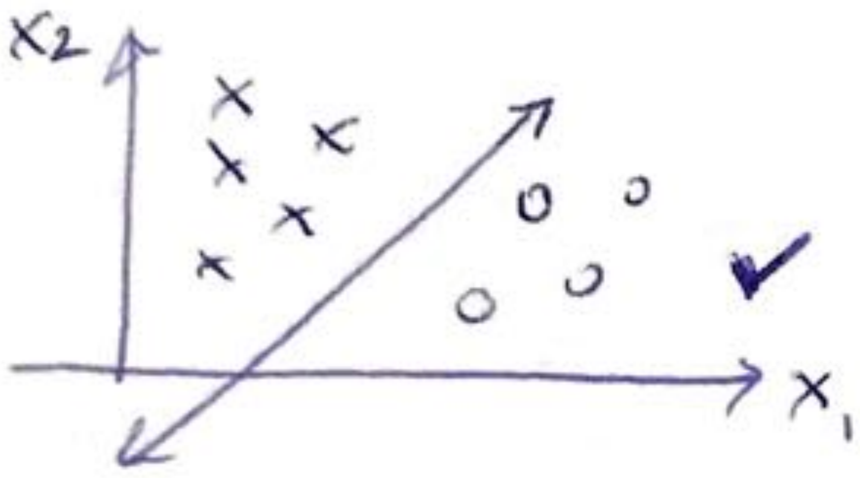
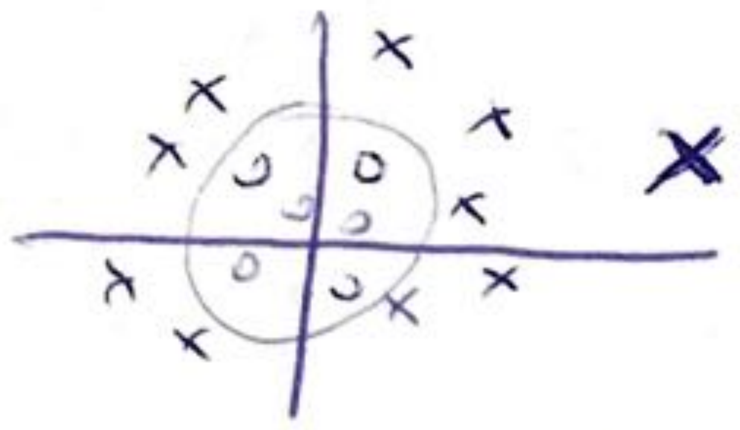
Chapter 8: Pattern Recognition

Perceptron Algorithm:

↳ used only for linearly seperable ~~algorithms~~ datasets:





Parameters ① (number of dim + 1)
↓
coefficients

$$w = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix}$$

② Learning rate (n): how fast the perceptron algorithm converges to this line (that seperates the classes)

$\uparrow n \Rightarrow$ faster convergence but might overshoot often

$\downarrow n \Rightarrow$ slower -//- less mistakes

Inputs: $x = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$
↑
number of dim + 1

$$w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 \quad \begin{cases} > 0 \Rightarrow x \\ \leq 0 \Rightarrow 0 \end{cases}$$

$$w^T \cdot x$$

update parameters:

let $w \begin{pmatrix} 0 \\ 1 \\ 0.5 \end{pmatrix}$ $x = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$

