the estimation error will be small. In the next chapter we will study in more detail the behavior of the estimation error. In Chapter 7 we will discuss alternative ways to express prior knowledge.

5.4 BIBLIOGRAPHIC REMARKS

(Wolpert & Macready 1997) proved several no-free-lunch theorems for optimization, but these are rather different from the theorem we prove here. The theorem we prove here is closely related to lower bounds in VC theory, as we will study in the next chapter.

5.5 EXERCISES

5.1 Prove that Equation (5.2) suffices for showing that $\mathbb{P}[L_{\mathcal{D}}(A(S)) \geq 1/8] \geq 1/7$.
Hint: Let θ be a random variable that receives values in $[0, 1]$ and whose expectation satisfies $\mathbb{E}[\theta] \geq 1/4$. Use Lemma B.1 to show that $\mathbb{P}[\theta \geq 1/8] \geq 1/7$.

https://doi.org/10.1017/CBO9781107298019.006 Published online by Cambridge University Press

5.2 Assume you are asked to design a learning algorithm to predict whether patients are going to suffer a heart attack. Relevant patient features the algorithm may have access to include blood pressure (BP), body-mass index (BMI), age (A), level of physical activity (P), and income (I).

You have to choose between two algorithms; the first picks an axis aligned rectangle in the two dimensional space spanned by the features BP and BMI and the other picks an axis aligned rectangle in the five dimensional space spanned by all the preceding features.

1. Explain the pros and cons of each choice.
2. Explain how the number of available labeled training samples will affect your choice.

5.3 Prove that if $|\mathcal{X}| \geq km$ for a positive integer $k \geq 2$, then we can replace the lower bound of $1/4$ in the No-Free-Lunch theorem with $\frac{k-1}{2k} = \frac{1}{2} - \frac{1}{2k}$. Namely, let A be a learning algorithm for the task of binary classification. Let m be any number smaller than $|\mathcal{X}|/k$, representing a training set size. Then, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0, 1\}$ such that:

- There exists a function $f : \mathcal{X} \to \{0, 1\}$ with $L_{\mathcal{D}}(f) = 0$.
- $\mathbb{E}_{S \sim \mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \geq \frac{1}{2} - \frac{1}{2k}$.

https://doi.org/10.1017/CBO9781107298019.006 Published online by Cambridge University Press