Lecture 4: Bias, complexity, and the VC dimension

Machine Learning 2025

Federico Chiariotti (federico.chiariotti@unipd.it)

Università
degli Studi
di Padova

Lecture plan

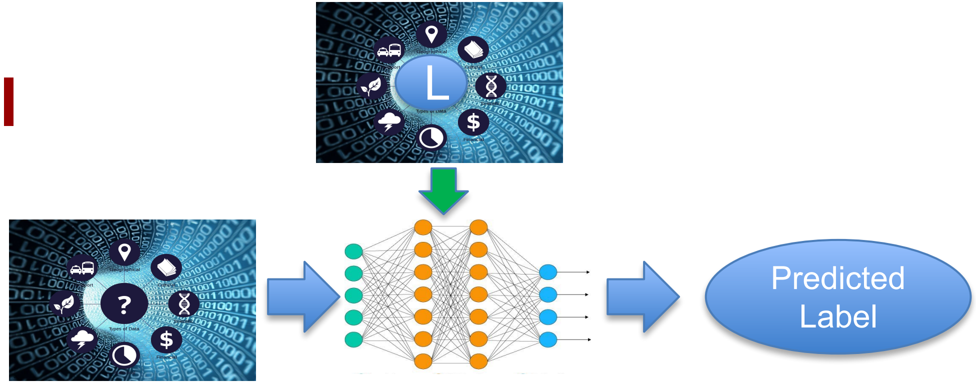
Date	#	Topic	Date	#	Topic	Date	#	Topic
Sep. 30	1	Introduction	Nov. 4	L2	*Model selection*	Nov. 28	12	CNNs
Oct. 7	2	PAC learning	Nov. 7	8	SVMs	???	13	PCA
Oct. 10	3	Uniform convergence	Nov. 11	L3	*Linear models*	Dec. 5	14	Clustering models
Oct. 14	L1	*Python basics*	???	9	Kernels	Dec. 9	L6	*Neural networks*
Oct. 17	4	VC dimension	Nov. 14	10	Ensemble models	Dec. 16	L7	*Clustering*
Oct. 21	5	Model selection	Nov. 18	L4	*SVMs*	Dec. 19	15	Reinforcement
Oct. 24	6	Linear classification	Nov. 21	11	Neural networks	???	L8	*Reinforcement*
Oct. 31	7	Linear regression	Nov. 25	L5	*Random forests*	???	16	Exercises and Q&A

IMPORTANT: no lecture on October 28!

Recap

Università
degli Studi
di Padova

Supervised learning model



- Prediction rule $h : X \rightarrow Y$
  - The learner’s output (hypothesis)
  - $A(S)$: hypothesis produced by algorithm A when it is fed training set S
- Data generation model
  - D is a distribution over X **(unknown to the machine learning algorithm)**
  - Instances are labeled according to $f : X \rightarrow Y$ **(unknown to the ML algorithm)**
  - Training set: sampling according to $D, y_i = f(x_i) \ \forall x_i \in S$
- Success metric
  - Classifier error: probability of predicting the wrong label over D

Empirical Risk Minimization (ERM)

- ➔ Learner outputs $h_S : X \to Y$
- ➔ **Goal: find $\mathbf{h}^*$ that minimizes $\mathbf{L_{D,f}(h)}$**
- ➔ Both D and f are unknown!

ERM: we minimize the loss on the training set $L_S(h) = \frac{\sum_{i=1}^{m} |h(x_i) - y_i|}{m}$

This works if we use a 0-1 loss: the definition can be generalized

The empirical risk is also called training error or training loss

PAC learnability

A hypothesis class H is PAC learnable if there is a function $m_H : (0,1)^2 \rightarrow \mathbb{N}$ and a learning algorithm such that for every $\delta, \varepsilon \in (0,1)$ and for every learning problem D, f with the realizability assumption, the algorithm will satisfy the PAC condition over an IID training set of size $m \geq m_H$