$w^T \cdot x = 0 \cdot 1 + 1 \cdot x_1 + 0{,}5 \cdot x_2 > 0 \iff x_2 > -2x_1$ → line on the plane

anything above the line will be x and below 0 ~~cl~~ categorised

update step: (if misclassification)

$$w_i' = w_i + n d x_i$$

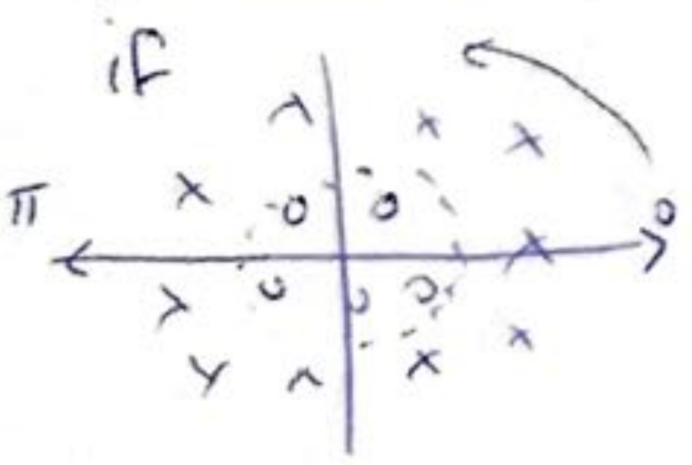
$$d = \begin{cases} 1, & \text{if should be in upper part the misclassified} \\ -1, & \text{if should be in lower part} \end{cases}$$

let $n = 0.2$ $\quad x = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ -2 \end{pmatrix}$

$$\left.\begin{aligned} w_0' &= w_0 + n d x_0 = 0 + (0.2) \cdot (-1) \cdot (1) \xrightarrow{\text{your } x} = 0.2 \\ w_1' &= w_1 + n d x_1 = 1 + (0.2) \cdot (-1) \cdot (2) = 0.6 \\ w_2' &= w_2 + n d x_2 = 0.5 + (0.2) \cdot (-1) \cdot (2) = 0.9 \end{aligned}\right\} \text{new line: } x_2 > -\frac{2}{3} x_1 + \frac{2}{9}$$

Extension:

