Bias Complexity TradeOff

Unit 6

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.

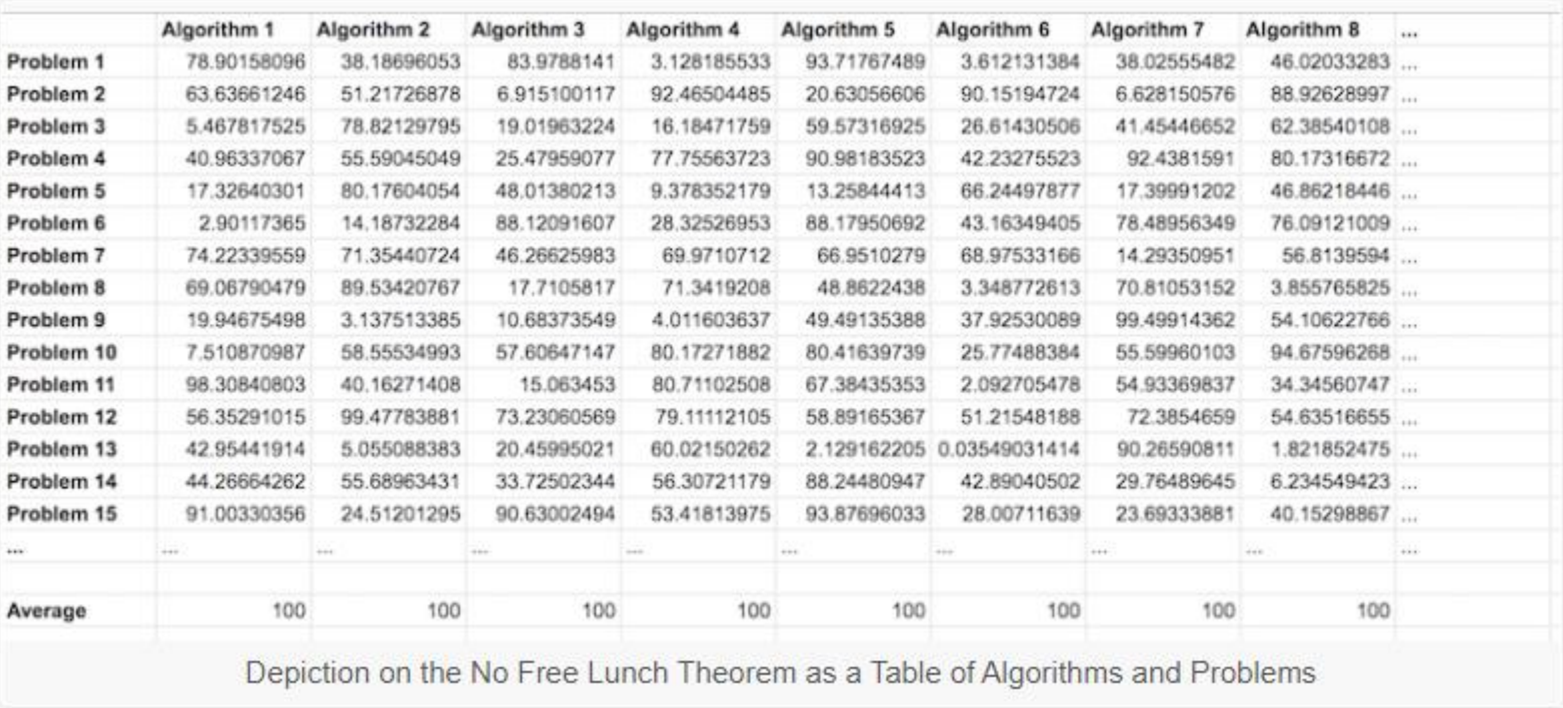
Is such prior knowledge really necessary for the success of learning? Maybe there exists some kind of universal learner, that is, a learner who has no prior knowledge about a certain task and is ready to be challenged by any task? Let us elaborate on this point. A specific learning task is defined by 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.

FREE LUNCH

No such thing.

No free Lunch Theorem

- The No Free Lunch Theorem, often abbreviated as NFL or NFLT, is a theoretical finding that suggests all optimization algorithms perform equally well when their performance is averaged over all possible objective functions.