Agnostic PAC learnability

Since we dropped the realizability assumption, we cannot reach 0 loss. What we can try and guarantee is a bound on the loss with respect to the minimum possible loss in the hypothesis class

A hypothesis class H is **agnostic** PAC learnable if there is a function $m_H : (0,1)^2 \to \mathbb{N}$ and a learning algorithm such that for every $\delta, \varepsilon \in (0,1)$ and for every learning problem D, the algorithm will satisfy the following condition over an IID training set of size $m \geq m_H$:

$$L_D(h) \leq \min_{h^* \in H} L_D(h^*) + \varepsilon$$

Definition: ε-representative sets

$$L_S(h) \simeq L_D(h) \forall h \in H \implies \text{PAC}$$

What does it mean for empirical risk to be a good approximation? We need a definition before we can get any results from this intuition

A training set S is ε-representative with respect to domain Z, distribution D, loss function ℓ, and hypothesis class H, if

$$|L_D(h) - L_S(h)| \leq \varepsilon \ \forall h \in H$$

Uniform convergence and PAC learnability

A hypothesis class H has the uniform convergence property with respect to a domain Z and a loss function ℓ if there exists a function $m_H^{\mathrm{UC}} : (0,1)^2 \rightarrow \mathbb{N}$ such that for every $\varepsilon, \delta \in (0,1)$ and every distribution D, any set S of size $m \geq m_H^{\mathrm{UC}}(\varepsilon, \delta)$ is ε-representative with probability higher than $1 - \delta$

If a class H has the uniform convergence property with function m_H^{UC}, we have:

1. The class is agnostically PAC learnable with sample complexity

$$m_H(2\varepsilon, \delta) \leq m_H^{\mathrm{UC}}(\varepsilon, \delta)$$

2. ERM is a valid PAC learning algorithm for H

No free lunch theorem

Let A be any learning algorithm for binary classification, with 0-1 loss, over a domain X. Let m be any number smaller than $\frac{|X|}{2}$. There exists a distribution D over $X \times \{0,1\}$ such that:

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

Corollary: if H is the set of all functions and X is an infinite set, we do **not** have PAC learnability

Consequences

- ➔ We can design a task to make any ML algorithm fail (even if another algorithm is able to solve it)
- ➔ Idea of the proof *(not part of the course, it’s in the book if you’re curious)*: since our training set is smaller than half the domain, we have no idea what happens in the other half, so we can design a function that contradicts our predictor on that part
- ➔ We have to use **prior knowledge** to restrict the hypothesis class

Part 1:
Selecting hypothesis classes

Choosing a good hypothesis class

If the hypothesis class is **too small**:

- ➔ Good generalization: $L_S \simeq L_D$
- ➔ Bad approximation: $L_S \gg 0$

If the hypothesis class is **too big**:

- ➔ Overfitting: $L_D \gg L_S$
- ➔ Good approximation: $L_S \simeq 0$

The amount of available data is the key factor: big hypothesis classes require larger training sets to avoid overfitting (the absolute limit is the no free lunch theorem). If we have to reduce the hypothesis class, we can use prior domain knowledge to make an educated guess

Error decomposition

We can decompose the true error into two separate components:

$$L_D(h_S) = \boxed{\varepsilon_{\text{app}}} + \boxed{\varepsilon_{\text{est}}}$$

$$\varepsilon_{\text{app}} = \min_{h \in H} L_D(h)$$

Approximation error:

