Lecture 3: No Free Lunch Theorem and Infinite Hypothesis Classes

Instructor: Aditya Bhaskara *Scribe: Your Name*

CS 5966/6966: Theory of Machine Learning

January 18th, 2022

Abstract

Today's lecture will begin by explaining why no algorithm can learn every possible function and then discuss what hypotheses are learnable with PAC algorithms.

1 Introduction

Recap of last time:

Formally defining learning:
Notation:

- X is the space of all possible inputs
- Inputs are taken from X according to some distribution D over X.
- Y is the set of all labels/outputs

We'll mostly be considering binary classification, where given some inputs taken from X we return 0 or 1.

Risk, sometimes also called Loss, is, in English: the probability that for some unknown input from X, taken i.i.d. according to distribution D, our hypothesis produces the wrong output. In a very general sense, basically all learning seeks to minimize this Risk.

In mathematical terms, Risk is:

$$R_D(h) = \Pr_{X \sim D}(h(x) \neq f(x))$$

Learnability:
A concept is considered learnable if there exists an efficient algorithm A which can produce a hypothesis with less risk than some acceptable error bound ϵ given m samples from D (the distribution over the space of all possible inputs X) with their labels (the corresponding ground truths in Y)

Today's material: A hypothesis class (a.k.a. concept class) is a set of functions each mapping X to Y.

No free lunch theorem: There is no universal learning algorithm which can learn any arbitrary function as a hypothesis to best fit the data.

Considering the "No Free Lunch Theorem", we can't have some completely general learner who just always learns the best possible function, we must instead pick a hypothesis class which restricts the types of functions we can learn. Our problem then is what type of function is best to learn? Or, to phrase it differently, within what hypothesis class should we search?

2 Common ML assumptions

The old approach from the 90's was to assume if you can pick the right feature set then any data is linearly separable. This has advantages in that the hypothesis class of linear separators is pretty simple.

This linearly separable approach works relatively well, but if we have more complex data, we may need a more complex model to fit it well. Thus, we'll sometimes want a complex model like a neural network with many layers instead. Balancing between complex models which can fit complex data very well but risk overfitting or simpler models which are easier to train and don't harder to overfit can be very tricky and is a large part of machine learning.

3 No Free Lunch Theorem

Theorem informally: there is no universal learner, even if it's allowed to be inefficient, even for binary classification.

Theorem: consider some algorithm for binary classification which uses up to $\frac{|X|}{2}$ training samples taken from X. Then there exists some distribution D over X and a hypothesis h : X $\leftarrow \{0,1\}$ that cannot be learned by A. In other words, there is some h such that the hypothesis h' produced by A after seeing up to $\frac{|X|}{2}$ examples from X where $R_D = Pr_{X\sim D}(h(x) \neq h'(x)) \geq \frac{1}{8}$ and this is produced with probability at least $\frac{1}{8}$

What this means practically, is that if there is not any constraint on what class of functions A is trying to learn, or if algorithm A is not aware of any constraints on what is in H, than regardless of what the actual hypothesis class is it will not be PAC learnable by A.

Informal Proof:
The general idea is that there are simply too many boolean functions on X. we'll ecamine a case where a learner sees —X— / 16 training examples and show that this results in a greater than 1/16 chance of the hypothesis not giving the right results.

If there's X inputs and we make a classification of some size m subset of x, then there are $\binom{n}{m}$ possible inputs to A. There are also $2^{|X|}$ possible boolean functions for mapping inputs to Y. So the size of the space we are mapping all possible inputs to is $2^{|}n|$, which is much larger than $\binom{n}{n/10}2^{n/10}$. Therefore we can see that most of the possible hypotheses that are possible are not

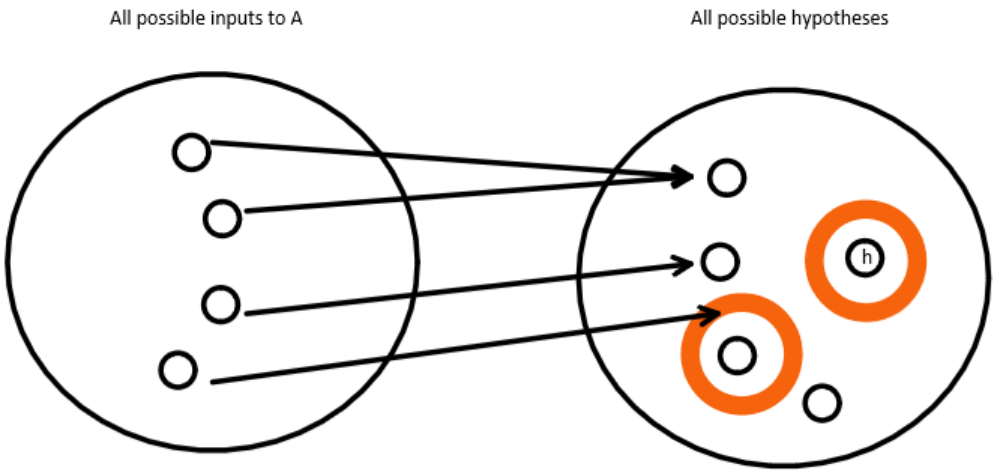

Figure 1: No mapping by A for any input from the LHS produces a function within the acceptable closeness bound of h, therefore h is not PAC learnable.

producible by A for *any* input that the Algorithm could receive.

More formally, we start by saying that $g : X \mapsto \{0,1\}$ (i.e. g is a binary classifier who maps input to true or false). We consider some g to be mapped if there is some input to A which will yield the hypothesis g. G is the set of all g which are mapped. According to the statements in the above paragraph, $|G| \leq \binom{n}{n/10} 2^{n/10}$. Now we calculate how many functions $g : X \mapsto 0,1$ which only differ from g in $1/10$ cases. For any $g \epsilon G$ the number of functions close to it are:

$$\binom{n}{1} + \binom{n}{2} + ... \binom{n}{n/10} \qquad \textit{This sum is} \leq \binom{n+1}{n/16}$$

Now, multiplying that by the size of $|G|$ we get that the number of functions that are close to a function in G is upper bounded by $\binom{n+1}{n/10}\binom{n}{n/10}2^{n/10}$. Since this upper bound is less than 2^n, there must be some hypotheses which differs from any of the mapped functions h' by more than $1/10$ of the inputs, or in other words, there is some hypothesis h which we will not even be able to get 90% close to with A for any input, and thus it is not PAC learnable. □

4 PAC learning and learnability of finite concept classes

Did not get to this section in class today