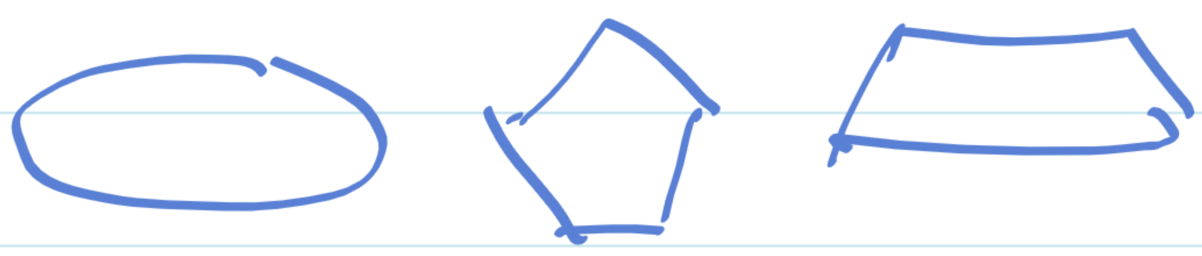
1st June 2021 — Lecture - 6 — VC - Dimension

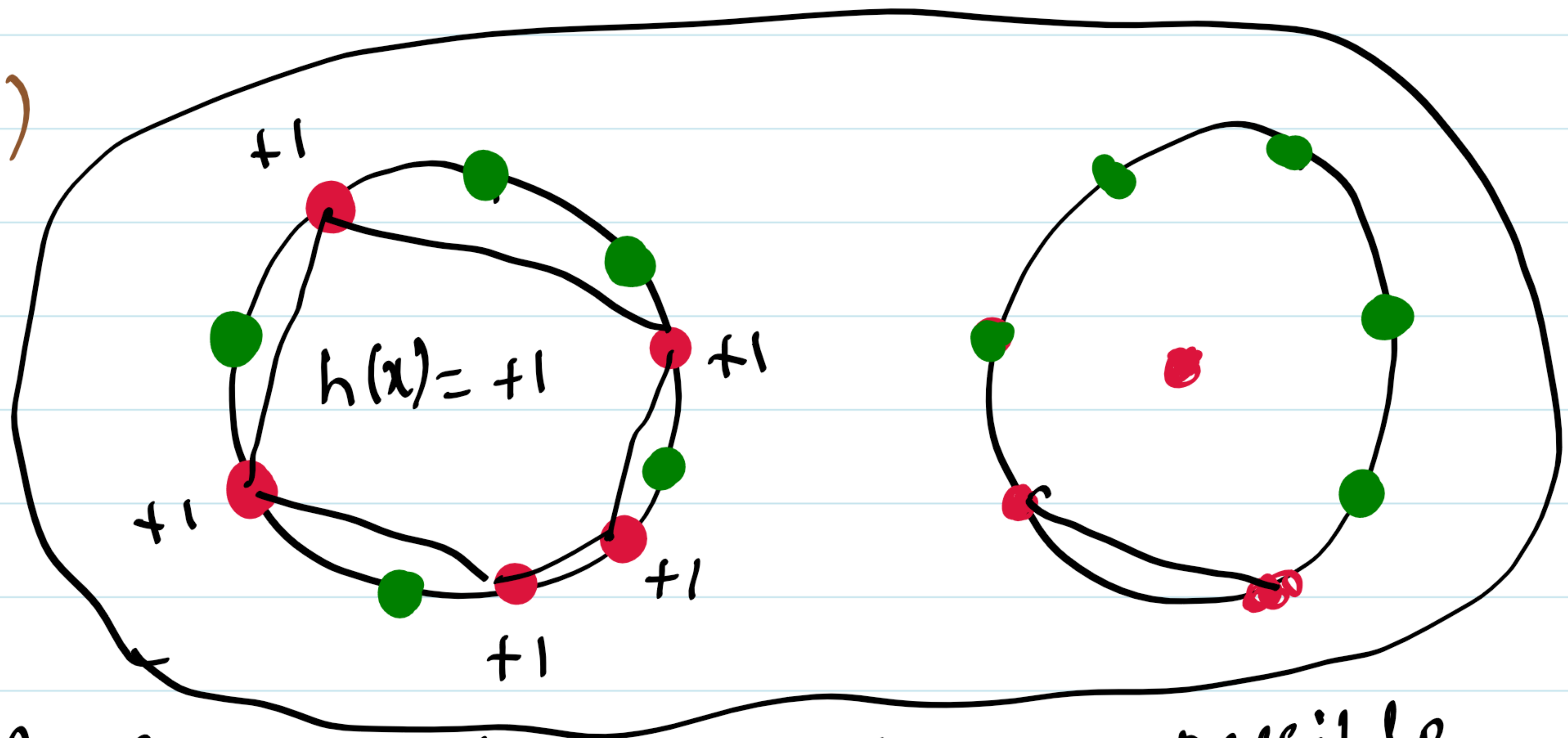
Example of calculating growth function

Convex sets : $\mathcal{H}$ = Set of $h : \mathbb{R}^2 \longrightarrow \{+1, -1\}$.

$h(x) = +1$ inside some convex set & -1 else where



To compute $m_{\mathcal{H}}(N)$ in this case one needs to choose the N points on the perimeter of the circle.



The growth fun^c has the maximum possible values. $m_{\mathcal{H}}(N) = 2^N$.

Summary of examples :

- → Positive ray : $m_{\mathcal{H}}(N) = N+1$
- → Positive interval : $m_{\mathcal{H}}(N) = \binom{N+1}{2} + 1$
- → If $\mathcal{H}$ is convex : $m_{\mathcal{H}}(N) = 2^N$.