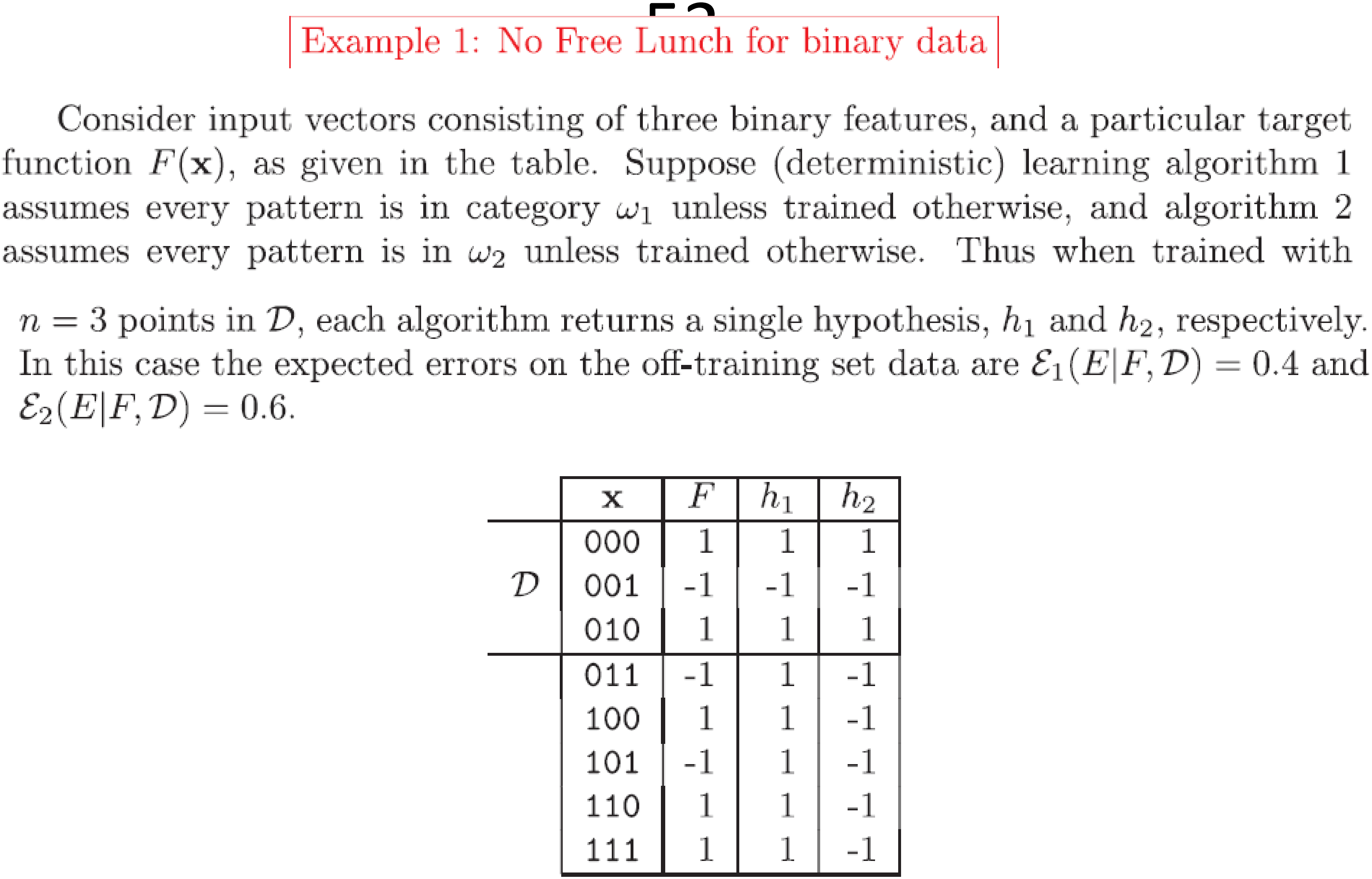


	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4	Algorithm 5	Algorithm 6	Algorithm 7	Algorithm 8	...
Problem 1	78.90158096	38.18696053	83.9788141	3.128185533	93.71767489	3.612131384	38.02555482	46.02033283	...
Problem 2	63.63661246	51.21726878	6.915100117	92.46504485	20.63056606	90.15194724	6.628150576	88.92628997	...
Problem 3	5.467817525	78.82129795	19.01963224	16.18471759	59.57316925	26.61430506	41.45446652	62.38540108	...
Problem 4	40.96337067	55.59045049	25.47959077	77.75563723	90.98183523	42.23275523	92.4381591	80.17316672	...
Problem 5	17.32640301	80.17604054	48.01380213	9.378352179	13.25844413	66.24497877	17.39991202	46.86218446	...
Problem 6	2.90117365	14.18732284	88.12091607	28.32526953	88.17950692	43.16349405	78.48956349	76.09121009	...
Problem 7	74.22339559	71.35440724	46.26625983	69.9710712	66.9510279	68.97533166	14.29350951	56.8139594	...
Problem 8	69.06790479	89.53420767	17.7105817	71.3419208	48.8622438	3.348772613	70.81053152	3.855765825	...
Problem 9	19.94675498	3.137513385	10.68373549	4.011603637	49.49135388	37.92530089	99.49914362	54.10622766	...
Problem 10	7.510870987	58.55534993	57.60647147	80.17271882	80.41639739	25.77488384	55.59960103	94.67596268	...
Problem 11	98.30840803	40.16271408	15.063453	80.71102508	67.38435353	2.092705478	54.93369837	34.34560747	...
Problem 12	56.35291015	99.47783881	73.23060569	79.11112105	58.89165367	51.21548188	72.3854659	54.63516655	...
Problem 13	42.95441914	5.055088383	20.45995021	60.02150262	2.129162205	0.03549031414	90.26590811	1.821852475	...
Problem 14	44.26664262	55.68963431	33.72502344	56.30721179	88.24480947	42.89040502	29.76489645	6.234549423	...
Problem 15	91.00330356	24.51201295	90.63002494	53.41813975	93.87696033	28.00711639	23.69333881	40.15298867	...
...	...	...	...	...	...	...	...	...	...
Average	100	100	100	100	100	100	100	100	

Depiction on the No Free Lunch Theorem as a Table of Algorithms and Problems