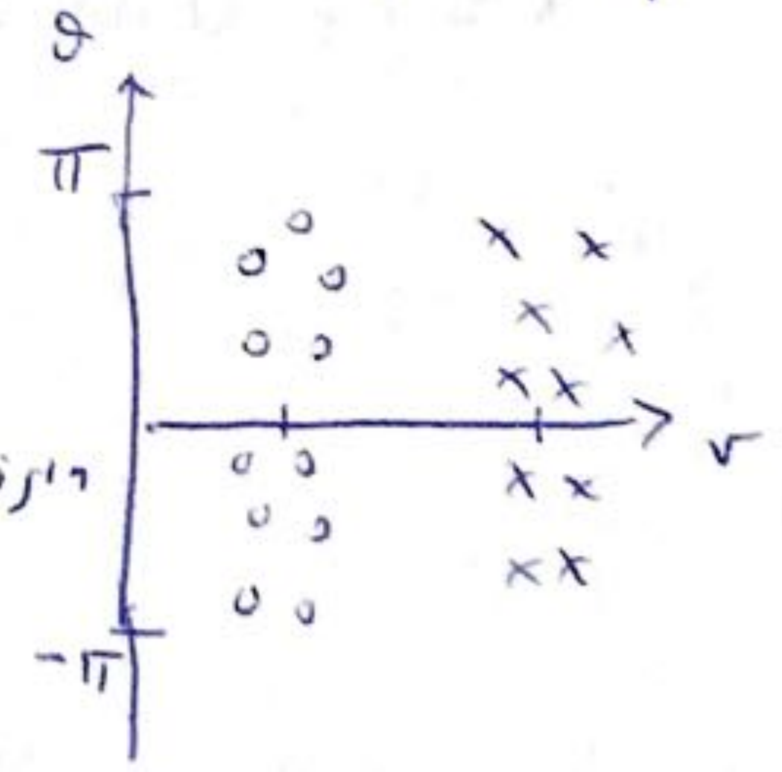
if



polar coordinates: maps cartesian coordinates to r and ϑ

r: how far a point is from the origin

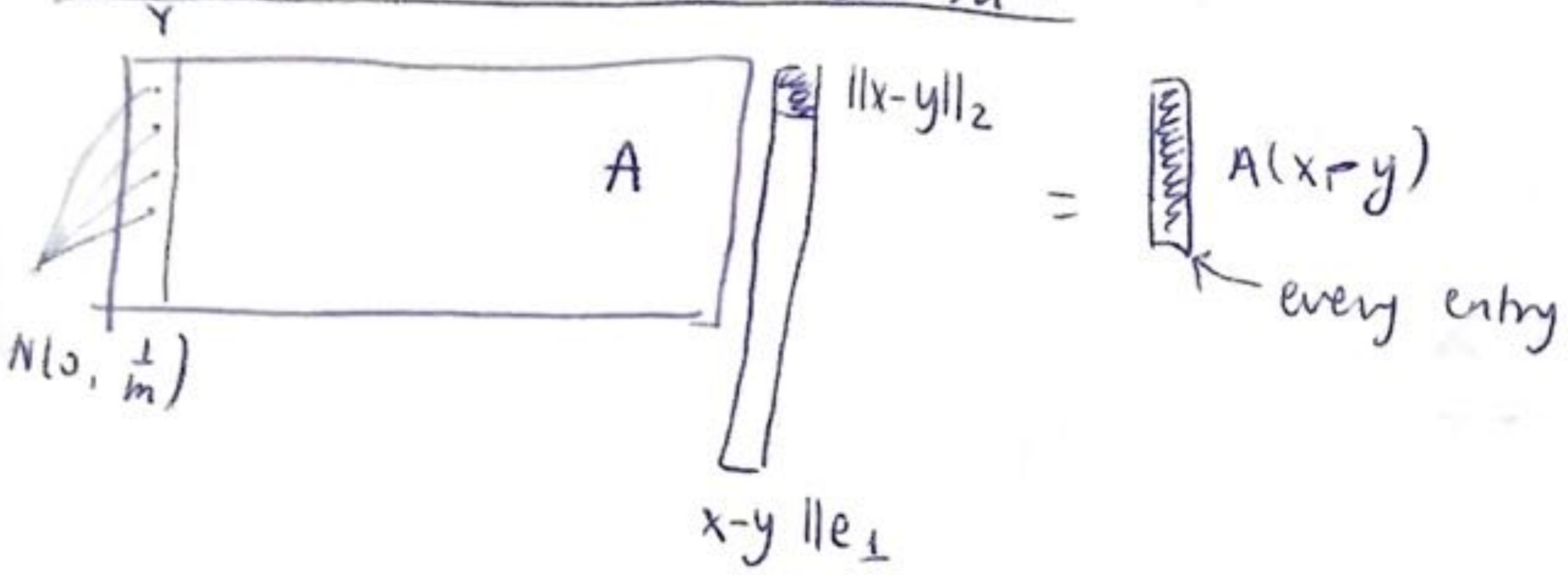
ϑ: which angle from the origin



* for the perceptron algorithm the VC dim $= d + 1$

Chapter 11: Dimensionality

Johnson - Lidenstrauss Lemma:



$$\|A(x-y)\|_2 = (1 \pm \varepsilon)\|x-y\|_2$$
$$\Leftrightarrow \|Y\|_2 = 1 \pm \varepsilon \quad (1)$$

To prove (1) we can use:

Chernoff-style Claim:
- Let $z_1, z_2, \ldots z_m \sim N(0,1)$
- $Pr[\sum_i z_i^2 > (1+\varepsilon)\cdot m] \le e^{-m \cdot \varepsilon^2/8}$

$$P[\|Y\|_2 > 1+\varepsilon] = P[\|Y\|_2^2 > (1+\varepsilon)^2] \le P[\|Y\|_2^2 > 1+\varepsilon]$$
$$= P(\textstyle\sum_i z_i^2 > m(1+\varepsilon)] \le e^{-m \cdot \varepsilon^2/8}$$