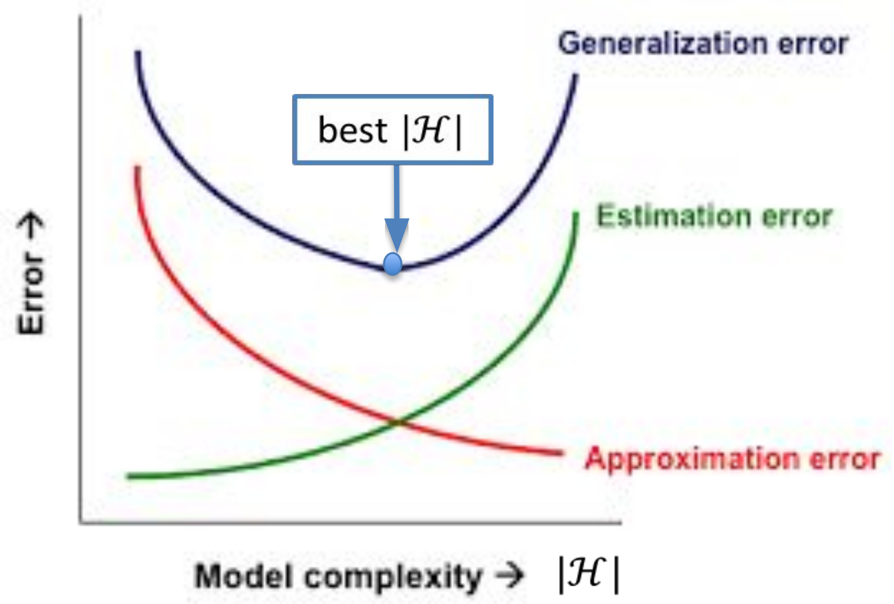
- Minimum risk achievable by the hypothesis class
- Does not depend on the training set
- To decrease it, we need a larger hypothesis class
- Equal to 0 if realizability holds

$$\varepsilon_{\text{est}} = L_D(h_S) - \min_{h \in H} L_D(h)$$

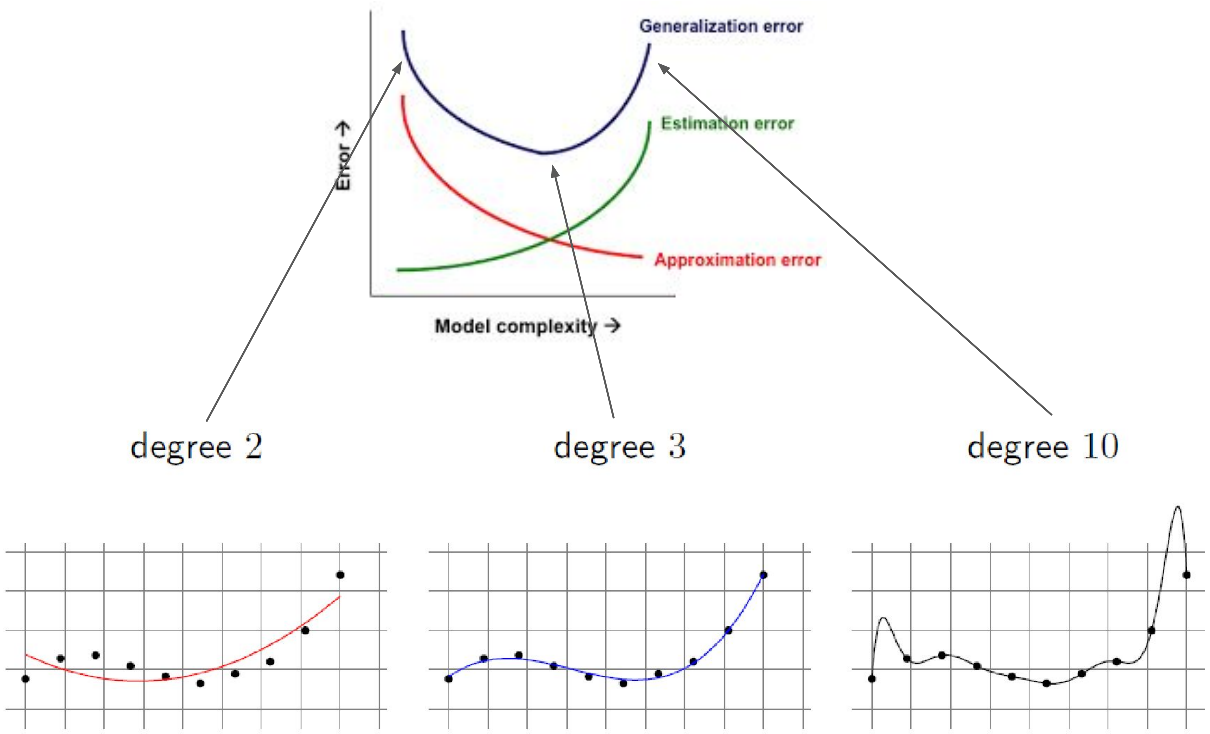
Estimation error:

- Due to the algorithm being unable to find the best hypothesis
- Depends on the training set (and on its size)
- To decrease it, we need a smaller hypothesis class

The bias-complexity trade-off



Example: polynomial fitting



Part 2:
VC dimension

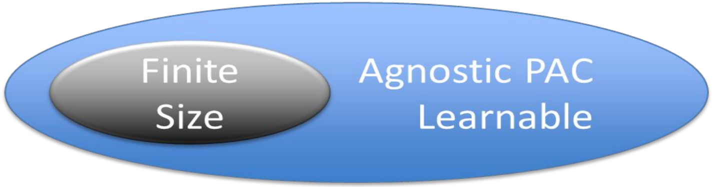
UNIVERSITÀ
DEGLI STUDI
DI PADOVA

What can we learn?

We proved that all finite hypothesis classes are PAC learnable, so we have a solid basis.

However, a lot of hypothesis classes (e.g., linear functions!) have infinite size. Are there classes that can still provide nice bounds, even though they are infinite?

In other words, can we find a looser condition for learnability?



Definition: restriction of a function

Let us define a class of functions $H \subset \{h : X \to \{0, 1\}\}$

Let $C = \{c_1, \ldots, c_m\} \subset X$

The restriction H_C of H over C is the set of functions from C to $\{0, 1\}$ that can be derived from H:

$$H_C = \{\{h(c_1), \ldots, h(c_m)\} : h \in H\}$$

Definition: restriction of a function

Let us define a class of functions $H \subset \{h : X \to \{0,1\}\}$

Let $C = \{c_1, \ldots, c_m\} \subset X$

The restriction H_C of H over C is the set of functions from C to $\{0,1\}$ that can be derived from H:

$$H_C = \{\{h(c_1), \ldots, h(c_m)\} : h \in H\}$$

H_C *shatters* C if it contains **all** the $2^{|C|}$ possible functions from C to $\{0,1\}$

Definition: VC dimension

The Vapnik–Chervonenkis (VC) dimension of H is the maximum possible size d of a set $C = \{c_1, \ldots, c_d\} \subset X$ that can be shattered by H_C

In finite hypothesis classes, sets with a size larger than $\log_2(|H|)$ can never be shattered (there aren't enough functions to cover all points)

Infinite hypothesis classes may shatter sets of arbitrary size: if $VC(H) = \infty$, the class is not PAC learnable due to the corollary we just proved

Shattering and the no free lunch theorem

We can add another corollary to the no free lunch theorem: if m is our training set size, and there exists a set $C \subset X$ of size $2m$ that is shattered by H, any learning algorithm fails over at least one distribution (i.e., there exists a distribution D such that $L_D\left(h_{\mathrm{ERM}}\right) > \frac{1}{8}$ with probability of at least $\frac{1}{7}$)

This is easy to prove: if we consider C as the domain, the restriction H_C is the set of all functions. This then reduces to the statement of the no free lunch theorem

Computing the VC dimension

To show that $VC(H) = d$ we need to prove that:

1. There is a set of size d that is shattered by H ($VC(H) \geq d$)
2. No set of size $d + 1$ is shattered by H ($VC(H) < d + 1$)

The second step is harder than the first (in the first one, we just need to find an example)

