Foundations for Machine Learning

L. Y. Stefanus

TU Dresden, June-July 2018

Slides 05

Reference

- Shai Shalev-Shwartz and Shai Ben-David. UNDERSTANDING MACHINE LEARNING: From Theory to Algorithms. Cambridge University Press, 2014.

The Bias-Complexity Tradeoff

The No-Free-Lunch Theorem

The Bias-Complexity Tradeoff

- We have seen 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 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 H is a low-error model for the task).
- For example, in our papayas problem, on the basis of previous experience with other fruits, we may assume that some rectangle in the color-softness 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?
- What is a universal learner? A specific learning task is defined by an unknown distribution D over $X \times Y$, where the goal of the learner is to find a predictor $h: X \rightarrow Y$, whose risk, $L_D(h)$, is small enough. The question is therefore whether there exists a learning algorithm A and a training set size m, such that for every distribution D, if A receives m i.i.d. instances from D, there is a high chance it outputs a predictor h that has a low risk.

- We will study the No-Free-Lunch theorem which states that no such universal learner exists.

Theorem (No Free Lunch)

Let A be any learning algorithm for the task

of binary classication with respect to the 0-1 loss function over a domain X. Let m be any number smaller than $|X|/2$, representing a training set size. Then, there exists a distribution D over $X \times \{0,1\}$ such that:

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

- 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.
- In this case, a successful learner would be an ERM_H learner w.r.t. any finite hypothesis class that contains f and whose sample size satisfies the condition $m \geq 8 \ln(7 |H| / 6)$, where $\epsilon = \frac{1}{8}, \delta = \frac{6}{7}$.

Proof of No-Free-Lunch Theorem

1. Let C be a subset of 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 some target function f, that would contradict the labels that A(S) predicts on the unobserved instances in C.
2. Note that there are $T = 2^{2m}$ possible functions from C to {0,1}. Let us denote these functions by $f_1, \ldots, f_T$. For each such function, let D_i be a distribution over C × {0,1} defined by $D_i(\{(x,y)\}) = \begin{cases} 1/|C| & \text{if } y = f_i(x) \\ 0 & \text{otherwise} \end{cases}$

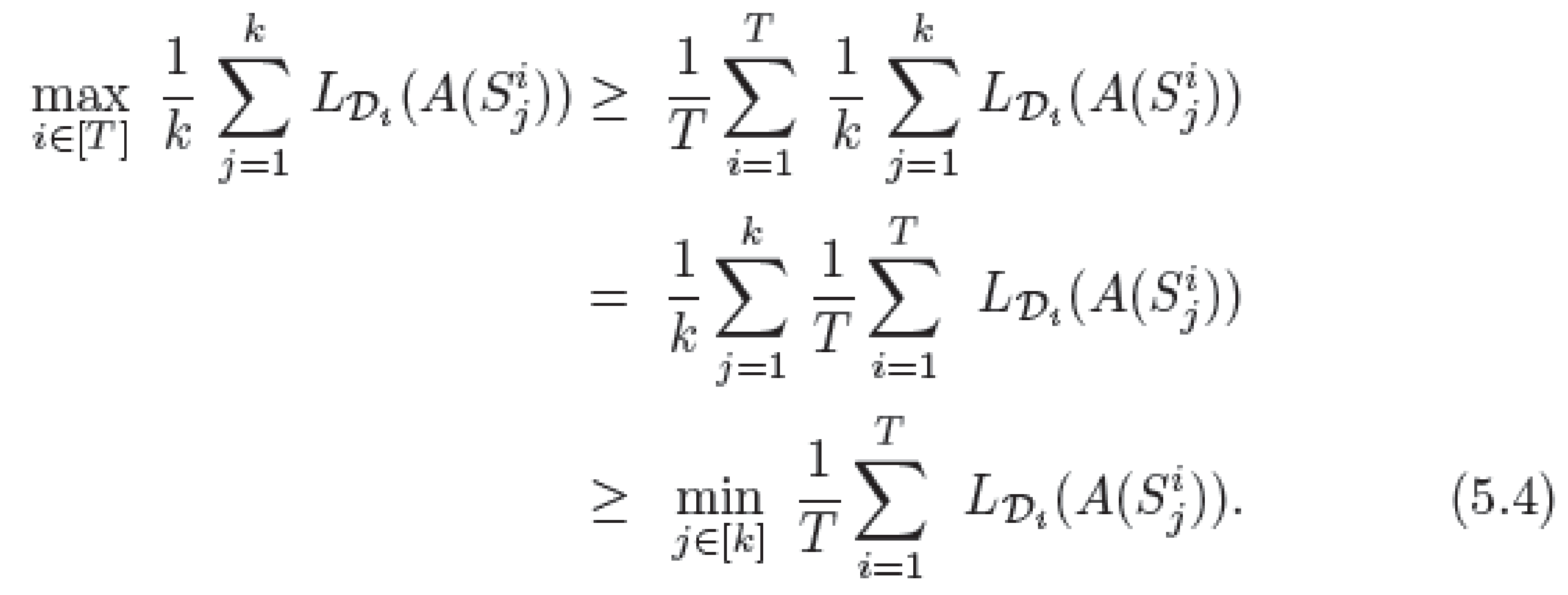
3. That is, the probability to choose a pair (x, y) is $1/|C|$ if the label y is indeed the true label according to f_i, and the probability is 0 if $y \neq f_i(x)$. Therefore, $L_{D_i}(f_i) = 0$.
4. We will first show that for every algorithm, A, that receives a training set of m instances from $C \times \{0,1\}$ and returns a function $A(S): C \rightarrow \{0,1\}$, it holds that

$$\max_{i \in [T]} \mathop{\mathbb{E}}_{S \sim D_i^m} [L_{D_i}(A(S))] \geq \frac{1}{4}. \qquad (5.1)$$

5. There are $k = (2m)^m$ possible sequences of m instances from C. Let us denote these sequences by $S_1, \ldots, S_k$. Also, for $S_j = (x_1, \ldots, x_m)$ 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 D_i then the possible training sets that A can receive are S_1^i, …, S_k^i, and all these training sets have the same probability of being sampled. Therefore,

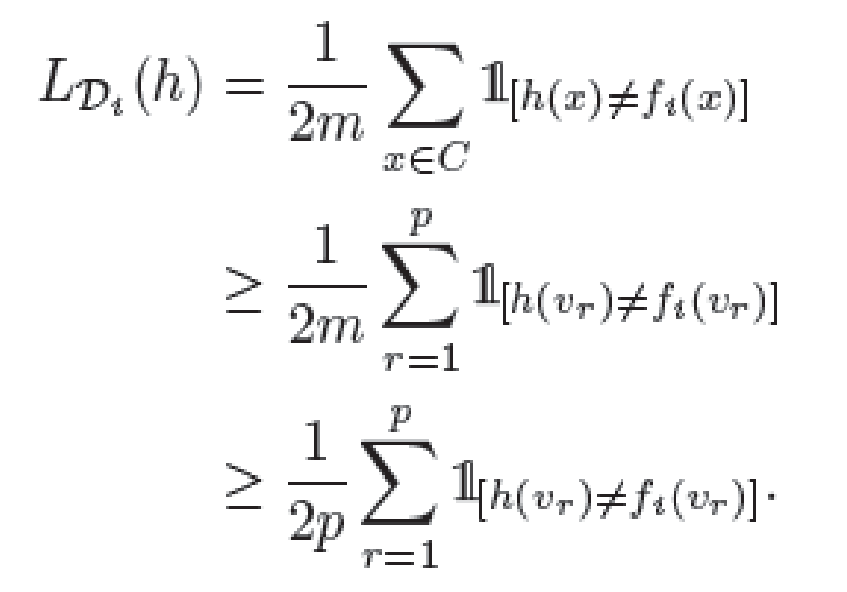
$$\mathbb{E}_{S\sim D_i^m}[L_{D_i}(A(S))] = \frac{1}{k}\sum_{j=1}^{k} L_{D_i}\left(A\left(s_j^i\right)\right). \quad (5.3)$$

6. Using the facts that “maximum” is larger than “average” and that “average” is larger than “minimum,” we have



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

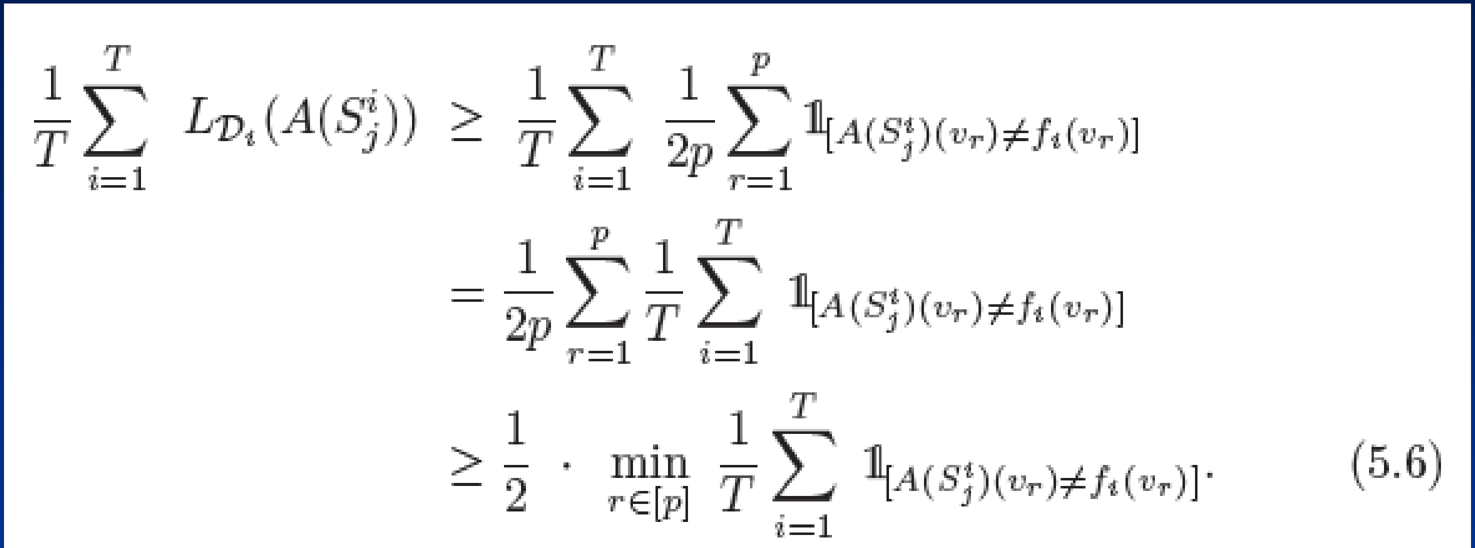
7. Next, we fix some $j \in [k]$. Denote $S_j = (x_1, \ldots, x_m)$ and let $v_1, \ldots, v_p$ be the instances in C that do not appear in S_j. We have that $p \geq m$. Therefore, for every function $h: C \rightarrow \{0,1\}$ and every i we obtain



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

where $\mathbb{1}_{[\text{boolean expression}]}$ denotes indicator function (equals 1 if boolean expression is true and 0 otherwise).

8. Hence,



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

9. Next, we 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

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

10. Combining the previous equation with Equation (5.6), Equation (5.4) and Equation (5.3), we obtain Equation (5.1):

$$\max_{i\in[T]} \underset{S\sim D_i^m}{\mathbb{E}}[L_{D_i}(A(S))] \geq \frac{1}{4}.$$

This means that for every algorithm A' that receives a training set of m instances from $X \times \{0,1\}$ there exists a function $f: X \rightarrow \{0,1\}$ and a distribution D over $X \times \{0,1\}$, such that $L_D(f) = 0$ and

$$\underset{S\sim D^m}{\mathbb{E}}[L_D(A'(S))] \geq \frac{1}{4}. \quad (*)$$

11. Applying Markov’s Inequality from the Theory of Probability to Equation (*), we obtain

$$\mathbb{P}[L_D(A'(S)) \geq 1/8] \geq 1/7$$

which is what we need to prove.

Q.E.D

Markov’s Inequality

Let X be a non-negative random variable and suppose that $E(X)$ exists. For any $t > 0$,

$$\mathbb{P}[X > t] \leq \frac{E(X)}{t}.$$

- Let Z be a random variable that takes values in [0,1]. Assume that E[Z] = μ. Let Y = 1 - Z. Then Y is a non-negative random variable with E[Y] = 1 – E(Z) = 1- μ. Applying Markov's inequality on Y with $a \in (0,1)$, we obtain

$$\mathbb{P}[Z < 1 - a] = \mathbb{P}[1 - Z > a] = \mathbb{P}[Y > a] \leq \frac{E(Y)}{a} = \frac{1-\mu}{a}$$

Therefore, $\mathbb{P}[Z \geq 1 - a] \geq 1 - \frac{1-\mu}{a} = \frac{a+\mu-1}{a}$, which implies that

$$\mathbb{P}[Z \geq a] \geq \frac{\mu - a}{1 - a}$$

No-Free-Lunch Theorem 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 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 H, and in particular the ERM predictor, will fail on some learning task. Therefore, this class is not PAC learnable.

- 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 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 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).