$$P[\|A(x-y)\|_2 \ge (1+\varepsilon)\|x-y\|_2] \le e^{-m\cdot\varepsilon^2/8}$$
$$P[\exists x, y \in X \text{ s.t. } \|A(x-y)\|_2 \ge (1+\varepsilon)\|x-y\|_2] \le n^2 e^{-m\varepsilon^2/8}$$

$$m = O\left(\frac{\log n}{\varepsilon^2}\right) \le 1/\text{polynomial}(n)$$

$$Pr[\textstyle\sum_i z_i^2 > (1+\varepsilon) m] = P[e^{t\sum z_i^2} > e^{t(1+\varepsilon)m}] \overset{\text{use Markov's Inequality}}{\Longrightarrow} \le \frac{\prod_i \mathbb{E}\, e^{t z_i^2}}{e^{tm(1+\varepsilon)}}$$

(z: independent so $\prod_i$ can go out of expectation)

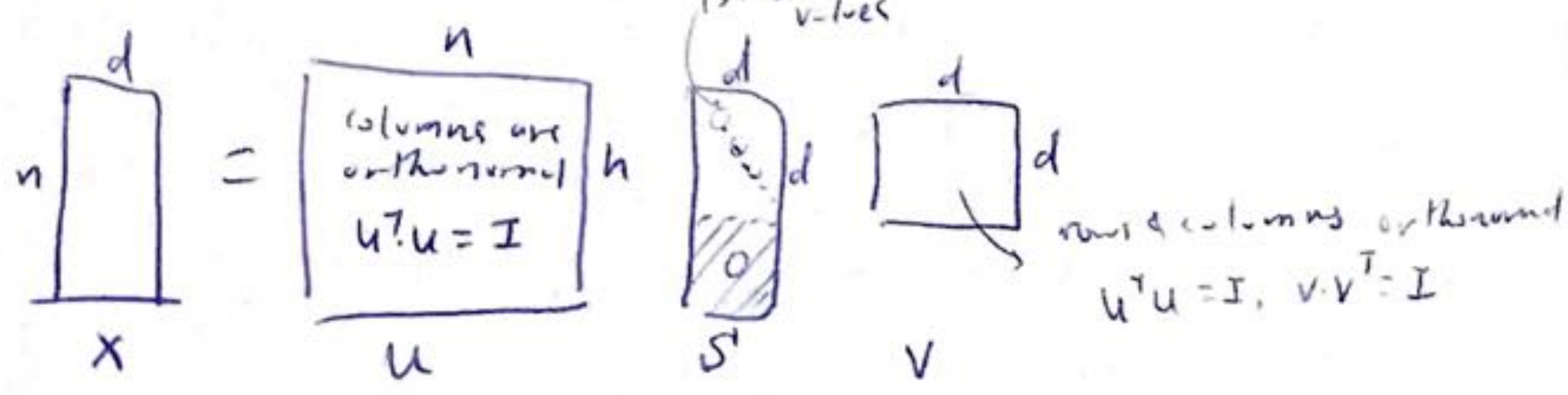
$$= \left(\frac{1}{1-2t}\right)^{m/2} \cdot \frac{1}{e^{tm(1+\varepsilon)}} \quad \text{for } t < 1/2$$

$$= \exp\left(m\left[\tfrac{1}{2}\log\left(\tfrac{1}{1-2t}\right) - t(1+\varepsilon)\right]\right) \le \exp(m(2t^2 - \varepsilon t)) \quad \text{for } 0 < t < \tfrac{1}{3}$$

for $t = \frac{\varepsilon}{4}$: $\quad = \exp(-m\varepsilon^2/8)$.