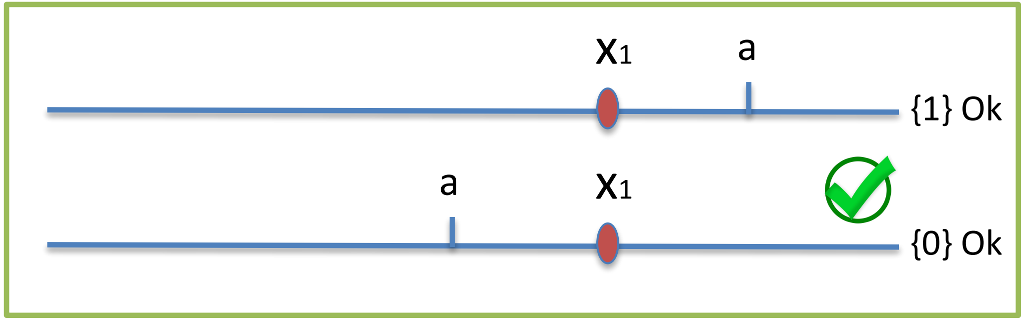
Example: threshold function

Threshold function

$\mathcal{H} = \{h_a : a \in \mathbb{R}\}$

$h_a : \mathbb{R} \rightarrow \{0,1\}$ is:

$$h_A(x) = \begin{cases} 1 \; if \; x < a \\ 0 \; if \; x \geq a \end{cases}$$



$$VC(H) \geq 1$$

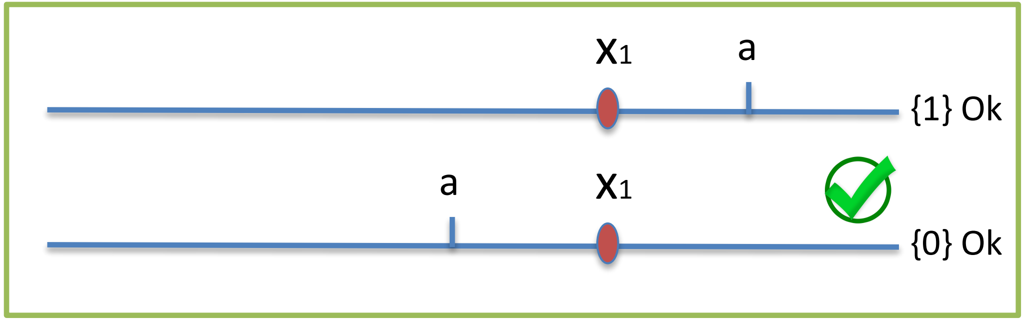
Example: threshold function

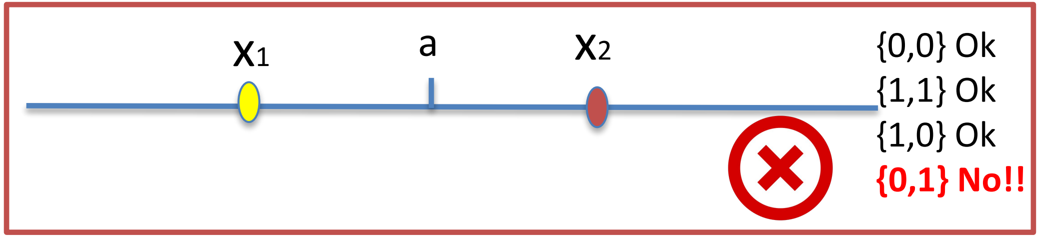
Threshold function

$\mathcal{H} = \{h_a : a \in \mathbb{R}\}$

$h_a : \mathbb{R} \to \{0,1\}$ is:

$$h_A(x) = \begin{cases} 1\ if\ x < a \\ 0\ if\ x \geq a \end{cases}$$





$$VC(H) = 1$$

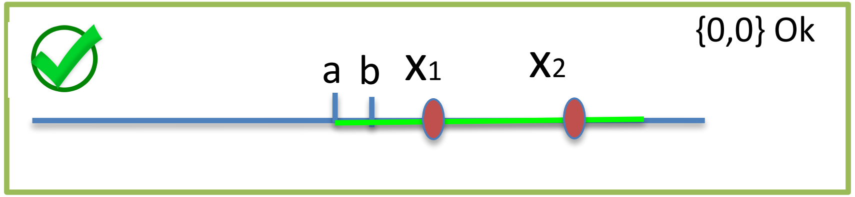
Example: interval function

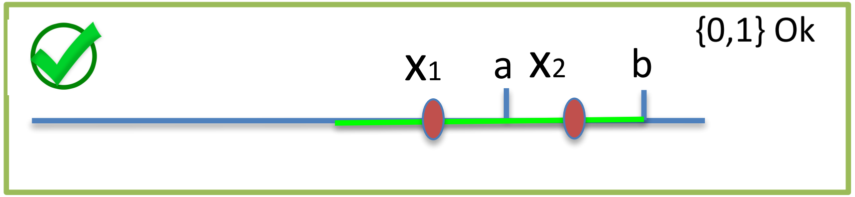
Interval

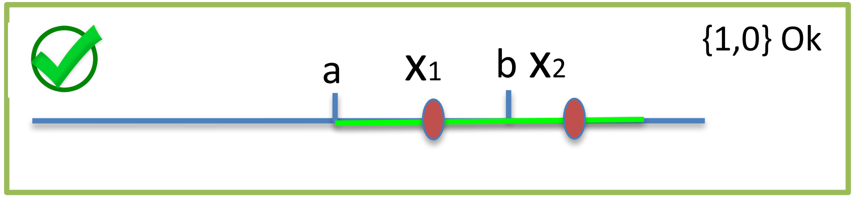
$$\mathcal{H} = \{h_{a,b} : a, b \in \mathbb{R} \ , a < b\}$$

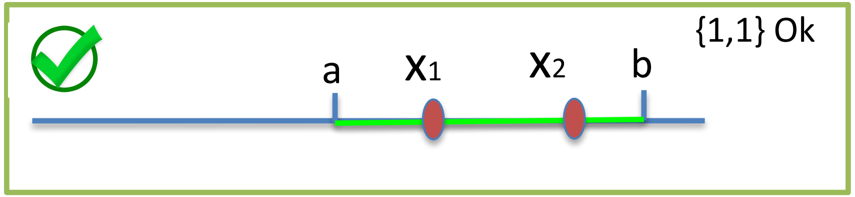
$h_{a,b} : \mathbb{R} \rightarrow \{0,1\}$ is:

$$h_{a,b}(x) = \begin{cases} 1 \ if \ a < x < b \\ 0 \quad otherwise \end{cases}$$









