

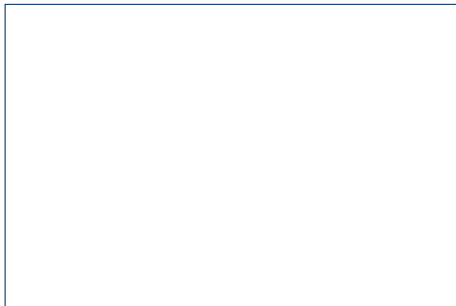
# Machine Learning: Generalization 2

Hao Tang

March 19, 2025

# No free lunch theorem

all functions



# No free lunch theorem

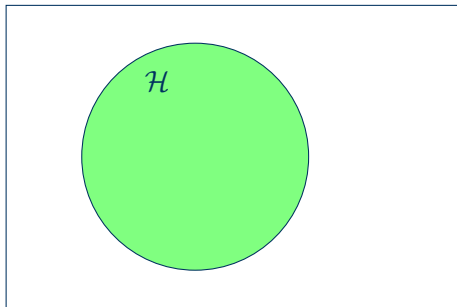
all functions



$\mathcal{H}$

# No free lunch theorem

all functions



# No free lunch theorem

- A cheater is an algorithm that knows the distribution  $\mathcal{D}$ .
- If  $\mathcal{H}$  is the universe, we inadvertently include all the options a cheater has.
- If  $\mathcal{H}$  is the set of all functions,  $\mathcal{H}$  is not PAC learnable.

# Error decomposition

$$L_{\mathcal{D}}(h) = \underbrace{L_{\mathcal{D}}(h) - \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h')}_{\text{estimation error}} + \underbrace{\min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h')}_{\text{approximation error}} \quad (1)$$

- Approximation error is due to the choice of  $\mathcal{H}$ .
- Estimation error is due to not finding the best program in  $\mathcal{H}$ .

# Tradeoff between model complexity and generalization

- When we say we only compare to the best in  $\mathcal{H}$ , we are comparing against  $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$ .
- When  $\mathcal{H}$  is large,  $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$  becomes lower.
- When  $\mathcal{H}$  is the universe of all functions, we cannot learn.
- $\mathcal{H}$  needs to be about the right size.
- $\mathcal{H}$  can actually be a large, but the range of  $A$  needs to be about the right size.
- For example, we can only run a finite number of steps with stochastic gradient descent, so the range we can explore is limited by the algorithm.

# Uniform convergence

A hypothesis class  $\mathcal{H}$  has uniform convergence property if for any distribution  $\mathcal{D}$ , and any  $\epsilon > 0$  and  $0 \leq \delta \leq 1$ , there exists  $N > 0$  such that for every  $h \in \mathcal{H}$ ,

$$\mathbb{P}_{S \sim \mathcal{D}^n} [|L_S(h) - L_{\mathcal{D}}(h)| > \epsilon] < \delta \quad (2)$$

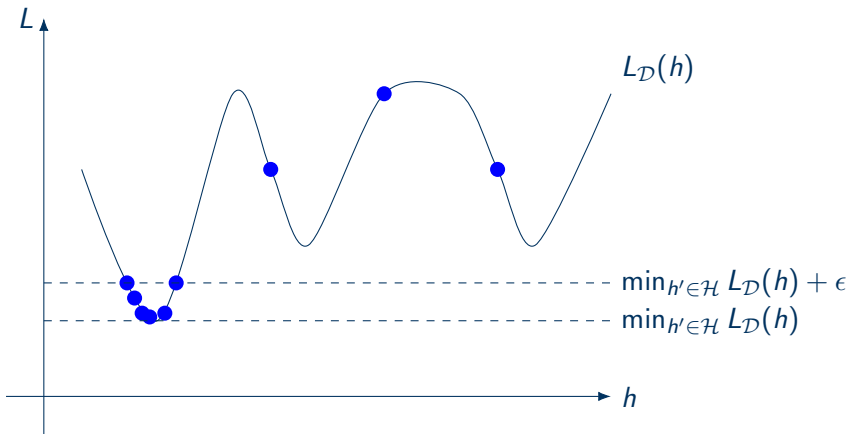
for all  $n \geq N$ .