SVD

$$\underset{n\times d}{X} = \underset{n\times n}{U} \cdot \underset{n\times d}{S} \cdot \underset{d\times d}{V^T}$$



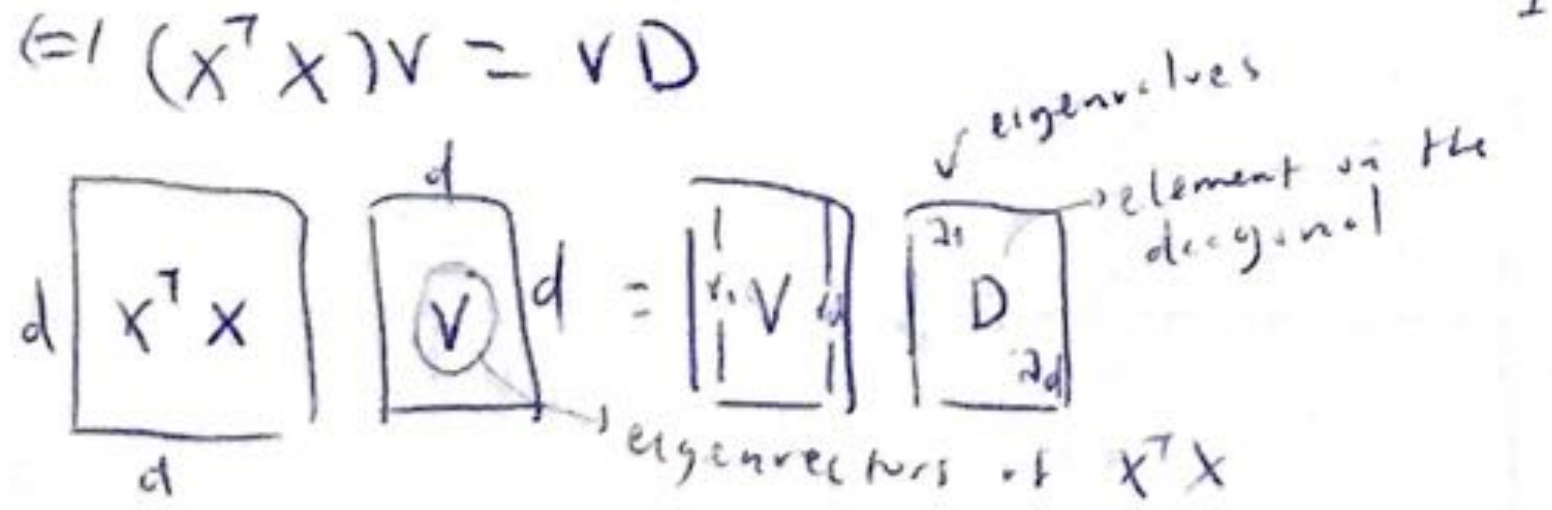
PCA

$X = USV^T$ Then:

$$X^T \cdot X = V S^T \underbrace{U^T U}_{I} \cdot S \cdot V^T = V S^T S \cdot V^T = V D \cdot V^T$$

, $D = S^T S$ → diagonal matrix with the squares of singular values

$\Leftrightarrow (X^T X) V = V D$

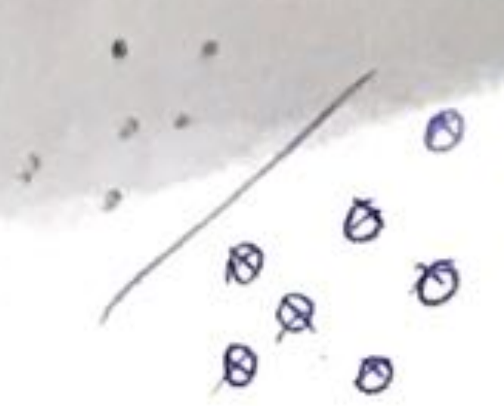


Covariance matrix: $\hat{\Sigma} = \frac{1}{N}\sum^{N} x^n (x^n)^T = \frac{1}{n} X^T X$

(PCA)

* You can get the eigenvalues/vectors of the Covariance matrix by doing SVD on $X^T X$
* Based on what projection I want I keep that many V-number vectors

