Machine learning

Some notes on Statistical Learning Theory

Corso di Laurea Magistrale in Informatica

Università di Roma Tor Vergata

Giorgio Gambosi

a.a. 2024–2025

Learning Algorithms and ERM

Learning Algorithm $\mathcal{A}$:

- Takes a dataset $\mathcal{T}$ with pairs from $\mathcal{X} \times \mathcal{Y}$
- Returns a predictor $A_{\mathcal{T}}$ computing a function $h_{\mathcal{T}} : \mathcal{X} \mapsto \mathcal{Y}$

Hypothesis Class $\mathcal{H}$:

- The search space for selecting $h_{\mathcal{T}}$
- Also known as the *Inductive bias*

Empirical Risk Minimization (ERM)

ERM Algorithm:

- Finds the predictor $h_{\mathcal{T}}$ minimizing the training error:

$$ERM(\mathcal{T}) = h_{\mathcal{T}} = \operatorname*{argmin}_{h} \overline{\mathcal{R}}_{\mathcal{T}}(h)$$

  where

$$\overline{\mathcal{R}}_{\mathcal{T}}(h) = \frac{1}{|\mathcal{T}|} \sum_{(x,t)\in\mathcal{T}} L(h(x), t) = 0$$

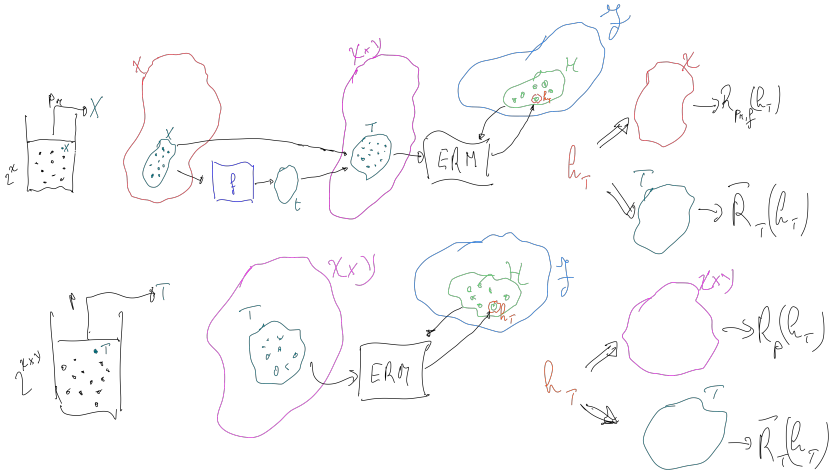
- Requires the specification of $\mathcal{H}$:

$$ERM(\mathcal{T}, \mathcal{H}) = h_{\mathcal{T},\mathcal{H}} = \operatorname*{argmin}_{h\in\mathcal{H}} \overline{\mathcal{R}}_{\mathcal{T}}(h)$$

Key Question in Learning Theory:

- Over which hypothesis classes will a learning algorithm (e.g., ERM) result in limited risk for various training sets?

Sketch of the situation



Finite Hypothesis Class $\mathcal{H}$, Realizability, and 0-1 loss

A bounded hypothesis class $\mathcal{H}$ ensures that overfitting does not occur if the dataset $\mathcal{T}$ is large enough.

- **Realizability Assumption**: There exists a predictor $h^* \in \mathcal{H}$ with no classification errors:

$$\mathcal{R}_{p_M,f}(h^*) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_M}[\, L(h^*(\mathbf{x}), f(\mathbf{x}))\,] = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_M}[\, |\mathbf{x} \in \mathcal{X} : h^*(\mathbf{x}) \neq f(\mathbf{x})|\,] = 0$$

- h^* correctly classifies all elements in $\mathcal{T}$:

$$\overline{\mathcal{R}}_{\mathcal{T}}(h^*) = \frac{1}{|\mathcal{T}|}\sum_{(x,t)\in\mathcal{T}} L(h^*(x), t) = \frac{|(\mathbf{x}, t) \in \mathcal{T} : h^*(\mathbf{x}) \neq t|}{|\mathcal{T}|} = 0$$