- → Replace M by $m_{\mathcal{H}}(N)$ (Polynomial)

dichotomies ⟹ Shattering ⟹ VC dimension

Shattering : If a hypothesis set $\mathcal{H}$ is able to generate 2^N dichotomies, then we say that $\mathcal{H}$ shatter $x_1, \dots, x_N$.

Example : $\mathcal{H}$ = hyperplane returned by a binary classifier in 2D.

→ If $N = 3$, can $\mathcal{H}$ shatter? Yes
$2^3 = 8$ dichotomies.

→ If $N = 4$, can a linear classifier shatter?
$2^4 = 16$. We can only achieve 14.

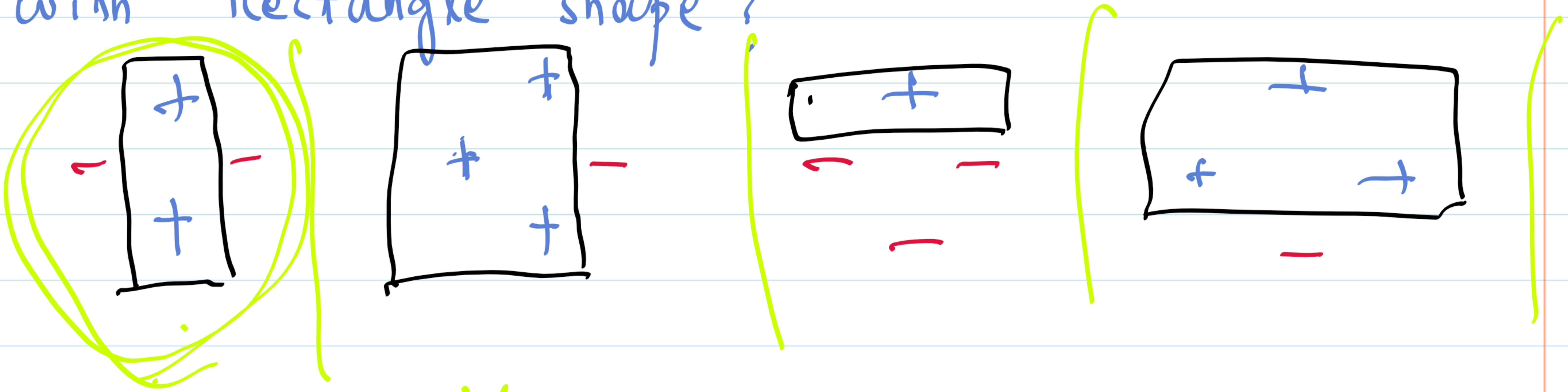
VC Dimension :

The Vapnik-Chervonenkis dimension of a hypothesis set $\mathcal{H}$, denoted by d_{VC}, is the largest value of N for which $\mathcal{H}$ can shatter all N training samples, i.e., $m_{\mathcal{H}}(N) = 2^N$.