Vapnik-Chervonenkis Dimension:



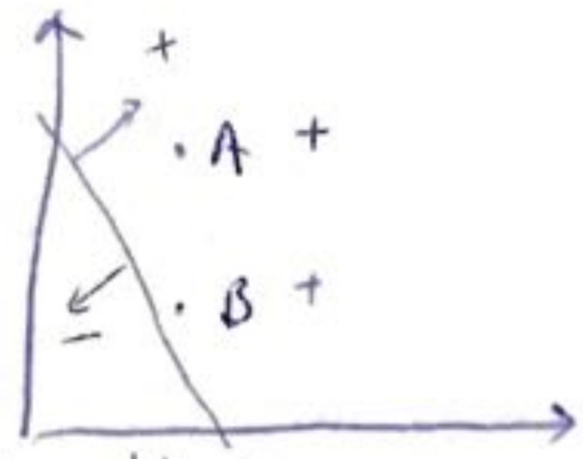
VC-dim

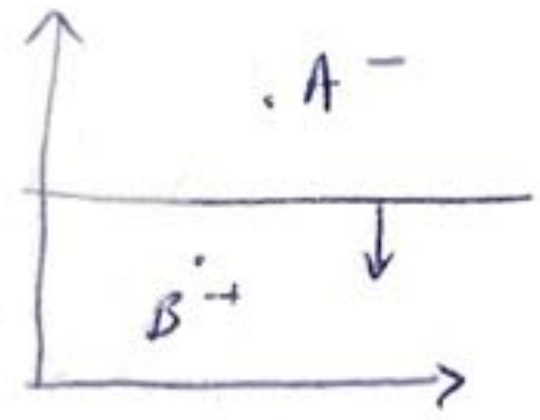
"Shattering"

$\{x_i, \theta_i\}$, $i = 1, 2, \dots n$

$\theta_i = \{+, -\}$ $x_i \in \mathbb{R}^n$

$n = 2$ (2 dimensions)





when I have 2 points

for $n=2 \Rightarrow 4$ possible different labelling

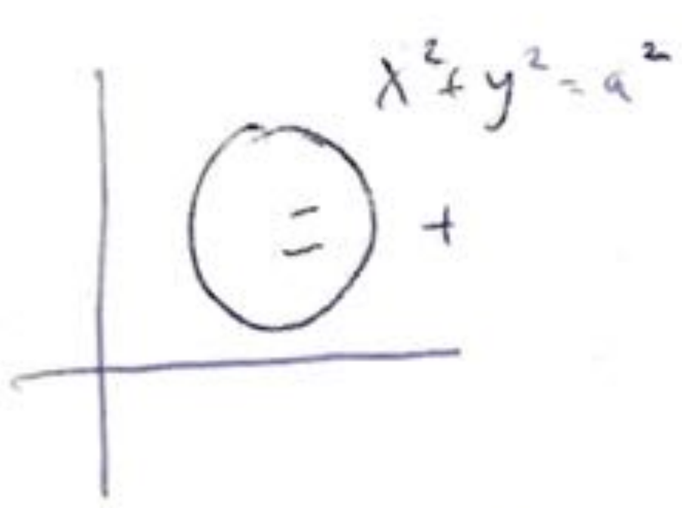
if $n=3 \Rightarrow 8$ —''—

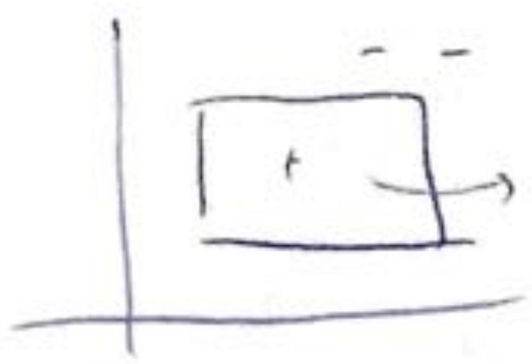
$2^n \rightarrow$ points

The max number of points that can be shattered by a specific set of T (functions)