$$VC(H) \geq 2$$

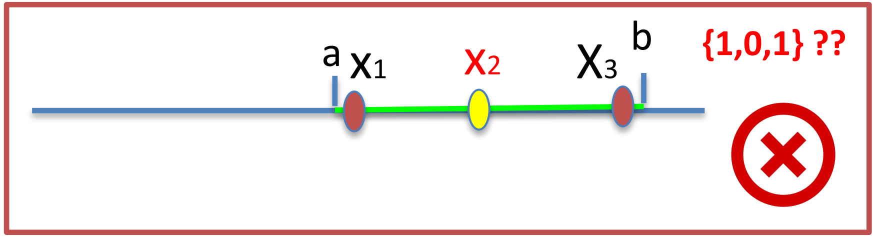
Example: interval function

Interval

$$\mathcal{H} = \{h_{a,b} : a, b \in \mathbb{R} \;, a < b\}$$

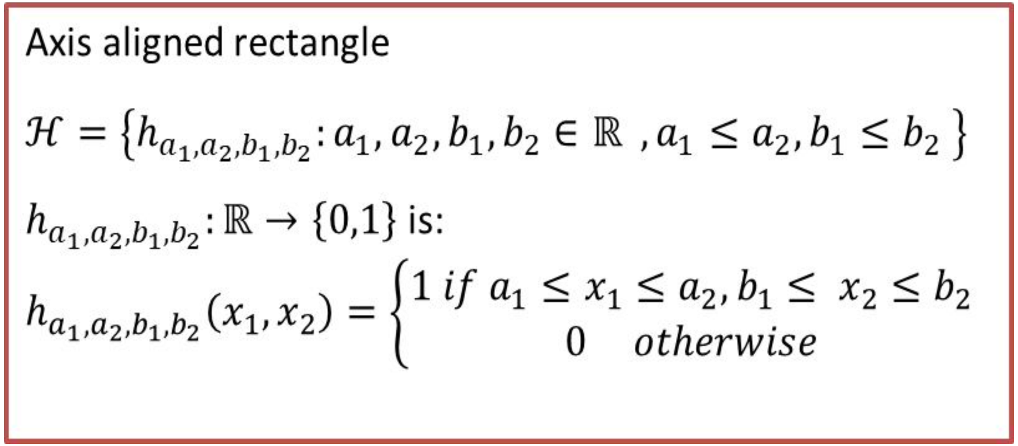
$h_{a,b} : \mathbb{R} \rightarrow \{0,1\}$ is:

$$h_{a,b}(x) = \begin{cases} 1 \; if \; a < x < b \\ 0 \quad otherwise \end{cases}$$



$$VC(H) = 2$$

Example: axis-aligned rectangle

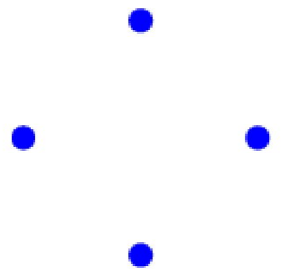


Axis aligned rectangle

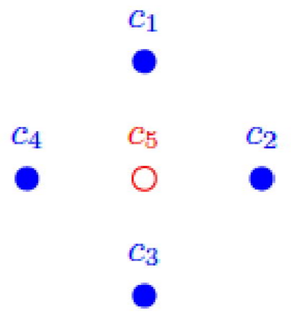
$$\mathcal{H} = \{h_{a_1,a_2,b_1,b_2} : a_1, a_2, b_1, b_2 \in \mathbb{R}\ , a_1 \leq a_2, b_1 \leq b_2\}$$

$h_{a_1,a_2,b_1,b_2} : \mathbb{R} \rightarrow \{0,1\}$ is:

$$h_{a_1,a_2,b_1,b_2}(x_1, x_2) = \begin{cases} 1\ if\ a_1 \leq x_1 \leq a_2, b_1 \leq\ x_2 \leq b_2 \\ \quad 0 \quad otherwise \end{cases}$$



The set can be shattered



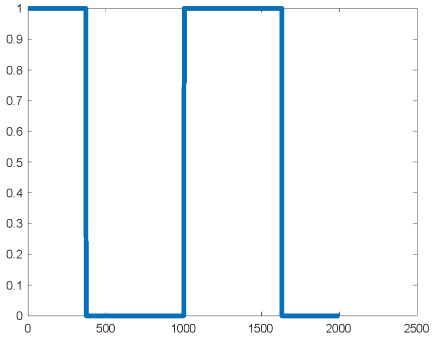
{1,1,1,1,0} No!!

$$VC(H) = 4$$

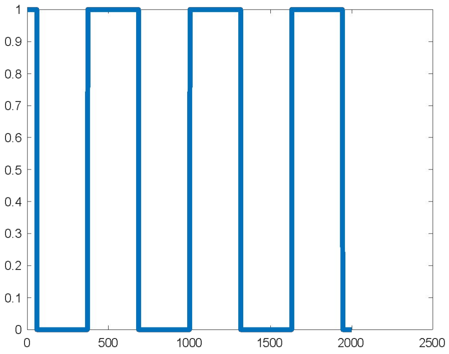
Finite parameters do not imply a finite VC dimension