Explanation : Give me a hypothesis set $\mathcal{H}$ (i.e. (a linear model))

→ Tell me the no. of training sample.
→ I will be able to shatter for a while until I hit a bump.
→ E.g. Linear model in 2D, $N=3$ (okay), $N=4$ (Not okay), $d_{VC} = 3$.
→ Good thing : Does not depend on $p(x)$, $\mathcal{A}$ and f.

Q:- What is the VC dimension of a 2D classifier with rectangle shape?



$2^4 = 16$ configuration

→ $d_{VC} = 4$ for a rectangular classifier in 2D.

Theorem :- (VC dimension of a perceptron)

Consider the input space $X = \mathbb{R}^d \cup \{1\}$, i.e., $(x = [1, x_1, \ldots, x_d]^T)$. Then the VC dimension of a perceptron is $d_{VC} = d+1$

Perceptron : It is an algorithm for learning a binary classifier called a threshold function : a func that maps its input x (a real valued vector) to an output value $f(x)$ (a single binary value) :

$$f(x) = \begin{cases} +1 & \text{if } w^T x + b > 0 \\ -1 & \text{otherwise}, \end{cases}$$

w is a vector of real value weights,

$$w \cdot x = \sum_{i=1}^{m} w_i x_i \,. \quad m = \text{no. of inputs}$$

and b is the bias term.

→ $+1$ comes from the bias term.

→ So the linear classifier is 'no more complicated' than $d+1$

→ The best it can shatter is $d+1$ in a d-dim. space. $d=2$, $d_{VC} = 3$.

Proof : $d_{VC} \geq d+1$, $d_{VC} \leq d+1$.

$d_{VC} \geq d+1$ ↓ It can shatter at least $d+1$ points

$d_{VC} \leq d+1$ ↓ It cannot shatter more than $d+1$ points

Case-1 ($d_{vc} \geq d+1$)

→ To show that there is at least one configuration of d+1 pts. that can be shattered by $\mathcal{H}$.

(For example, think about 2D case: Put 3 pts. anywhere, but not on the same line)

→ In high dimensions, choose

$$x_n = [1, 0, \ldots, 1, \ldots, 0]^T$$

(first entry 1: bias; the other 1: somewhere in the middle)

→ Linear classifier: $\text{Sign}(w^T x_n) = y_n$ (label)

→ For all d+1 pts.,

$$\text{Sign}\left(\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 0 & 0 & \cdots 0 & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}\right) = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{d+1} \end{bmatrix} = \begin{bmatrix} \pm 1 \\ \pm 1 \\ \vdots \\ \pm 1 \end{bmatrix}.$$

(matrix: as orthogonal as possible) (right-hand vector: different dichotomies)

(It's just one configuration, think of this as locations)

Aim: Try to find one configuration of pts. that can be shattered. So we are only interested in whether the problem is solvable, i.e., we need to see if we can find a w that shatters.

→ Is $(d+1)\times(d+1)$ system invertible?
Yes, It can shatter at least $d+1$ points.

Case-2 ($d_{vc} \le d+1$)

Can we shatter more than $d+1$ points?
No, you have only $d+1$ variables.
If you have $d+2$ equations, then one eq^n will be either redundant or contradictory
⇓ ignore it
⇓ then you can not solve

→ You give me $x_1, \dots, x_{d+1}, x_{d+2}$.
→ I can write x_{d+2} as

$$x_{d+2} = \sum_{i=1}^{d+1} a_i x_i \,, \quad \text{not all } a_i = 0.$$

→ My job is to construct a dichotomy which can not be shattered by any h.

→ $x_1, \dots, x_{d+1}$ get $y_i = \text{Sign}(a_i)$
→ x_{d+2} get $y_{d+2} = -1$

$$w^T x_{d+2} = \sum_{i=1}^{d+1} a_i (w^T x_i)$$

→ Perceptron: $y_i = \text{Sign}(w^T x_i)$
→ By our design, $y_i = \text{sign}(a_i)$.

$$\sum_{i=1}^{d+1} a_i w^T x_i > 0$$

$$y_{d+2} = \text{Sign}(w^T x_{d+2}) = +1 \Leftarrow \;\Rightarrow\!\Leftarrow$$

$d_{vc} \le d+1$

Summary of examples : (VC dimension)

→ $\mathcal{H}$ is positive ray : $m_{\mathcal{H}}(N) = N+1$

If $N=1$, $m_{\mathcal{H}}(1) = 2 = 2^1 = 2$

$N=2$, $m_{\mathcal{H}}(2) = 3 \neq 2^2 = 4$

$d_{vc} = 1$.