Empirical Risk Minimization (ERM) and Realizability

Under the realizability assumption, ERM returns an optimal predictor $h_{\mathcal{T}}$ on $\mathcal{T}$:

$$\overline{\mathcal{R}}_{\mathcal{T}}(h_{\mathcal{T}}) = 0$$

- ERM may return $h_{\mathcal{T}} = h^*$, which would be optimal for all elements in $\mathcal{X}$.
- However, it is possible that $h_{\mathcal{T}} \neq h^*$, meaning ERM performs optimally on $\mathcal{T}$ but may not generalize perfectly:

$$\mathcal{R}_{p_M,f}(h_{\mathcal{T}}) > 0$$

Definitions: Bad Predictors and Bad Sets

- A predictor $h \in \mathcal{H}$ is **bad** if it makes too many (expected) errors on $\mathcal{X}$:

$$\mathcal{R}_{p_M,f}(h) > \varepsilon$$

- A set $\mathcal{X} \subset \mathcal{X}$ is **bad** if applying ERM on it could result in selecting a bad predictor, that is if there exists a predictor $h_{\mathcal{T}}$ such that:

$$\overline{\mathcal{R}}_{\mathcal{T}}(h) = 0 \quad \text{but} \quad \mathcal{R}_{p_M,f}(h_{\mathcal{T}}) > \varepsilon$$

- If $h_{\mathcal{T}}$ is ideed the predictor returned by ERM, then $\mathcal{X}$ is **very bad**.

Studying Bad Sets and Dataset Size

We want to study how many examples are necessary to ensure that the probability of a bad dataset is small, for example less than a given $\delta \in (0, 1)$

$$\mathop{\mathbb{P}}_{\mathcal{T} \sim p^n} \left[\exists \tilde{h} \text{ bad} : \overline{\mathcal{R}}_{\mathcal{T}}(\tilde{h}) = 0 \right] \leq \delta$$

- This holds if:
$$\delta \geq |\mathcal{H}| e^{-\varepsilon n}$$
- Which implies:
$$n \geq \frac{1}{\varepsilon} \ln \frac{|\mathcal{H}|}{\delta}$$

That is, if n is greater than this bound, ERM returns with probability at least $1 - \delta$ a predictor with makes an expected fraction of errors smaller than ε.

Implications of Dataset Size n

- The probability of a bad dataset decreases as n increases.
- n must increase (logarithmically) if:
  - The size of $\mathcal{H}$ increases.
  - The definition of a bad predictor is made stricter (smaller ε).

PAC Learning

Probably Approximately Correct (PAC) Learning applies to binary classification problems with 0-1 loss as a measure of error.

- A hypothesis class $\mathcal{H}$ is PAC learnable if there exists a learning algorithm $\mathcal{A}$ that, with high probability, returns a predictor with low risk, if it may access enough training examples.
- that is, given $\varepsilon, \delta \in (0, 1)$, $\mathcal{A}$ returns a predictor with risk $R_{p_M, f}(h_{\mathcal{T}}) \leq \varepsilon$, with probability at least $1 - \delta$, given enough training examples.

PAC Learnability Definition

Definition (PAC Learnability)

A hypothesis class $\mathcal{H}$ is PAC learnable if there exists a function $m_{\mathcal{H}}(\varepsilon, \delta)$ and a learning algorithm $\mathcal{A}$ such that:

- For every distribution p_M over $\mathcal{X}$ and every function f, under the realizability assumption ($\mathcal{R}_{p_M,f}(h^*) = 0$),
- For a training set $\mathcal{T}$ of size $n \geq m_{\mathcal{H}}(\varepsilon, \delta)$,
- $\mathcal{A}$ returns a predictor $h_{\mathcal{T}}$ with probability at least $1 - \delta$ that has risk $R_{p_M,f}(h_{\mathcal{T}}) \leq \varepsilon$.

Accuracy and Confidence Parameters

- **Accuracy parameter** ε: Determines how close the output predictor is to the optimal one ("approximately correct").
- **Confidence parameter** δ: Indicates the likelihood that the predictor meets the accuracy requirement ("probably correct").

Sample Complexity in PAC Learning

The sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$ defines the minimum number of examples required to ensure that an approximately correct (with risk less than ε) predictor is probably (with probability greater than $1 - \delta$) selected.

- For finite $\mathcal{H}$, the sample complexity is upper bounded by the previously obtained value:

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq \left\lceil \frac{1}{\varepsilon} \ln \frac{|\mathcal{H}|}{\delta} \right\rceil$$

Extending PAC Learnability: Probabilistic Framework

In the probabilistic setting, target values t and inputs $\mathbf{x}$ are related by a conditional distribution $p_C(\mathbf{x}, t)$. The goal is to minimize the expected risk, that is finding the predictor h^* such that:

$$h^*(\mathbf{x}) = \operatorname*{argmin}_{y \in \mathcal{Y}} \mathop{\mathbb{E}}_{t \sim p_C(\cdot|\mathbf{x})} [\, L(y, t) \,] = \operatorname*{argmin}_{y \in \{0,1\}} p_C(t \neq y | \mathbf{x})$$

- h^* is called the Bayes predictor, h_{Bayes}
- since h_{Bayes} is optimal, for any learning algorithm $\mathcal{A}$ (including *ERM*) and for any training set $\mathcal{T}$, the risk of the predictor $h_{\mathcal{T}}$ returned by $\mathcal{A}$ when applied on $\mathcal{T}$ will be greater then (or equal at least) than the minimal possible risk, that of h_{Bayes}, that is $\mathcal{R}_p(h_{\mathcal{T}}) \geq \mathcal{R}_p(h_{\text{Bayes}})$
- however, h_{Bayes} requires knowledge of $p_C(t|\mathbf{x})$, which is unknown by hypothesis

Agnostic PAC Learning Definition

The No Free Lunch theorem (later on this) states that if no prior assumptions about $p(\mathbf{x}, t)$ is made, then there exists no learning algorithm that guarantees that, for any $\mathcal{T}$, the predictor $h_{\mathcal{T}}$ returned is as good as the bayesian one.

We may then require that the learning algorithm for most datasets returns a predictor $h_{\mathcal{T}}$ with risk greater, but not too much greater, than $\mathcal{R}_p(h^*)$, the risk of the best predictor $h^* \in \mathcal{H}$, whose risk is in general itself greater than h_{Bayes}. In doing this, we also generalize to the case when the realizability assumption does not hold (called agnostic)

Agnostic PAC Learning Definition

In the agnostic setting, the goal is to return a predictor with risk close to the best possible within $\mathcal{H}$:

Definition (Agnostic PAC Learnability)

A hypothesis class $\mathcal{H}$ is agnostic PAC learnable if for every $\varepsilon, \delta \in (0,1)$, there exists a function $m_{\mathcal{H}}(\varepsilon, \delta)$ and an algorithm that, given $n \geq m_{\mathcal{H}}(\varepsilon, \delta)$ training examples, returns a predictor h such that:

$$\mathcal{R}_p(h^*) \leq \mathcal{R}_p(h) \leq \mathcal{R}_p(h^*) + \varepsilon$$

with probability at least $1 - \delta$, where $\mathcal{R}_p(h) = \mathop{\mathbb{E}}_{(x,t)\sim p}[\,|h(x) \neq t|\,)\,]$ and h^* is the best predictor in $\mathcal{H}$.

Generalizing to General Loss Functions

Agnostic PAC Learnability can be extended to general loss functions:

Definition (Agnostic PAC Learnability for General Loss Functions)

A hypothesis class $\mathcal{H}$ is agnostic PAC learnable with respect to a loss function l if, for every $\varepsilon, \delta \in (0,1)$, the algorithm returns a predictor h such that:

$$\mathcal{R}_p(h^*) \leq \mathcal{R}_p(h) \leq \mathcal{R}_p(h^*) + \varepsilon$$

with probability at least $1-\delta$, where $\mathcal{R}_p(h) = \underset{(x,t)\sim p}{\mathbb{E}} [\, |h(x) \neq t|) \,]$ and h^* is the best predictor in $\mathcal{H}$.

Empirical Risk, True Risk, and representative sets

ERM selects a predictor $h_{\mathcal{T}}$ that minimizes the empirical risk $\overline{\mathcal{R}}_{\mathcal{T}}(h)$ on the training set $\mathcal{T}$. It should closely approximate the true risk across the entire hypothesis class for ERM to be effective. This is a property of $\mathcal{T}$:

Definition (ε-representative sample)

A training set $\mathcal{T}$ is ε-representative if:

$$\forall h \in \mathcal{H}, |\overline{\mathcal{R}}_{\mathcal{T}}(h) - \mathcal{R}_p(h)| \leq \varepsilon$$

ERM and Approximation Quality

If $\mathcal{T}$ is $\frac{\varepsilon}{2}$-representative, the predictor returned by ERM satisfies:

$$\mathcal{R}_p(h_{\mathcal{T}}) \leq \mathcal{R}_p(h^*) + \varepsilon$$

This guarantees that the ERM predictor is close to the best predictor in $\mathcal{H}$, with only a small error margin.

Ensuring ERM's Effectiveness: Uniform Convergence

Definition (Uniform Convergence)

A hypothesis class $\mathcal{H}$ has the uniform convergence property if there exists a function $m^{UC}_{\mathcal{H}}(\varepsilon, \delta)$ such that for all $\varepsilon, \delta \in (0, 1)$, and any distribution $p(\mathbf{x}, t)$, a training set $\mathcal{T}$ of size $n \geq m^{UC}_{\mathcal{H}}(\varepsilon, \delta)$ is ε-representative with probability $1 - \delta$.

Sample Complexity for Uniform Convergence

The sample complexity $m_{\mathcal{H}}^{UC}(\varepsilon, \delta)$ for finite hypothesis classes is given by:

$$m_{\mathcal{H}}^{UC}(\varepsilon, \delta) \leq \left\lceil \frac{1}{2\varepsilon^2} \ln \frac{2|\mathcal{H}|}{\delta} \right\rceil$$

Thus, $\mathcal{H}$ is PAC learnable using the ERM algorithm with sample complexity:

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq \left\lceil \frac{1}{\varepsilon^2} \ln \frac{2|\mathcal{H}|}{\delta} \right\rceil$$

Finite vs. Infinite Classes

- Finite hypothesis classes are PAC learnable via ERM with logarithmic sample complexity.
- For infinite hypothesis classes, discretization can give a rough sample complexity estimate.

Generalizing to Infinite Hypothesis Classes

For a hypothesis class parameterized by d real-valued parameters, the effective size in practice is constrained by floating-point precision:

$$|\mathcal{H}| \approx 2^{64d}$$

Thus, the sample complexity is approximately:

$$\frac{128d + 2\ln\frac{2}{\delta}}{\varepsilon^2}$$

What about if we do not rely on discretization?

Inductive Bias and Hypothesis Class

- Choosing a hypothesis class $\mathcal{H}$ incorporates prior knowledge about the data.
- This prior knowledge reflects the belief that $\mathcal{H}$ contains a low-risk predictor.

A universal learner would find a low-risk hypothesis for any distribution p.

No-Free-Lunch Theorem

No universal learner exists.

Theorem (No-Free-Lunch)

Let $\mathcal{A}$ be a learning algorithm over domain $\mathcal{X}$, and $n < \frac{|\mathcal{X}|}{2}$. There exists a distribution $\overline{p}_{\mathcal{A}}$ such that:

1. *There exists a predictor $h^* : \mathcal{X} \mapsto \{0, 1\}$ with $R_{\overline{p}_{\mathcal{A}}}(h^*) = 0$ (that is the realizability assumption holds on $\mathcal{X} \mapsto \{0, 1\}$ if pairs are distributed according to $\overline{p}_{\mathcal{A}}$).*
2. *With probability at least $1/7$ over the choice of a dataset $\mathcal{T}$ of size n of i.i.d. pairs, each sampled according to $\overline{p}_{\mathcal{A}}$, we have that $R_{\overline{p}_{\mathcal{A}}}(h_{\mathcal{A},\mathcal{T}}) \geq 1/8$, where $h_{\mathcal{A},\mathcal{T}}$ is the predictor returned by $\mathcal{A}$ when applied on $\mathcal{T}$.*

Implications of No-Free-Lunch

- For every learner, there exists a task (a distribution on $\mathcal{X} \times \mathcal{Y}$) on which it fails, even though that task can be successfully learned by another learner.
- Let us consider the hypothesis class $\mathcal{F}$ of all the functions *f* from an infinite-size $\mathcal{X}$ to $\{0, 1\}$. This class represents lack of prior knowledge: every possible function from $\mathcal{X}$ to $\mathcal{Y} = \{0, 1\}$ is considered. According to the No Free Lunch theorem, any learning algorithm that chooses a predictor from hypotheses in $\mathcal{F}$, and in particular the *ERM* algorithm, will fail on some learning task. Therefore, the absence of prior knowledge results in the class $\mathcal{F}$ that is not PAC learnable.
- If we do not restrict ourselves to a subset of all functions from $\mathcal{X}$ to $\{0, 1\}$ (i.e. choose a hypothesis space), there will always be a probability distribution $\overline{p}$ that makes any learning algorithm return a "bad" predictor with high probability, even though there exists one with zero error. This implies that no algorithm will be able to PAC-learn this target function.
- Choosing a suitable hypothesis class is crucial for learning a given function. This way we restrict ourselves to a subset of all possible functions from $\mathcal{X}$ $\{0, 1\}$, which helps us avoiding unfavourable distributions and might allow us to find a low-error hypothesis with high probability.

Bias-Complexity Tradeoff

- The chosen hypothesis class might exclude the best possible predictor.
- But we could find an approximation in the hypothesis class.
- However, this best approximation might be a poor predictor for the true target.
- This tradeoff is referred to as the Bias-Complexity Tradeoff.

Risk Decomposition

$$\mathcal{R}_p(h_{\mathcal{T}}) - \mathcal{R}_p(h_{\text{Bayes}}) = \underbrace{(\mathcal{R}_p(h_{\mathcal{T}}) - \mathcal{R}_p(h^*)}_{\text{estimation error}} + \underbrace{(\mathcal{R}_p(h^*) - \mathcal{R}_p(h_{\text{Bayes}}))}_{\text{approximation error}} = \varepsilon_V + \varepsilon_B$$

- h^*: Best predictor in $\mathcal{H}$
- h_{Bayes}: Absolute best predictor for the task

Approximation error

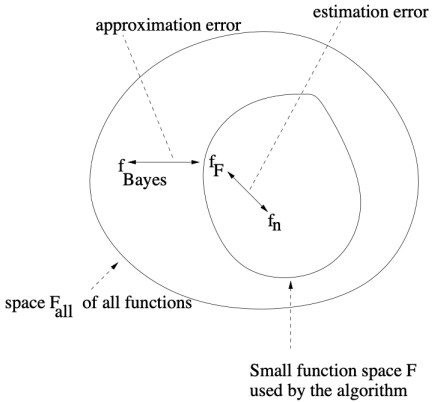
- ϵ_B: it is a function of the minimum risk achievable by any $h \in \mathcal{H}$.
- It is a property of the hypothesis class $\mathcal{H}$ with respect to the prediction task.
- It is independent from the training set.
- This is referred to as bias.

Estimation error

- ϵ_V: it is the difference between the minimum risk achievable in $\mathcal{H}$ and the risk of the best predictor in $\mathcal{H}$ obtained by considering the training set.
- Related to how well ERM estimates the best predictor based on the given training set.
- Reflects how much a predictor from a random training set may perform worse than the best possible predictor.
- Its expectation with respect to all possible training sets is a measure of how much a predictor derived from a random training set may result in poorer performances with respect to the best possible one. This is called variance

Bias-Variance Tradeoff in Hypothesis Class $\mathcal{H}$

- The choice of hypothesis class $\mathcal{H}$ is subject to a bias-variance tradeoff.
- Higher bias tends to induce lower variance, and vice versa.



Estimation and approximation error illustration.

High Bias and Low Variance: Underfitting

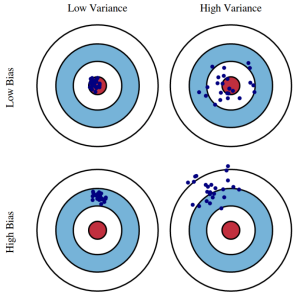
- Predictors from different training sets behave similarly with low variance.
- All predictors perform poorly (high bias), as $\mathcal{H}$ is too poor for the task.
- This results in underfitting.

Low Bias and High Variance: Overfitting

- $\mathcal{H}$ contains many predictors, including a good one (low bias).
- Predictors can vary significantly across training sets (high variance).
- While a good performance may be achieved on the training set, the predictor might behave poorly on new data, leading to overfitting.

Large Hypothesis Space and Overfitting

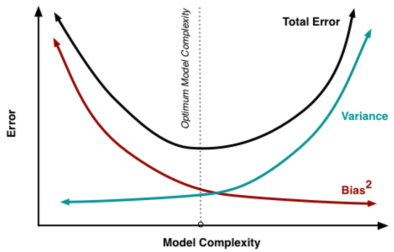
- A large $\mathcal{H}$ may contain complex functions, making the approximation error small.
- The Bayes classifier might even be contained in $\mathcal{H}$ or closely approximated.
- However, the estimation error increases, leading to overfitting.



Bias and variance illustration.

Small Hypothesis Space and Underfitting

- A small hypothesis class $\mathcal{H}$ results in a large approximation error.
- However, the estimation error is small, leading to underfitting.



Bias and variance vs model complexity.

Learning Theory: Balancing $\mathcal{H}$

- Learning theory studies how rich we can make $\mathcal{H}$ while maintaining a reasonable estimation error.
- Good predictor classes should have low approximation error and moderate estimation error.
- Practical approaches focus on balancing bias and variance.

Model Selection

- In practice, predictors are defined by specific hyper-parameters and types.
- The process of selecting the right type of predictor and hyper-parameters is called model selection.
- Learning algorithms like ERM help select the best predictor from the defined class.

Hypothesis Classes and set shattering

Finiteness is sufficient but not necessary for learnability. We wish to define a more general and useful measure of complexity,

Given a subset $C = \{c_1, ..., c_m\} \subset \mathcal{X}$ of $\mathcal{X}$, we define the *restriction* of $\mathcal{H}$ to C as the set of functions $f : C \mapsto \{0, 1\}$ that can be derived from predictors in $\mathcal{H}$ (i.e., such that for each $f \in C$ there exists a predictor $h \in \mathcal{H}$ for which $f(c_i) = h(c_i), i = 1, \dots, m$). If we describe each function from C to $\{0, 1\}$ as a vector in $\{0, 1\}^{|C|}$, we can formally write it as

$$\mathcal{H}_C = \{(h(c_1), ...h(c_m)) : h \in \mathcal{H}\}.$$

This means that for every binary labeling of the elements of C (and thus for every possible binary classification task on C), there exists a predictor in $\mathcal{H}$ that separates the two classes, in the sense that it correctly predicts the target values of each element c_i. In this case, we say that $\mathcal{H}$ *shatters* C.

The Vapnik-Červonenkis Dimension

The VC-Dimension VCdim$(\mathcal{H})$ of a class $\mathcal{H}$ is the size of the largest subset of $\mathcal{X}$ which is shattered by $\mathcal{H}$.

From the No-Free-Lunch theorem, we know that the set of all functions from a domain to $\{0, 1\}$ is not PAC-learnable. However, the proof of this statement is based on the assumption that we are considering all possible functions: it is reasonable to assume that introducing limitations on the hypothesis class might bring advantages

VC-Dimension makes it possible to characterize “good” limitations (at least in a theoretical framework)

Example: Threshold Functions $\mathcal{H}^{\text{THR}}$

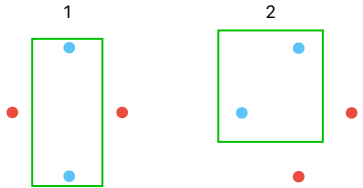
Threshold function with threshold θ:

$$\mathcal{H}_\theta = \{\mathbb{1}[x < \theta]; \theta \in \mathbb{R}\}.$$

- VCdim$(\mathcal{H}^{\text{thr}}) = 1$
- For 1 point set, $C = \{c_1\}$ can be shattered by $\theta = c_1 + 1$ which implies $h_\theta(c_1) = 1$, or $\theta = c_1 - 1$, which results into $h_\theta(c_1) = 0$
- For 2 point set, $C = \{c_1, c_2\}$ with $c_1 > c_2$ with labeling $c_1 = 1, c_2 = 0$ cannot be shattered

Example: Axis-Aligned Rectangles $\mathcal{H}^{\text{RECT}}$

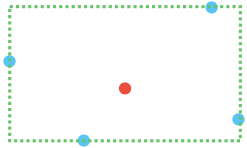
- VCdim$(\mathcal{H}^{\text{rect}}) = 4$: 4 points can be shattered.



Shattering a set of 4 points with axis-aligned rectangles.

Example: Axis-Aligned Rectangles $\mathcal{H}^{\text{RECT}}$

- For any set of 5 points, there is always one point inside the bounding box, so 5 points cannot be shattered.



The impossibility of shattering a set of 5 elements using axis-aligned rectangles.

Example: Intervals on $\mathbb{R}$ $\mathcal{H}^{\text{INT}}$

- VCdim$(\mathcal{H}^{\text{int}}) = 2$: Only sets of 2 points can be shattered.
- For $C = \{c_1, c_2, c_3\}$, the labeling $(1, 0, 1)$ cannot be obtained.

Shattering a 2-element set using intervals.

Finite Hypothesis Classes $\mathcal{H}^{\text{FIN}}$

- In general, in order to shatter a set C we need $2^{|C|}$ predictors.
- For a finite class $\mathcal{H}^{\text{fin}}$, $|\mathcal{H}^{\text{fin}}_C| \leq |\mathcal{H}^{\text{fin}}|$
- C cannot be shattered by $\mathcal{H}^{\text{fin}}$ if $|\mathcal{H}^{\text{fin}}| < 2^{|C|}$
- Then, $\text{VCdim}(\mathcal{H}^{\text{fin}}) \leq \log_2|\mathcal{H}|$

The PAC learnability of finite classes then derives from the more general property PAC learnability of classes with finite VC-dimension.

Finite Hypothesis Classes $\mathcal{H}^{\text{FIN}}$

However, note that the VC-dimension of a finite class $\mathcal{H}^{\text{fin}}$ can be significantly smaller than $\log_2(|\mathcal{H}^{\text{fin}}|)$. For example, let $\mathcal{X} = \{1, \dots, k\}$ for some integer k, and consider the class of threshold functions on $\mathcal{H}$. Then, $|\mathcal{H}| = k$ but $\text{VCdim}(\mathcal{H}) = 1$. Since k can be arbitrarily large, the difference between $\log_2(|\mathcal{H}|)$ and $\text{VCdim}(\mathcal{H})$ can be arbitrarily large.

Fundamental Theorem of Statistical Learning

Let $\mathcal{H}$ be a class of hypotheses $h : \mathcal{X} \to \{0, 1\}$ for binary classification, and let the $0 - 1$ loss be the considered cost function. Then, the following statements are equivalent:

1. $\mathcal{H}$ has a finite VC-dimension.
2. $\mathcal{H}$ is agnostic PAC-learnable, and there exist constants $c_1 < c_2$ such that its sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$ is upper and lower bounded as

$$\frac{c_1}{\varepsilon^2}\left(d + \ln\frac{1}{\delta}\right) \leq m_{\mathcal{H}}(\varepsilon, \delta) \leq \frac{c_2}{\varepsilon^2}\left(d + \ln\frac{1}{\delta}\right)$$

   Moreover, this property holds also when ERM is applied (that is, it is a successful agnostic PAC-learning algorithm for $\mathcal{H}$).
3. $\mathcal{H}$ is PAC-learnable, and its sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$ is upper and lower bounded as

$$\frac{c_1}{\varepsilon}\left(d + \ln\frac{1}{\delta}\right) \leq m_{\mathcal{H}}(\varepsilon, \delta) \leq \frac{c_2}{\varepsilon}\left(d + \ln\frac{1}{\delta}\right)$$

   Moreover, this property holds also when ERM is applied (that is, it is a successful PAC-learner for $\mathcal{H}$).