$T = \{f(a)\} \Rightarrow T = \{$"set of line"$\}$

$T = \{$"a circle"$\}$ for non-linear decision boundary





→ decision trees

when VC dim = 2 you can classify up to 3 data points, for more than this you cannot

Define "risk" and "empirical risk" as the long term expected test & observed training error.
↳ test error ↳ train error

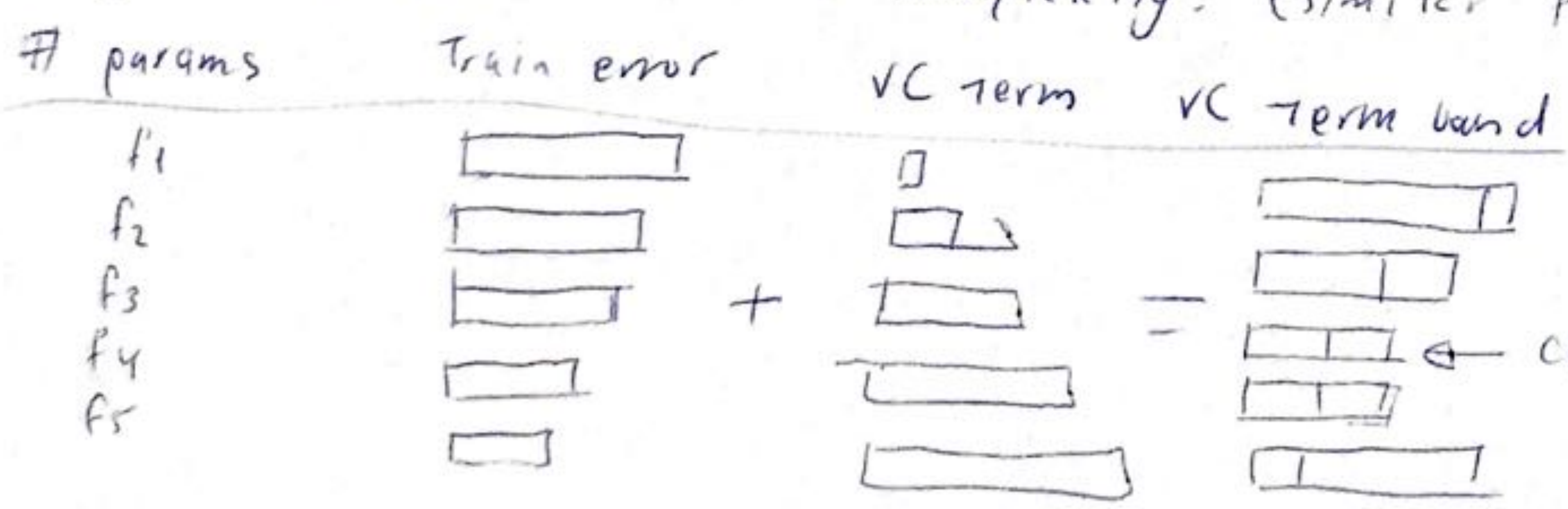
underfitting: will be similar overfitting: test getting worse

Let VC dimension be H, then with "high-probability" $(1-n)$ Vapnik showed:
(size n)

$$\text{Test error} \leq \text{Train error} + \sqrt{\frac{H \log(2m/H) + H - \log(n/4)}{m}}$$

shattering: we say a classifier $f(x)$ can shatter points $x^{(1)} \dots x^{(n)}$ iff for all $y^{(1)} \dots y^{(n)}$ $f(x)$ can achieve zero error on training data $(x^{(1)}, y^{(1)}), \dots (x^{(n)}, y^{(n)})$ - there exists some θ.

using VC dimension to select complexity: (similar to CV)



← Choose this - Structural Risk Minimisation