ELEG/CISC 867: Advanced Machine Learning **Spring 2019**

Lecture 4: Bias-Complexity Tradeoff

Lecturer: Xiugang Wu *03/05/2019*

In previous lectures, we saw that unless one is careful, using ERM on 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.

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 P over $\mathcal{X} \times \mathcal{Y}$, where the goal of the learner is to find a predictor h, whose risk, $L(h, P)$, is small enough. The question is therefore whether there exist a learning algorithm A and a training set size n, such that for every distribution P, if A receives n i.i.d. examples from P, there is a high chance it outputs a predictor h that has a low risk.

1 The No-Free-Lunch (NFL) Theorem

In the first part of this lecture, we show that no such universal learner exists by proving the NFL theorem. In particular, the NFL theorem states that for binary classification prediction tasks, for every learner there exists a distribution on which it fails (i.e. upon receiving i.i.d. examples from that distribution, its output hypothesis is likely to have a large risk), 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.

Theorem 1.1 (No-Free-Lunch) *Let A be any learning algorithm for the task of binary classification with respect to 0-1 loss over a domain $\mathcal{X}$. Let n be any number small than $|\mathcal{X}|/2$, representing the training set size. Then there exists a distribution P over $\mathcal{X} \times \{0, 1\}$ such that:*

1. *There exists a function f with $L(f, P) = 0$.*
2. *With probability at least $1/7$ we have that $L(A(Z^n), P) \geq 1/8$, i.e.,*

$$P^n(L(A(Z^n), P) \geq 1/8) \geq 1/7.$$

1.1 Proof Idea of NFL Theorem

Let C be a subset of $\mathcal{X}$ of size $2n$. 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 some target function f that would contradict the labels that $A(Z^n)$ predicts on the unobserved instances in C.

Note that there are $T = 2^{2n}$ possible functions from C to $\{0, 1\}$. Denote these functions by $f_1, \ldots, f_T$. For each such function, let P_i be a distribution over $C \times \{0, 1\}$ such that $P_i(X = x, Y = y) = 1/|C|$ if $y = f_i(x)$

and $P_i(X = x, Y = y) = 0$ otherwise. Clearly, we have $L(f_i, P_i) = 0$ for any $i \in [1 : T]$. On the other hand, it can be shown that for every algorithm A, there exists some i such that

$$\mathbb{E}_{Z^n \sim P_i^n}[L(A(Z^n), P_i)] \geq 1/4, \tag{1}$$

where the $1/4$ on the right-hand-side of the inequality comes from the fact that i) with probability $1/2$ you will encounter an unseen instance, and ii) for those unseen instances you cannot do anything better than a random guess, which leads to a $1/2$ error probability. Thus, we can conclude that for any learning algorithm A, there exists a distribution P such that $L(f, P) = 0$ for some f but

$$P^n(L(A(Z^n), P) \geq 1/8) \geq 1/7, \tag{2}$$

where (2) follows immediately from (1) by using the reverse Markov inequality.

1.2 No Free Lunch and Prior Knowledge

This theorem states that if the sample size n is smaller than $|\mathcal{X}|/2$, then for every learner, there exists a task (distribution) 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 $n \geq 8 \log(7|\mathcal{H}|/6)$.

To relate the NFL theorem to the need for prior knowledge, consider a learner over the hypothesis class $\mathcal{H}$ of all the functions h from an infinite domain set $\mathcal{X}$ to $\{0, 1\}$. This class represents lack of prior knowledge: Every possible function from the domain to the label set is considered a candidate. According to the NFL theorem, no matter which learning algorithm is used and how large the training sample is, the output hypothesis from $\mathcal{H}$ will fail on some learning task. Therefore, this class is not PAC learnable. This said, when approaching a particular learning problem, reflected by the distribution P, we should have some prior knowledge on P.

Incorporating Prior Knowledge: Discriminative vs. Generative Approach.

- One type of such prior knowledge is that P comes from some specific parametric family of distributions and our goal is to estimate the parameters of the model — this is known as the *generative approach*.
- Another type of prior knowledge on P, which we assumed when defining the PAC learning model, is that there exists h in some predefined hypothesis class $\mathcal{H}$, such that $\min_{h \in \mathcal{H}} L(h, P)$ is small. In a sense, this assumption on P is a prerequisite for using the PAC model, in which we require that the risk of the output hypothesis will not be much larger than $\min_{h \in \mathcal{H}} L(h, P)$; indeed, if $\min_{h \in \mathcal{H}} L(h, P)$ is very bad compared to the Bayes risk $\min_h L(h, P)$, then why bother to use $\mathcal{H}$ as a benchmark hypothesis class? This approach of directly optimizing the quantity of interest (the prediction accuracy) instead of learning the underlying distribution, is known as the *discriminative approach*.
- Of course, if we succeed in learning the underlying distribution accurately, we are considered to be "experts" in the sense that we can predict by using the Bayes predictor. The problem is that it is usually more difficult to learn the underlying distribution than to learn an accurate predictor. This was phrased as follows by Vladimir Vapnik in his principle for solving problems using a restricted amount of information: "*When solving a given problem, try to avoid a more general problem as an intermediate step.*"

2 Error Decomposition

An immediate question is then 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 realizable PAC setting), or at

least that the smallest error achievable within the 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. So here we are facing a tradeoff, where we want to make the hypothesis class $\mathcal{H}$ rich enough so that $\min_{h\in\mathcal{H}} L(h,P)$ is small, but at the same time still learnable for the given number of training examples. This tradeoff is best reflected when decomposing the error of an $\mathrm{ERM}_{\mathcal{H}}$ output predictor into two parts as follows.

Let h_{Z^n} be the output predictor of the $\mathrm{ERM}_{\mathcal{H}}$ learner. Then we have

$$L(h_{Z^n},P) - \min_h L(h,P) = \underbrace{\min_{h\in\mathcal{H}} L(h,P) - \min_h L(h,P)}_{L_{\mathrm{app}}} + \underbrace{L(h_{Z^n},P) - \min_{h\in\mathcal{H}} L(h,P)}_{L_{\mathrm{est}}}.$$

- The left-hand-side of the equation, $L(h_{Z^n},P) - \min_h L(h,P)$, is called the excess risk; it is always nonnegative since the Bayes risk $\min_h L(h,P)$ is the smallest possible risk for any predictor.
- The first term on the right-hand-side, L_{app}, is the approximation error. It measures how much excess error we have because we restrict ourselves to a specific class $\mathcal{H}$, namely, how much inductive bias we have. The approximation error does not depend on the sample size and is determined by the hypothesis class $\mathcal{H}$ chosen. Enlarging $\mathcal{H}$ can decrease the approximation error.
- The second term on the right-hand-side, L_{est}, is the estimation error. It measures the difference between the smallest error achievable within $\mathcal{H}$ and the error achieved by the ERM predictor. The estimation error happens 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, L_{est} increases (logarithmically) with $|\mathcal{H}|$ and decreases with n.

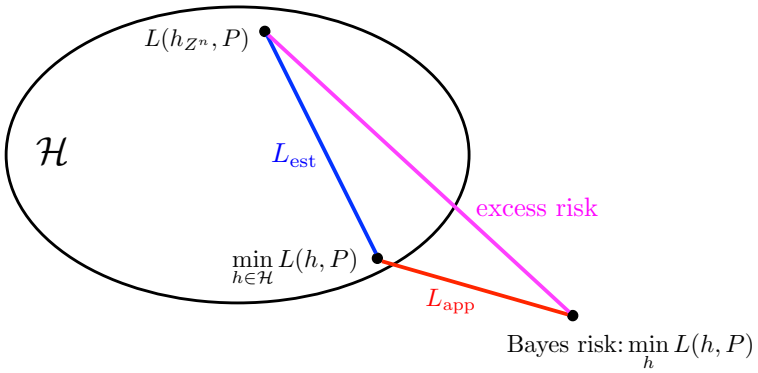

Figure 1: Error decomposition.

2.1 Bias-Complexity Tradeoff

Since our goal is to minimize the total risk, we naturally face 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. This tradeoff is illustrated in Fig. 2, which plots, for a fixed training sample size n, how the true risk, approximation error, and estimation error vary with the model complexity.

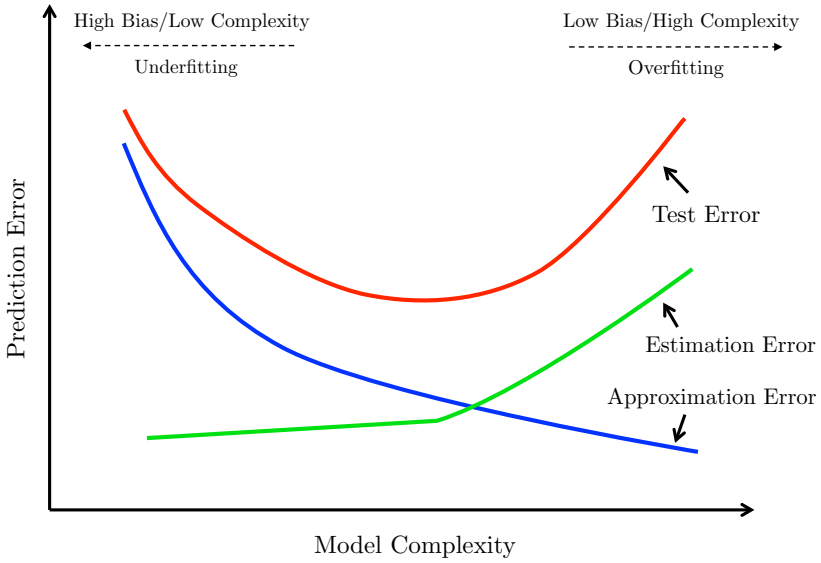

Figure 2: Bias-Complexity tradeoff: approximation error vs. estimation error.

One can also view the bias-complexity tradeoff from a different angle, by decomposing the true risk into training error and generalization error:

$$L(h_{Z^n}, P) = \underbrace{\min_{h \in \mathcal{H}} L(h, P_n)}_{\text{training error}} + \underbrace{L(h_{Z^n}, P) - L(h_{Z^n}, P_n)}_{\text{generalization error}}. \tag{3}$$

Fig. 3 illustrates the bias-complexity tradeoff from the angle of (3).

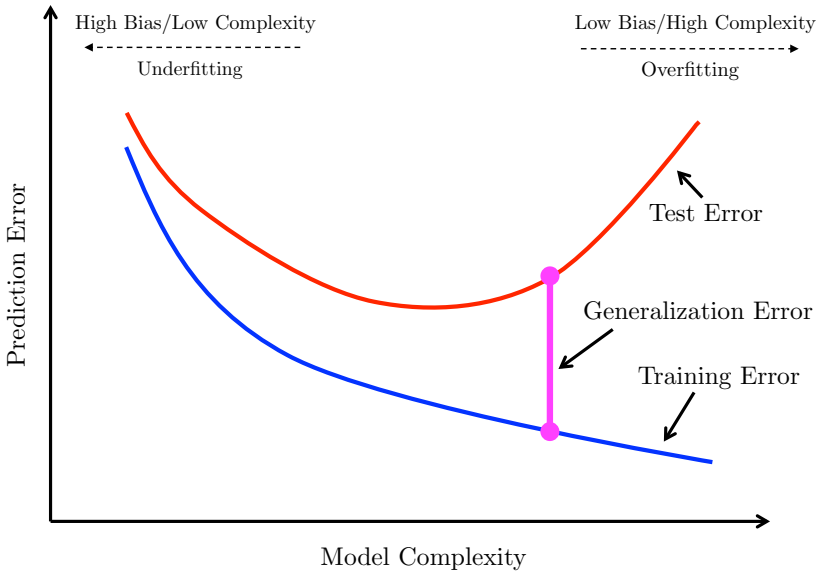

Figure 3: Bias-Complexity tradeoff: training error vs. generalization error.