Machine Learning

CSE427

Mahbub Majumdar
Typeset by:
Syeda Ramisa Fariha

BRAC University
66 Mohakhali
Dhaka, Bangladesh

July 16, 2018

Inspiring Excellence

Superhumungous Thanks

These slides were typeset by Syeda Ramisa Fariha.

Without her tremendous dedication, these slides would not exist.

Motivation of the No Free Lunch Theorem

- **Motivation:**

  a) Training data can mislead the trainer

  b) Restrict search to some $\mathcal{H}$

- How to choose $\mathcal{H}$?

  $\Rightarrow$ USE PRIOR KNOWLEDGE

- For example, an American baseball scout is scouting cricket players in Bangladesh

Motivation of the No Free Lunch Theorem

- S/he will choose a (*pace*, *accuracy*) rectangle because *pace* and *accuracy* are the most important qualities of a US baseball pitcher.
- S/he is using their prior knowledge about what makes a good pitcher.
- The scout knows from prior experience what features to emphasize.
- But baseball is not the same as cricket.
- In cricket the ball bounces off the ground – so the scout probably won't select the the right $\mathcal{H}$.

Prior Knowledge

- QUESTION: But, is prior knowledge absolutely necessary?
- Is there a *super learner* who can learn just by observing the data?
- Specifically, is there a *learning algorithm* A and *training set* of size m, such that for every distribution D, that is outputs a low risk h?
- If this were true, then a future quantum computer could analyze data and using its unique algorithm, find the right predictor for every problem.
- No specialized knowledge, or intuition would be required.

No Free Lunch Theorem

- The *No Free Lunch Theorem* states that no universal learner exists
- There is a distribution for which the learner fails.
- We will specialize to binary tasks.
- Failure means that: after receiving iid samples from the distribution

  $\implies$ The output hypothesis will have *large risk*.

  $\implies$ Also, there is another learner that will output a low risk hypothesis.
- Thus, we should generally use some *prior knowledge* when faced with a learning problem defined by a distribution D.

No Free Lunch Theorem

- One type of prior knowledge is
  - D comes from a specific parametric family of distributions
  - For example, suppose we want to predict the stock market return for Beximco.
  - Then prior knowledge tells us the that the distribution will be close to Lognormal.
  - This prior knowledge tells us not to consider for example flat distributions.

- Another type of prior knowledge is that there is a hypothesis $h \in \mathcal{H}$ such that, $L_D(h) = \text{small}$.
  - We should therefore try to mimic h.
  - For example, we know Warren Buffet is a good investor.
  - We should therefore try to copy some of his strategies.

The Bias-Complexity Tradeoff

- The error can be decomposed

$$\text{Total Error} = \text{Error in prior knowledge} + \text{Error from overfitting}$$

- Terminology

$$\begin{aligned} \text{Error in prior knowledge} &\equiv \text{approximation error} \\ &\equiv \text{bias} \\ \text{Error from overfitting} &\equiv \text{estimation error} \end{aligned}$$

- There is a tradeoff between *approximation error* and *estimation error*

- Approximation Error Up $\longrightarrow$ Estimation error Down

No Free Lunch Theorem

- **Basic theme:** No learner can succeed on all learning problems without knowing D

Theorem (No Free Lunch)

Let A be any learning problem for the task of binary classification,

Let m be any number smaller than $\frac{|\mathcal{X}|}{2}$. m is the training set size.

Then $\exists$ a distribution D over $X \times 0, 1$ such that,

1. $\exists\ f : X \to \{0, 1\}$ *with* $L_D(f) = 0$
2. *with probability of at least* $\frac{1}{7}$, *over the choice* $S \sim D^m$, *we have* $L_D(A(S)) \geq \frac{1}{8}$

Comments

- Every learner fails on some task that can be successfully learned by another learner

- Trivial successful learner is an *ERM* learner with

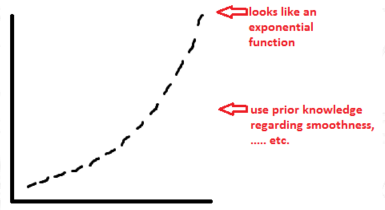
$$\mathcal{H} = \{f,\ \text{other hypotheses}\ \}$$

whose sample size satisfies

$$m \geq \left(\ln \frac{|\mathcal{H}|}{6/7}\right)\left(\frac{1}{1/8}\right)$$

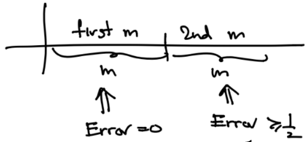
Proof:

- Intuition: any algorithm that observes $\frac{1}{2}$ of the instances in $C \subset X$ has no information on what the labels are for the rest of C.
- $C = subset\ of\ X$, $|C| = 2m$
- **Note:** *Don't assume prior knowledge*
- For example,



No Free Lunch Theorem

- Consider an appropriate example – Coin flipping.
  - ⇒ Flip a coin $2m$ times
  - ⇒ Know the result on the first m tosses
  - ⇒ Can we predict the results of the next m tosses?
- Expected minimal error



$$
\begin{aligned}
\mathbb{E}(\text{error}) &= \frac{1}{2} \times \text{error}(\text{first } m) + \frac{1}{2} \times \text{error}(\text{second } m) \\
&= \frac{1}{2} \times 0 + \frac{1}{2} \times \text{error}(\text{2nd } m) \\
&\geq 0 + \frac{1}{2} \times \frac{1}{2} \\
&= \frac{1}{4}
\end{aligned}
$$

No Free Lunch Theorem

- We can phrase the same argument more mathematically as follows,

$$
\begin{aligned}
\textbf{Error} &= \mathbb{E}_f\left[\underset{S\sim D^m}{\mathbb{E}}\left[\mathrm{Error}(A(S))\right]\right] \\
&= \mathbb{E}_f\left[\underset{S\sim D^m}{\mathbb{E}}\left[L_S(A(S))\right]\right] \\
&= \mathbb{E}_S\left[\mathbb{E}_f\left[\underset{x\sim X}{\mathbb{E}}\left[A(S)(x)\neq f(x)\right]\right]\right] \\
&= \underset{S,x}{\mathbb{E}}\ \underset{f}{\mathbb{E}}\left[A(S)(x)\neq f(x)\ \middle|\ x\in S\right]\mathbb{P}(x\in S) \\
&\quad + \underset{S,x}{\mathbb{E}}\ \underset{f}{\mathbb{E}}\left[A(S)(x)\neq f(x)\ \middle|\ x\notin S\right]\mathbb{P}(x\notin S) \\
&\geq 0+\frac{1}{2}\times\frac{1}{2} \\
&= \frac{1}{4}
\end{aligned}
$$

No Free Lunch Theorem

- Where we used $\mathbb{P}(\chi \notin S) = \frac{1}{2}$ and $\mathbb{E}\Big[A(S)(x) \neq f(x)\Big] \geq \frac{1}{2}$ for all $x \notin S$.

No Free Lunch Theorem

Back to proving NFL theorem:

- S is contained in C
- $|C| = 2m, \ |S| = m$
- There is a target function f that contradicts the labels that $A(S)$ predicts on the unobserved points in C
- f is a sequence since like $\underbrace{010101\ldots01}_{2m}$
- It assigns a 0 or a 1 to all of the $2m$ points of C.
- There are $T = 2^{2m}$ possible functions from $f : C \to \{0, 1\}$ and $\{f\} = \{f_1, f_2, \ldots, f_T\}$

No Free Lunch Theorem

- For each f_i, let D_i be the distribution over $C \times \{0, 1\}$ defined by

$$D_i(x, y) = \begin{cases} \frac{1}{|C|} & \textit{if } y = f(x_i), \quad i \in T, x \in X \\ 0 & \textit{otherwise} \end{cases}$$

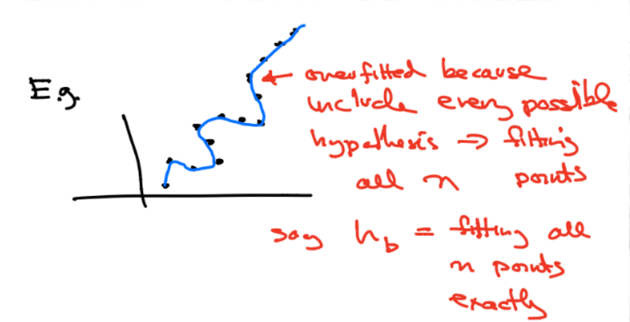
- Here, for each f_i, we are artificially creating a D_i for which f_i is the *true labeling function*
- Thus, the probability of choosing (x, y) is $\frac{1}{|C|}$ if y is the true label and 0 otherwise
- Thus, on C, $L_{D_i}(f_i) = 0$, since by construction f_i is the *true labeling function* for $x \sim D_i$
- Now, **NFL** basically says that, if you include every possible hypothesis (e.g. theory) you don't learn much

The Bias-Complexity Tradeoff

- Learnable problems are problems with good enough hypothesis to fit the problem, but restricted enough to not overfit the sample

- By including so many possible hypotheses → gets lots of *overfitting error*

⇒ That's why there is a *minimum expected error*



No Free Lunch Theorem

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

$$\max_{i \in [T]} \mathop{\mathbb{E}}_{S \in D_i^m} \left[\underbrace{L_{D_i}(A(S))}_{A(S)(x) \neq f_i(x)} \right] \geq \frac{1}{4}$$

- Here f_i is the *true labeling function* for some problem

No Free Lunch Theorem

- Using the fact that there is a problem for which the error is at least 25%, we can show that

$$\mathbb{P}\left[L_D(A'(S)) \geq \frac{1}{8}\right] \geq \frac{1}{7} \tag{1}$$

  This is the the second part of the NFL Theorem.

- This follows from *Markov's Inequality*.

- The basic message is that for every ML algorithm $A \in \mathcal{H}$, there is an ML problem which A does bad on.

Markov Inequality

Markov's inequality:

Suppose Z is a random variable in $[0,1]$ and $\mathbb{E}(Z) = \mu$. Then for any $a \in (0,1)$,

$$\mathbb{P}(Z > 1 - a) \geq \frac{\mu - (1 - a)}{a}$$

Or equivalently,

$$\mathbb{P}(Z > a) \geq \frac{\mu - a}{1 - a} \geq \mu - a$$

Applying the Markov Inequality

- Thus, if for some ML problem, $\mathop{\mathbb{E}}_{S \in D_i^m} \left[\underbrace{L_{D_i}(A(S))}_{A(S)(x) \neq f_i(x)} \right] \geq \frac{1}{4}$

$$
\begin{aligned}
\mathop{\mathbb{P}}_{S \sim D^m}\left[L_D(A(S)) \geq \frac{1}{8}\right] &= \mathop{\mathbb{P}}_{S \sim D^m}\left[L_D(A(S)) \geq (1 - \frac{7}{8})\right] \\
&= \frac{\mathbb{E}(L_D(A(S))) - (1 - \frac{7}{8})}{\frac{7}{8}} \\
&= \frac{\frac{1}{4} - \frac{1}{8}}{\frac{7}{8}} \\
&\geq \frac{\frac{1}{8}}{\frac{7}{8}} \\
&= \frac{1}{7}
\end{aligned}
$$

Applying the Markov Inequality

- This shows that

$$\mathbb{P}\left[L_D(A(S)) \geq \frac{1}{8}\right] \geq \frac{1}{7} \tag{2}$$

Aside: Markov's Inequality

- The expected value of the non-negative random variable Z can be written as,

$$\begin{aligned}
\mathbb{E}(Z) &= \int_0^1 z\,\mathbb{P}(Z=z)dz \\
&= \int_0^a z\,\mathbb{P}(Z=z)dz + \int_a^1 z\,\mathbb{P}(Z=z)dz \\
&\geq \int_a^1 z\,\mathbb{P}(Z=z)dz \\
&\geq a\int_a^1 \mathbb{P}(Z=z)dz \\
&= a\mathbb{P}(Z\geq a)
\end{aligned}$$

- Therefore

$$\mathbb{P}(Z\geq a)\leq \frac{1}{a}\,\mathbb{E}(Z)$$

Aside: Reverse Markov Inequality

Reverse Markov Inequality:

Let $Y = 1 - Z$. Since $0 \leq Z \leq 1$, we have $0 \leq Y \leq 1$.

Then,

$$\mathbb{E}(Y) = \mathbb{E}(1 - Z) = 1 - \mathbb{E}(Z) = 1 - \mu$$

Applying Markov's inequality to Y.

$$\mathbb{P}(Z \leq 1 - a) = \mathbb{P}(1 - Z \geq a) = \mathbb{P}(Y \geq a) \leq \frac{\mathbb{E}(y)}{a} = \frac{1 - \mu}{a}$$

Thus,

$$\mathbb{P}(Z > 1 - a) > 1 - \frac{1 - \mu}{a} = \frac{\mu - (1 - a)}{a}$$

Back to the No Free Lunch Theorem

- Recall that f_i is the true labeling function given that the distribution is D_i
- If the distribution is D_i, then the possible training sets that can be given to the algorithm are

$$\{S_1^i, S_2^i, \ldots, S_k^i\}$$

- How many training sets of size m are there?

$$\text{Number of training sets} = (2m)^m$$

Back to the No Free Lunch Theorem

- A training set where the true labeling function is f_i is denoted by

$$S_j^i = \{x_1, \ldots, x_m\}$$

- For example, suppose x_i is the result of the ith coin toss.

- For example, suppose the 12th training set can, where the true label is f_i, might be denoted by

$$S_{12}^i = \{x_1, x_{16}, x_{22}, \ldots, x_{64}\}$$

- Thus S_{12}^i consists of the first coin toss, the 16th coin toss, the 22nd coin toss, ..., the 64th coin toss.

Back to the No Free Lunch Theorem

- All of the training sets have the same probability of being sampled.

$$\mathop{\mathbb{E}}_{S \sim D_i^m}\left[L_{D_i}(A(S))\right] = \frac{1}{k}\sum_{i=1}^{k} L_{D_i}(A(S_j^i))$$

- Thus

$$\max_{i \in [T]}\left[\frac{1}{k}\sum_{j=1}^{k} L_{D_i}(A(S_j^i))\right] = \max_{i \in [T]}\left[\mathop{\mathbb{E}}_{S \sim D^m}\left(L_{D_i}(A(S^i))\right)\right]$$

- Why are we taking $\max_{i \in [T]}$?

Back to the No Free Lunch Theorem

- The idea is:

- We are looking for a distribution D_i for which $\mathbb{E}[L_{D_i}(A(S))]$ is largest.

- I.e, we are looking for a machine learning problem for which our predictor $A(S)$ gives the largest expected error.

Back to the No Free Lunch Theorem

- All we are trying to show is that there exists at least 1 problem for which our universal algorithm $A(S)$ fails.
- Note: MAX OF SETS $\geq$ AVERAGE OF SETS $\geq$ MINIMUM OF SETS
- Therefore,

$$\begin{aligned}
\max_{i\in[T]} \frac{1}{k}\sum_{j=1}^{k} L_{D_i}(A(S_j^i)) &\geq \underbrace{\frac{1}{T}\sum_{i=1}^{T}}_{\text{average}} \frac{1}{k}\sum_{j=1}^{k} L_{D_i}(A(S_j^i)) \\
&= \frac{1}{k}\sum_{j=1}^{k}\frac{1}{T}\sum_{i=1}^{T} L_{D_i}(A(S_j^i)) \\
&\geq \min_{j\in[k]} \frac{1}{T}\sum_{i=1}^{T} L_{D_i}(A(S_j^i))
\end{aligned}$$

Back to the No Free Lunch Theorem

- Here, we have fixed the $j \in [k]$ that gives the minimum of $\mathbb{E}_f(L_D(A(S))$
- Call $S_j = (x_1, \ldots, x_m)$
- $v_1, \ldots, v_p$ are examples/instances in C that don't appear in S_j
- $p \geq m$, since $|C| = 2m, |S| = m$.
- For example, if there are repetitions in S_j, such that $S_j = (x_1, \ldots, x_1)$, then $p > m$

Back to the No Free Lunch Theorem

- For every $h \in \mathcal{H}$

$$
\begin{aligned}
L_{D_i}(h) &= \frac{1}{2m} \sum_{x \in C} \mathbb{1}\Big(h(x) \neq f_i(x)\Big) && \mathbb{1} = \text{Indicator function} \\
&\geq \frac{1}{2m} \sum_{r=1}^{p} \mathbb{1}\Big(h(v_r) \neq f_i(v_r)\Big) && \text{Because less points in } v_1, \ldots, v_p \\
&\geq \frac{1}{2p} \sum_{r=1}^{p} \mathbb{1}\Big(h(v_r) \neq f_i(v_r)\Big) && \text{since } p \geq m
\end{aligned}
$$

Back to the No Free Lunch Theorem

- Thus,

$$
\begin{aligned}
&\frac{1}{T}\sum_{i=1}^{T} L_{D_i}(A(S_j^i)) \\
&\geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{2p}\sum_{r=1}^{p}\mathbb{1}\Big(A(S_j^i)(v_r) \neq f_i(v_r)\Big), \text{ where } h(v_r) \equiv A(S_j^i)(v_r) \\
&= \frac{1}{2p}\sum_{r=1}^{p}\frac{1}{T}\sum_{i=1}^{T}\mathbb{1}\Big(A(S_j^i)(v_r) \neq f_i(v_r)\Big) \\
&\geq \frac{1}{2}\min_{r\in[p]}\frac{1}{T}\sum_{i=1}^{T}\mathbb{1}\Big(A(S_j^i)(v_r) \neq f_i(v_r)\Big) \text{ choosing the minimum term}
\end{aligned}
$$

Back to the No Free Lunch Theorem

- As before, fix the r that gives minimum contribution

- Now, partition the different hypothesis functions f_i into disjoint pairs $(f_i, f_{i'})$.

- Choose the partitioning to satisfy,

  1. for every $c \in C, f_i(c)$ and $f_i(C)$ are different if and only if c is outside of the training set.

  2. This means the f_i and $f_{i'}$ agree on the training set. Thus, $S_j^i = S_j^{i'}$ (i.e. the y values for $(x_1, \ldots, x_m)$ are the same. For example, $f_i(x_1) = f_{i'}(x_2), f_i(x_2) = f_{i'}(x_2), \ldots$ etc.

Back to the No Free Lunch Theorem

- This implies that if the condition $A(S_j^i)(v_r) \neq f_i(v_r)$ holds,

  then, if $A(S_j^i)(v_r) = 1, \implies f_i(v_r) = 0,$

  and the condition $A(S_j^i)(v_r) \neq f_i(v_r)$ will hold, since, then $f_i(v_r) = 0.$

- If $A(S_j^i)(v_r) \neq f_i(v_r)$ doesn't hold, then $A(S_j^i)(v_r) \neq f_{i'}(v_r)$

- Thus,

$$\mathbb{1}\Big(A(S_j^i)(v_r) \neq f_i(v_r)\Big) + \mathbb{1}\Big(A(S_j^i)(v_r) \neq f_{i'}(v_r)\Big) = 1$$

Back to the No Free Lunch Theorem

- Since, $S_j^i = S_j^{i'}$

$$\mathbb{1}\left(A(S_j^i)(v_r) \neq f_i(v_r)\right) + \mathbb{1}\left(A(S_j^{i'})(v_r) \neq f_{i'}(v_r)\right) = 1$$

- Averaging over all the functions $f \in \mathcal{H}$

$$\frac{1}{T}\sum_{i} \mathbb{1}\left(A(S_j^i)(v_r) \neq f_i(v_r)\right) = \frac{1}{T}\sum_{i'} \mathbb{1}\left(A(S_j^i)(v_r) \neq f_{i'}(v_r)\right)$$

- Thus,

$$\frac{1}{T}\sum_{i=1}^{T} \mathbb{1}\left(A(S_j^i)(v_r) \neq f_i(v_r)\right) = \frac{1}{2}$$

Back to the No Free Lunch Theorem

- Substituting everything in,

$$\begin{aligned}
\max_{i\in[T]} \left[\mathop{\mathbb{E}}_{S\sim D^m} L_{D_i}(A(S)) \right] &\geq \min_{j\in[k]} \frac{1}{T} \sum_{i=1}^{T} L_{D_i}(A(S_j^i)) \\
&\geq \frac{1}{2} \min_{r\in[p]} \frac{1}{T} \sum_{i=1}^{T} \mathbb{1}\Big(A(S_j^i)(v_r) \neq f_i(v_r) \Big) \\
&\geq \frac{1}{2} \times \frac{1}{2} \\
&= \frac{1}{4}
\end{aligned}$$

- Thus our predictor $h_a = A(S_j^i)$ which does well on D_a fails on D_i.

Consequences of the NFL Theorem

- Suppose, we have no *prior knowledge* for a binary prediction problem.
- Then, we should consider all possible hypotheses $h_i, \ i \in [T]$ where $T = 2^{\text{number of points}}$
- Every possible $h_i \in \mathcal{H}$ is then a possible best predictor for our problem
- A procedure such as ERM will output one of the h in $\mathcal{H}$ as our predictor.
- The NFL theorem says that, our hypothesis (such as the ERM predictor) will fail on some machine learning task
- Thus, $\mathcal{H}$ is not PAC Learnable.

Infinite Domain Sets

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

Proof:

- Use proof by contradiction. Assume that $\mathcal{H}$ is learnable
- Choose for example, $\epsilon = \frac{1}{8}$ and $\delta = \frac{1}{7}$.
- By definition of PAC learnability, $\exists$ an algorithm A and $m(\epsilon, \delta)$ such that for D over $X \times \{0,1\}$, we have $L_D(A(S)) \leq \epsilon$ with probability $1 - \delta$.