→ $\mathcal{H}$ is a positive interval : $m_{\mathcal{H}}(N) = \frac{N^2}{2} + \frac{N}{2} + 1$

$N=2$, $m_{\mathcal{H}}(2) = 4$

$N=4$, then $m_{\mathcal{H}}(4) = 5$

$d_{vc} = 2$.

→ $\mathcal{H}$ is a perceptron in d-dimensional space

$d_{vc} = d+1$.

→ $\mathcal{H}$ is a convex set. $m_{\mathcal{H}}(N) = 2^N$.

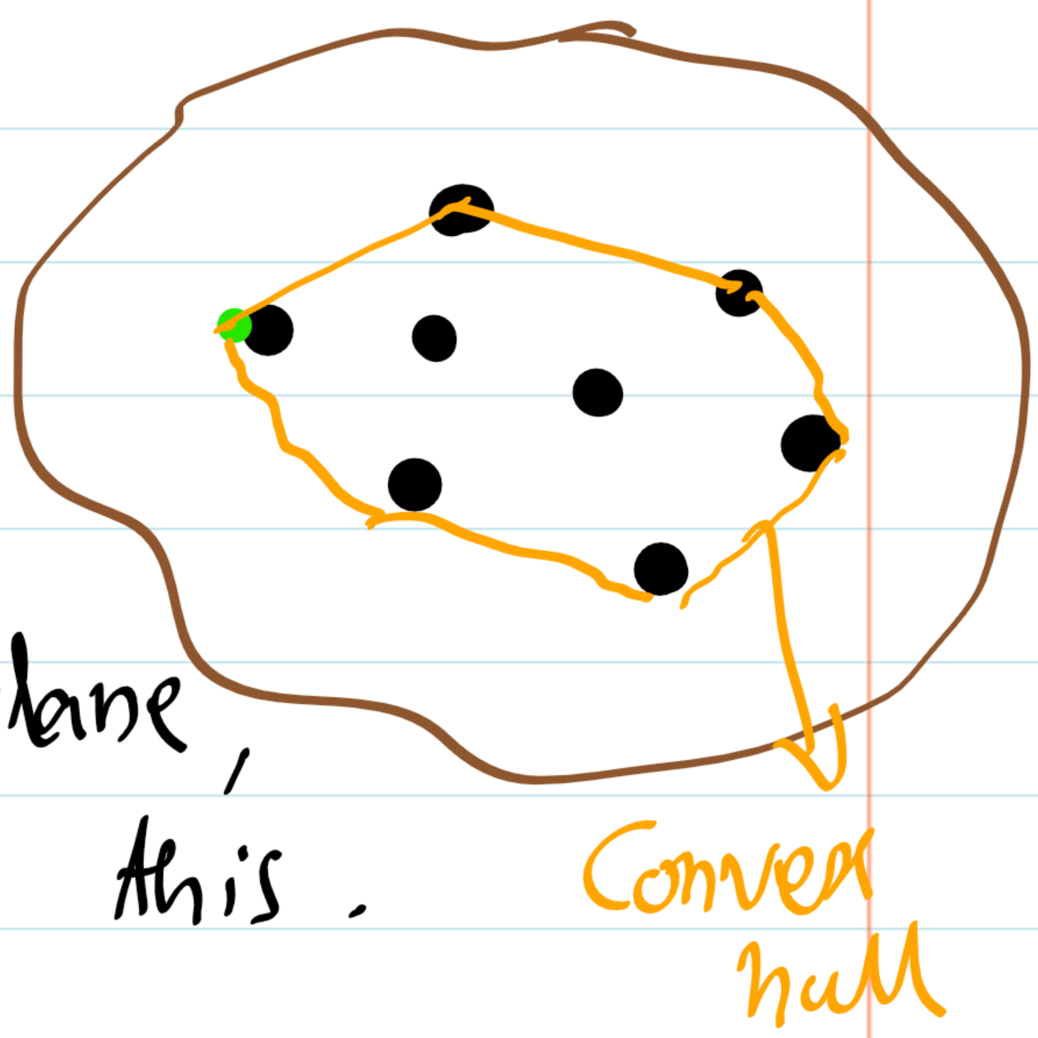
So no matter what N we choose, we will always get $m_{\mathcal{H}}(N) = 2^N$. So $d_{vc} = \infty$.