We now turn to the central question posed above: If we are interested solely in the generalization performance, are there any reasons to prefer one classifier or learning algorithm over another? If we make no prior assumptions about the nature of the classification task, can we expect any classification method to be superior or inferior overall? Can we even find an algorithm that is overall superior to (or inferior to) random guessing?

As summarized in the *No Free Lunch Theorem*, the answer to these and several related questions is *no*: on the criterion of generalization performance, there are no context- or problem-independent reasons to favor one learning or classification method over another. The apparent superiority of one algorithm or set of algorithms is due to the nature of the problems investigated and the distribution of data. It is an appreciation of the No Free Lunch Theorem that allows us, when confronting practical pattern recognition problems, to focus on the aspects that matter most — prior information, data distribution, amount of training data and cost or reward functions. The Theorem also justifies a scepticism about studies that purport to demonstrate the overall superiority of a particular learning or recognition algorithm.



Example 1: No Free Lunch for binary data

Consider input vectors consisting of three binary features, and a particular target function $F(\mathbf{x})$, as given in the table. Suppose (deterministic) learning algorithm 1 assumes every pattern is in category ω_1 unless trained otherwise, and algorithm 2 assumes every pattern is in ω_2 unless trained otherwise. Thus when trained with

$n = 3$ points in $\mathcal{D}$, each algorithm returns a single hypothesis, h_1 and h_2, respectively. In this case the expected errors on the off-training set data are $\mathcal{E}_1(E|F, \mathcal{D}) = 0.4$ and $\mathcal{E}_2(E|F, \mathcal{D}) = 0.6$.