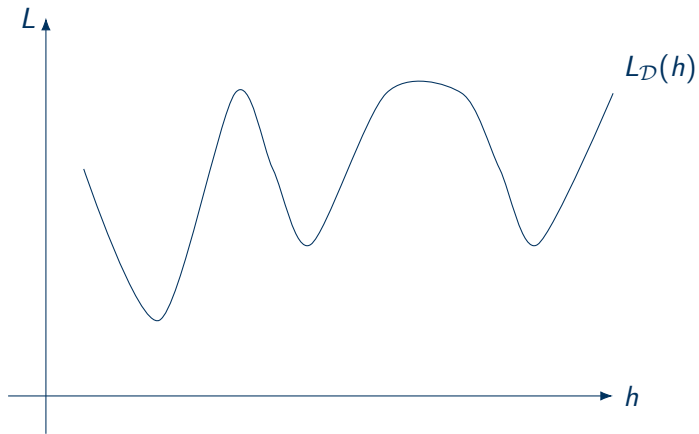
# Uniform convergence

- Uniform convergence assures that the training error and generalization error are not far from each other.
- This has to happen for all  $h \in \mathcal{H}$ , the uniform part (and a strong requirement).

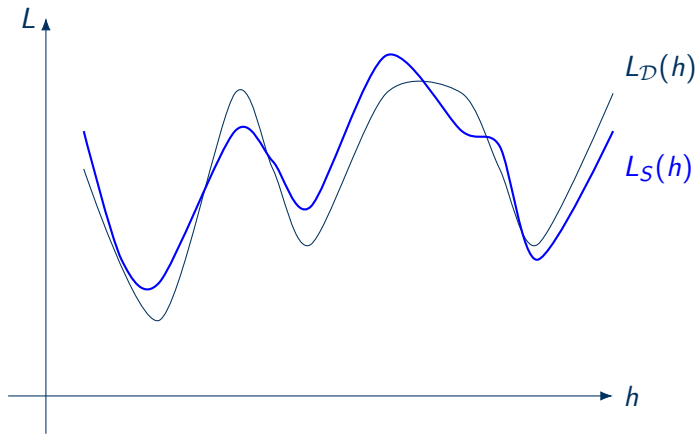
# Comparing PAC learning and uniform convergence



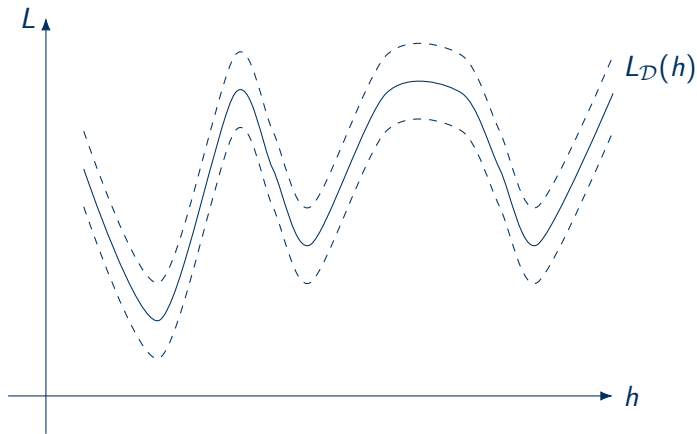
## Comparing PAC learning and uniform convergence



## Comparing PAC learning and uniform convergence



## Comparing PAC learning and uniform convergence



# Uniform convergence

- If we have uniform convergence,

$$L_{\mathcal{D}}(h_{\text{ERM}}) \leq L_S(h_{\text{ERM}}) + \epsilon \leq L_S(h) + \epsilon \leq L_{\mathcal{D}}(h) + \epsilon + \epsilon \quad (3)$$

for any  $h \in \mathcal{H}$ .

- In particular,

$$L_{\mathcal{D}}(h_{\text{ERM}}) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + 2\epsilon. \quad (4)$$

- If  $\mathcal{H}$  has uniform convergence property, then  $\mathcal{H}$  is PAC-learnable with ERM.

# Fundamental theorem of statistical learning



# Fundamental theorem of statistical learning





# Vapnik–Chervonenkis dimension

- VC dimension is the largest number of points that  $\mathcal{H}$  can shatter.
- Given  $n$  data points, there are  $2^n$  ways of label them  $\{+1, -1\}$ .
- A set of  $n$  points is **shattered** by  $\mathcal{H}$  if there is an arrangement of  $n$  points such that classifiers in  $\mathcal{H}$  can produce all  $2^n$  ways of labeling.

## Shattering points in 2D

- We could shatter 3 points with a line in 2D.
- However, we cannot shatter 4 points with a line in 2D.
- The VC dimension of lines in 2D is 3.
- In general, linear classifiers with  $p$  parameters have VC dimension  $p + 1$ .
- We can again shatter 4 points with a 2-layer MLP in 2D.
- Neural networks have larger VC dimension than linear classifiers.
- The sine function has infinite VC dimension.