Simple classes of functions can have an infinite VC dimension: a simple example is $H = \{h(x) = \lceil 0.5 \sin(\theta x) \rceil, \theta \in \mathbb{R}^+\}$

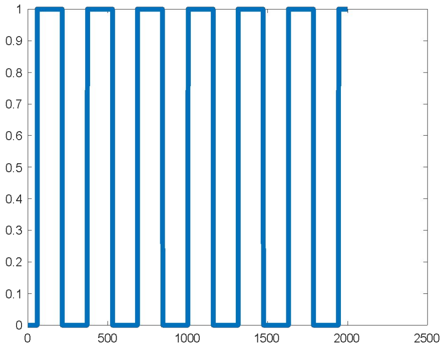
We can always construct a set that is shattered by it!



$\theta = 0.5$



$\theta = 1$



$\theta = 2$

Part 3:
Putting it all together

Università
degli Studi
di Padova

The fundamental theorem of PAC learning

Let H be a hypothesis class for a binary classification problem with 0-1 loss. The following statements are equivalent:

1. H has the uniform convergence property
2. Any ERM rule is a successful agnostic PAC learner for H
3. H is agnostic PAC learnable
4. H is PAC learnable (under realizability condition)
5. Any ERM rule is a successful PAC learner for H (under realizability condition)
6. H has a finite VC dimension

Proof: first step

We can easily prove that 1⇒2⇒3 from our results on uniform convergence:

1. H has the uniform convergence property
2. Any ERM rule is a successful agnostic PAC learner for H
3. H is agnostic PAC learnable

If a class H has the uniform convergence property with function m_H^{UC}, we have:

1. The class is agnostically PAC learnable with sample complexity

$$m_H(2\varepsilon, \delta) \leq m_H^{\mathrm{UC}}(\varepsilon, \delta)$$

2. ERM is a valid PAC learning algorithm for H

Proof: second step

3⇒4 is trivial, as agnostic PAC and PAC are the same if we add the realizability condition, and so is 2⇒5

2. Any ERM rule is a successful agnostic PAC learner for H
3. H is agnostic PAC learnable
4. H is PAC learnable (under realizability condition)
5. Any ERM rule is a successful PAC learner for H (under realizability condition)

We have 1⇒2⇒3⇒4 and 1⇒2⇒5

Proof: third step

4. H is PAC learnable (under realizability condition)
5. Any ERM rule is a successful PAC learner for H (under realizability condition)
6. H has a finite VC dimension

We now use the corollary of the no free lunch theorem: by contradiction, if H has an infinite VC dimension, it is **not** PAC learnable, as the theorem applies to any training set size. This proves that 4⇒5 and 4⇒6

We then have 1⇒2⇒3⇒4⇒5⇒6

Proof: final step (sketch)

6. H has a finite VC dimension
1. H has the uniform convergence property

The proof of this is very complex (the full proof is in the book). The main ideas are:

I. The “effective size” $|H_C|$ is $O(C^d)$ even if H is infinite: it grows polynomially rather than exponentially with respect to C (Sauer’s lemma)
II. We can generalize the claim that finite classes have uniform convergence to all classes that have a “small effective size,” i.e., grow polynomially.

A quantitative version

Let H be a hypothesis class for a binary classification problem with 0-1 loss. If its VC dimension is finite, there are absolute constants C_1, C_2 such that:

1. H has the uniform convergence property with sample complexity

$$C_1 \frac{d - \log(\delta)}{\varepsilon^2} \leq m_H^{\mathrm{UC}}(\varepsilon, \delta) \leq C_2 \frac{d - \log(\delta)}{\varepsilon^2}$$

2. H is agnostic PAC learnable with complexity

$$C_1 \frac{d - \log(\delta)}{\varepsilon^2} \leq m_H(\varepsilon, \delta) \leq C_2 \frac{d - \log(\delta)}{\varepsilon^2}$$