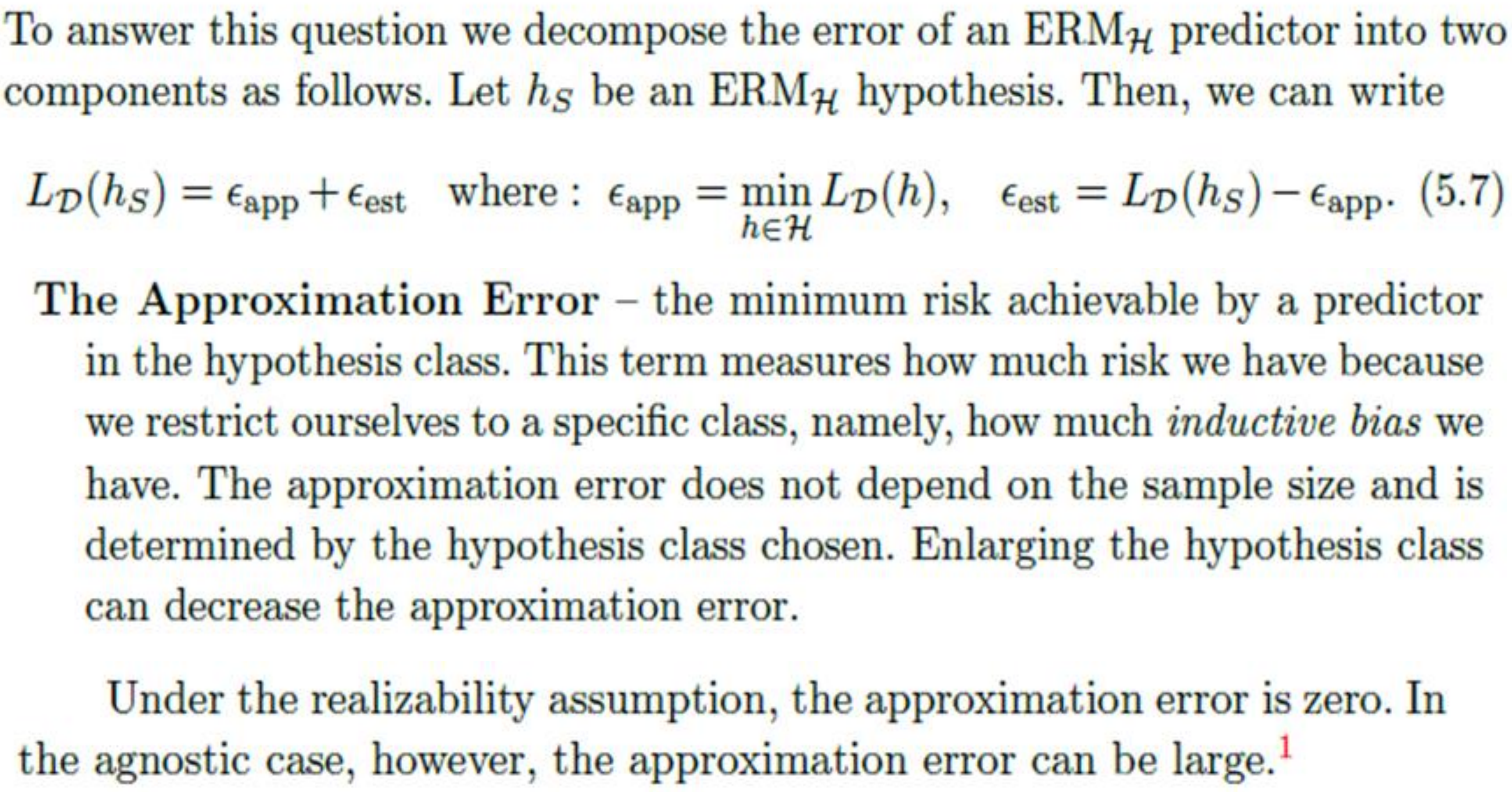
	$\mathbf{x}$	F	h_1	h_2
$\mathcal{D}$	000	1	1	1
	001	-1	-1	-1
	010	1	1	1
	011	-1	1	-1
	100	1	1	-1
	101	-1	1	-1
	110	1	1	-1
	111	1	1	-1

*Ugly Duckling Theorem

While the No Free Lunch Theorem shows that in the absence of assumptions we should not prefer any learning or classification algorithm over another, an analogous theorem addresses features and patterns. Roughly speaking, the Ugly Duckling Theorem states that in the absence of assumptions there is no privileged or "best" feature representation, and that even the notion of similarity between patterns depends implicitly on assumptions which may or may not be correct.

Theorem 9.2 (Ugly Duckling) *Given that we use a finite set of predicates that enables us to distinguish any two patterns under consideration, the number of predicates shared by any two such patterns is constant and independent of the choice of those patterns. Furthermore, if pattern similarity is based on the total number of predicates shared by two patterns, then any two patterns are "equally similar."* *

Error Decomposition



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

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

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]

Effect of hypothesis class size

As the hypothesis class size increases...

Approximation error decreases because:

taking min over larger set

Estimation error increases because:

harder to estimate something more complex

Estimation error analogy

Scenario 1: ask few people around

Is your name Joe?

Scenario 2: email all of Stanford

Is your name Joe?

people = hypotheses, questions = examples

- **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 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.

Thanks!!!