## VC generalization bounds

- With probability  $1 - \delta$ , for all  $h \in \mathcal{H}$

$$L_{\mathcal{D}}(h) \leq L_S(h) + 2\sqrt{\frac{8d \log(en/d) + 2 \log(4/\delta)}{n}} \quad (5)$$

- $d$  is called the VC dimension.
- For linear classifiers  $\mathcal{H}_{\text{lin}} = \{x \mapsto w^\top x : w \in \mathbb{R}^p\}$ ,  $\text{VC-dim}(\mathcal{H}_{\text{lin}}) = p + 1$ .
- For multilayer perceptrons with  $p$  edges,  $\text{VC-dim}(\mathcal{H}) = O(p \log p)$ .
- These results are independent of learning algorithms.
- In particular, it is independent of how ERM is done.

# Generalization bounds

- Many generalization bounds have the following form.
- With probability  $1 - \delta$ , for all  $h \in \mathcal{H}$

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{S}}(h) + \sqrt{\frac{C(\mathcal{H})}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}. \quad (6)$$

- $n$  is the number of samples.
- $C(\mathcal{H})$  is a capacity measure of  $\mathcal{H}$ .
- There is a family of uniform convergence results.

## Sample complexity

- How many samples do we need to achieve a certain error?
- How large should  $n$  to get to  $\epsilon$ ?

$$\sqrt{\frac{C(\mathcal{H})}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}} \leq \epsilon \quad (7)$$

- In other words,

$$n = O\left(\frac{C(\mathcal{H}) + \log(1/\delta)}{\epsilon^2}\right) \quad (8)$$

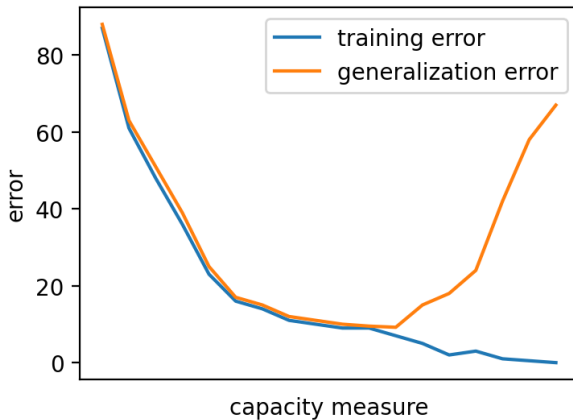
# Interpreting generalization bounds

- VC generalization bounds

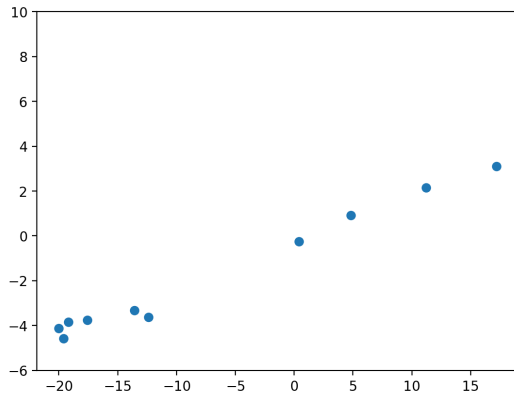
$$L_{\mathcal{D}}(h) \leq L_S(h) + 2\sqrt{\frac{8d \log(en/d) + 2 \log(4/\delta)}{n}} \quad (9)$$

- When  $\mathcal{H}$  is large,  $\min_{h \in \mathcal{H}} L_S(h)$  can be low.
- When  $\mathcal{H}$  is large,  $d$  becomes large.

# Capacity-generalization tradeoff

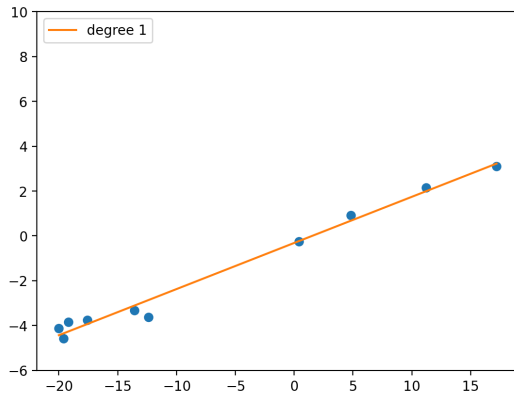


# Capacity-generalization tradeoff





# Capacity-generalization tradeoff



# Capacity-generalization tradeoff