  (Suppose, we assume realizability, $\exists f : X \to \{0,1\}$ such that $L_D(f) = 0$)

Infinite Domain Sets

- NFL states that since $|X| \geq 2m$, for every algorithm A, there is a distribution D, such that with probability more than $\delta = \frac{1}{7}$ that, $L_D(A(S)) > \frac{1}{8} = \epsilon$.

- CONTRADICTION: $\mathcal{H}$ is not PAC Learnable.

- To prevent this failure,

  $\Rightarrow$ Use a prior in terms of conditional probability.

  $\Rightarrow$ Use prior knowledge.

- This will help us avoid distributions that will cause us to fail

- Impose prior knowledge by restricting the hypothesis class $\mathcal{H}$

$$\mathbb{P}(h \mid \text{prior knowledge}) = \frac{\mathbb{P}(h \cap \text{prior knowledge})}{\mathbb{P}(\text{prior knowledge})}$$

How To Choose A Good Hypothesis Class?

- To choose a good hypothesis class: include enough hypothesis such that it includes the hypothesis with no error (in *PAC* context) or small error (in *agnostic PAC* context)

- But including the richest $\mathcal{H}$, which contains all possible hypothesis leads to failure $\rightarrow$ as just seen

- This leads to a trade-off (bigger/smaller $\mathcal{H}$).

The Bias-Complexity Tradeoff

- Decompose the error of an $ERM_{\mathcal{H}}$ predictor as follows

$$L_D(h_S) = \epsilon_{app} + \epsilon_{est}$$

where

$$\begin{aligned}
\epsilon_{app} &= \text{approximation error} \\
\epsilon_{est} &= \text{estimation error}
\end{aligned}$$

and

$$\begin{aligned}
\epsilon_{app} &= \min_{h \in \mathcal{H}} L_D(h) \\
\epsilon_{est} &= L_D(h_S) - \epsilon_{app}
\end{aligned}$$

- Recall for the agnostic case,

$$L_D(h) \leq \min_{h \in \mathcal{H}} L_D(h) \; + \; \epsilon$$

Here,

$$\begin{aligned} \min_{h \in \mathcal{H}} L_D(h) &= \epsilon_{app} \\ \epsilon &= \epsilon_{est} \end{aligned}$$

Approximation Error (Inductive bias)

- Risk comes from restricting to $\mathcal{H}$

- Doesn't depend on sample size, it is determined by $\mathcal{H}$

- Making $\mathcal{H}$ bigger can make ϵ_{app} smaller

- Under *Realizability*, $\epsilon_{app} = 0$

Estimation Error

- This is also known as *Empirical Error* or *Training Error*.

- Amount of ϵ_{est} depends on $|S|$

$$\epsilon_{est} \sim \frac{1}{m}$$

- For finite $\mathcal{H}$, ϵ_{est} depends on $|\mathcal{H}|$'s complexity

$$\epsilon_{est} \sim \log |\mathcal{H}|$$

Tradeoff between Estimation and Approximation Error

- Since,

$$\epsilon_{est} \sim -\epsilon_{app}$$

and

$$\begin{aligned} \epsilon_{est} &= \text{complexity} \\ \epsilon_{app} &= \text{bias} \end{aligned}$$

$\implies$ This gives us a

$$\boxed{\textit{Bias-Complexity Tradeoff Issue}}$$

Tradeoff between Estimation and Approximation Error

- $|\mathcal{H}|$ large $\rightarrow \epsilon_{app} \downarrow$, but $\epsilon_{est} \uparrow$, because of *overfitting*

- $|\mathcal{H}|$ small $\rightarrow \epsilon_{app} \uparrow$, but $\epsilon_{est} \downarrow$, because of *underfitting*

- Suppose, $\mathcal{H} = \{h\}$, h $=$ *Bayes' optimal classifier*

  This is a good choice, but this classifier depends on knowing the unknown distribution D

- Goal of *learning Theory:* Make $\mathcal{H}$ as rich as possible while keeping *estimation error* small

$\Rightarrow$ Design good hypothesis classes for which ϵ_{app} not large

Tackling unfamiliar problems

- When facing an unfamiliar problem
  - ⇒ Don't know the optimal classifier or how to construct it
  - ⇒ Do have some prior knowledge
  - ⇒ This enables us to design hypothesis classes with ϵ_{app} and ϵ_{est} not too high
- For example, when looking for life in the universe
  - ⇒ No idea what aliens are made up of, or what kind of environments they might live in
  - ⇒ Use *prior knowledge* that all known life needs *oxygen* and *water*
  - ⇒ Look for water/oxygen rich planets