Perceptron VC dimension :

Radon's theorem : Any set X of $d+2$ points in $\mathbb{R}^d$ can be partitioned into two subsets X_1 and X_2 s.t. the convex hulls X_1 and X_2 intersect.

Let X be a set of $d+2$ pts. By Radon's theorem, ...

Observe that when 2 sets of points X_1 and X_2 are separated by a hyperplane, their convex hulls also separated by this.



Link between VC dimension and growth function

Theorem (Sauer's lemma)

Let d_{vc} be the VC dimension of a hypothesis set $\mathcal{H}$, then

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{d_{vc}} \binom{N}{i}.$$

→ An interesting bound

$$\sum_{i=0}^{d_{vc}} \binom{N}{i} \leq N^{d_{vc}} + 1$$

$$\boxed{m_{\mathcal{H}}(N) \leq N^{d_{vc}} + 1}$$

→ Recall the generalisation bound.

$$E_{in}(g) - \sqrt{\frac{1}{2N} \log \frac{2M}{\delta}} \leq E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{1}{2N} \log \frac{2M}{\delta}}$$

→ We replace M by $m_{\mathcal{H}}(N)$ and then

$$m_{\mathcal{H}}(N) \leq N^{d_{vc}} + 1.$$

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{1}{2N} \log \frac{2(N^{d_{vc}}+1)}{\delta}}$$

($N^{d_{vc}}+1$:) much smaller than M

→ Everything is characterised by δ, N and d_{vc}.

(d_{vc}:) expressiveness of our model.

Some properties :

→ If $d_{vc} < \infty$, Then as $N \to \infty$, the accuracy

$$\epsilon = \sqrt{\frac{1}{2N} \log \frac{2(N^{d_{vc}}+1)}{\delta}} \longrightarrow 0$$

→ If $d_{vc} = \infty$, Then $\mathcal{H}$ is as diverse as it can be. we will not be able to generalise.

Message 1 : If you choose a complex model, then you need to pay the price of training sample.

Message 2 : If you have an extremely complex model, then it may not be able to generalise regardless the no. of samples.

VC generalisation bound :

For any tolerance $\delta > 0$

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} \log \frac{4\, m_{\mathcal{H}}(2N)}{\delta}}$$

with probability at least $1-\delta$.

Sample complexity VS Model complexity :

Sample complexity : what is the smallest no. of samples required ?

→ Required to ensure that the training and testing error are close. = with in certain ϵ with confidence $1-\delta$.

Model complexity : what is the largest model that you can use? ↓ In terms of VC dim.

→ Refers to hypothesis set, w.r.t. no. of training samples.
→ Regardless of algorithm.

Sample complexity :

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} \log \frac{4 m_{\mathcal{H}}(2N)}{\delta}}$$

If we want the generalisation error to be at most ε,

$$\sqrt{\frac{8}{N} \log \frac{4 m_{\mathcal{H}}(2N)}{\delta}} \leq \varepsilon$$

Using the VC dimension,

$$N \geq \frac{8}{\varepsilon^2} \log \left(\frac{4((2N)^{d_{vc}} + 1)}{\delta} \right)$$

Example : $d_{vc} = 3$, $\varepsilon = 0.1$, $\delta = 0.1$ (90% confidence)

$$N \geq \frac{8}{(0.1)^2} \log \left(\frac{4 (2N)^3 + 4}{0.1} \right)$$

Iteratively, $N = 1000$ in RHS

$N = 1000$

$$N \geq \frac{8}{(0.1)^2} \log (\quad) \approx 21,200$$

Then there is a mismatch.

Choose again $N = 21,200$ in above in RHS.

$$N \geq \left(. \quad . \lg ... \right) \approx 30,000.$$

It is over-estimate.

Error bar: