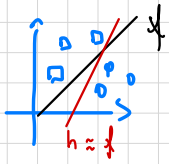
SUMMARY

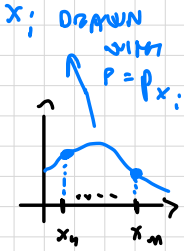


- **PAC LEARNING** :

LET f BE THE CORRECT CLASSIFIER ON $X = \{x_1, \dots, x_m\}$ (DATASET)

-> WE SHOULD FIND A FUNCTION $h \approx f$ (CORRECT ONLY FOR TRAINING SET ; CORRECT FOR ALL DATASET ~> AN APPROXIMATION)

- ONE POSSIBLE DEFINITION IS THE ERROR OF h WITH REFERENCE TO f :



$$L_{D,f}(h) := \underset{x \sim D}{P}\left[h(x) \neq f(x)\right] := D\left(\{x \in X : h(x) \neq f(x)\}\right)$$

- D : A DISTRIBUTION OVER X -> GIVEN $A \subset X$, $D(A) = P[x \in A]$

- SIMPLE DATA GENERATION MODEL :
  - iid : $\forall x_i$ IS SAMPLED INDEPENDENTLY ACCORDING TO D
  - REALIZABILITY : $\forall i \in [m]$, $y_i = f(x_i)$

- WE CANNOT HOPE TO FIND h / $L_{D,f}(h) = 0$
  - PROOF :
    - $\forall \varepsilon \in (0,1)$, TAKE $X = \{x_1, x_2\}$, D : $\begin{cases} D(\{x_1\}) = 1 - \varepsilon \\ D(\{x_2\}) = \varepsilon \end{cases}$
    - THE PROBABILITY OF NOT SEEING x_2 AMONG m EXAMPLES
      -> $(1-\varepsilon)^m \approx e^{-\varepsilon m}$ ~> PROPERTY
    - -> IF $\varepsilon \cdot m \ll 1$: WE ARE UNLIKELY NOT TO SEE x_2 AT ALL

L> RELAXATION n°1 : WE'D BE HAPPY TO FIND $L_{D,f}(h) \leq \varepsilon$ ~> USER SPECIFIED

- **N.B.** : $L_{D,f}(h) = 0 \not\Rightarrow h = f$, ∵ IS TRUE ONLY ON THE PROBABILITY SET CONSIDERED FOR TRAINING, NOT ALL DATA

- NO ALGORITHM CAN GUARANTE $L_{(D,f)}(h) \leq \varepsilon$

  ↳ RELAXATION n°2 . WE ALLOW ALGORITHM TO FAIL $\leq \varepsilon$ WITH PROBABILITY $\delta \in (0,1)$ ~> LIKE C.I.

  $1-\delta$ ↑ PROBABLY, $\leq \varepsilon$ ↑ APPROXIMATELY CORRECT

- PAC LEARNING := PROBABLY APPROXIMATELY CORRECT LEARNING
  - THE LEARNER DO NOT KNOW D AND f
  - THE LEARNER RECIEVES ε (ACCURACY PARAMETER) AND δ (CONFIDENCE PARAMETER)
  - THE LEARNER CAN ASK FOR TRAINING DATA S, WITH m SAMPLES / m (.) ε, δ
  - THE LEARNER SHOULD OUTPUT AN HYPOTHESIS h WITH PROBABILITY OF AT LEAST $1-\delta$ / IT HOLDS THAT $L_{D,f}(h) \leq \varepsilon$

  -> $P_{S \sim D^m}\left[L_{D,f}(h) \leq \varepsilon\right] \geq 1-\delta$

  δ -> P OF SUCCESS

  $1-\delta$ -> P OF FAILURE

- THEOREM (NO FREE LUNCH):

  FIX $\delta \in (0,1)$, $\varepsilon < 1/2$:

  -> $\forall$ LEARNER A AND TRAINING SIZE m :

  -> $\exists\, D, f$ / WITH $p \geq \delta$, ONDE TRAINING DATA S OF m EXAMPLES IT HOLDS THAT $L_{D,f}(A(S)) \geq \varepsilon$

  - EXPLAINATION :
    - $\forall$ LEARNER -> $\exists$ A FUNCTION WHICH DO NOT SUCCEDES IN $L_{D,f} \leq \varepsilon$
    - $\nexists$ A UNIQUE LEARNER : EVERY TIME A SPECIFIC ONE IS NEEDED
    - THE AVG RESULT OF ALL CLASSIFIER IS WORST THEN RANDOM GUESSING

      ↳ $L_{D,f}$ (RANDOM GUESSING) $= 1/2$
    - IF NO KNOWLEDGE IS INSERTED -> $\nexists$ SOMETHING BETTER THEN RANDOM GUESSING

- PRIOR KNOWLEDGE :

IF LEARNER KNOWS $H \subset Y^X$ (↳ HYPOTHESIS CLASS OF f) -> PAC LEARNING IS POSSIBLE IF $H \neq Y^X$ (NO FREE LUNCH THEOREM)

- PAC LEARNING OF FINITE HYPOTHESIS CLASS :

ASSUME H : FINITE HYPOTHESIS CLASS (ex. ALL $h: X \mapsto Y$)

- EMPIRICAL RISK MINIMIZATION (ERM) : -> $h = ERM_H(S)$

ERM LEARNING RULE

- INPUT : TRAINING SET $S = (x_1, y_1), \dots, (x_m, y_m)$
- EMPIRICAL RISK : $L_S(h) = \frac{1}{m} \left| \{ i : h(x_i) \neq y_i \} \right|$ } 0-1 LOSS
- OUTPUT : ANY $h \in H$ / h MINIMIZES $L_S(h)$

- THEOREM :

FIX ε, δ

IF $m \geq \frac{\log(|H|/\delta)}{\varepsilon}$ -> $\forall D, f$, WITH $p \geq 1-\delta$ (OVER THE CHOICE OF S SIZE m) : $L_{D,f}(ERM_H(S)) \leq \varepsilon$ (h / SATISFIES ERM)

(PROOF NOT DONE FOR NOW)

- PAC LEARNABILITY :

↳ $m \geq \frac{\log(|H|/\delta)}{\varepsilon} \geq m_H(\varepsilon, \delta)$

AN HYPOTHESIS CLASS H IS PAC LEARNABLE

IF $\exists$ FUNCTION $m_H : (0,1)^2 \mapsto \mathbb{N}$ (↳ FOR ε, δ) AND A LEARNING ALGORITHM / (SAMPLE COMPLEXITY OF LEARNING H) :

- $\forall \varepsilon, \delta \in (0,1)$
- $\forall D$ OVER X, AND $\forall$ LABELING FUNCTION $f : X \mapsto \{0,1\}$

WHEN RUNNING THE ALGORITHM ON $m \geq m_H(\varepsilon, \delta)$ iid EXAMPLES GENERATED BY D AND LABELED BY f (↳ SAMPLE COMPLEXITY : min # INPUT DATA TO ASSURE)

-> ALGORITHM RETURNS h / WITH $p \geq 1-\delta : L_{D,f}(h) \leq \varepsilon$

- COROLLARY : LET H BE A FINITE HYPOTHESIS CLASS:

-> H IS PAC LEARNABLE WITH $m_H(\varepsilon, \delta) \leq \frac{\log(|H|/\delta)}{\varepsilon}$ / m_H OBTAINED USING ERM_H LEARNING RULE

VC DIMENSION :

- LET $C = \{x_1, \dots, x_{|C|}\} \subset X$ — SET OF POINTS, WITH CARDINALITY $|C|$
- LET H_C BE A RESTRICTION OF H TO SET C : $H_C = \{h_C : h \in H\}$

  / $h_C : C \mapsto \{0, 1\}$ AND $h_C(x_i) = h(x_i), \forall x_i \in C$

  ↳ IT'S A FUNCTION THAT USE THE EACH POINT $\in C$ TO $\{0, 1\}$

  → h_C CAN BE REPRESENTED AS A VECTOR : $(h(x_1), \dots, h(x_{|C|})) = \{\pm 1\}^{|C|}$

  → $|H_C| \leq 2^{|C|}$

- WE SAY THAT H "SHATTERS" C IF $|H_C| = 2^{|C|}$

  ↳ IF A SET OF FUNCTIONS $\in H_C$ IS ABLE TO PRODUCE EVERY COMBINATION OF $\{\pm 1\}^{|C|}$ OUTPUTS → H SHATTERS C

- $VC\ dim(H) = d = \sup\{|C| : H \text{ SHATTERS } C\}$

  ↳ IT IS THE HIGHEST CARDINALITY OF A SET OF C / H SHATTERS THE SET C

- IN ORDER TO SHOW THAT $VC\ dim(H) = d$:

  1. SHOW THAT $\exists$ A SET C OF SIZE $|C| = d$ / C IS SHATTERED BY H → $VC\ dim(H) \geq d$ (→ JUST 1 SET IS NEEDED)
  2. $\forall$ SET C / $|C| = d+1$, C IS NOT SHATTERED BY H → $VC\ dim(H) < d+1$

  → ① + ② : $VC\ dim(H) = d$

ex.

$$H = \{x \mapsto sign(x - \theta) : \theta \in \mathbb{R}\}, \quad X \in \mathbb{R}$$

→

- SHOW THAT $\{0\}$ IS SHATTERED :

  $C = \{0\}$ → $|C| = 1$

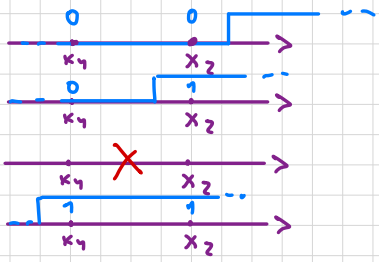
  → $H_C \overset{x=0}{=} \{sign(-\theta) : \theta \in \mathbb{R}\}, x=0 \mapsto \{\pm 1\}$ → H SHATTERS C ($\sim 2^{|C|} = 2^1 = 2$)

  → $VC\ dim(H) \geq 1$

- SHOW THAT ANY 2 POINTS CANNOT BE SHATTERED :

x_1	x_2
0	0
0	1
1	0
1	1

$2^{|C|} = 2^2$



$\rightarrow VC\ dim(H) < 2 \quad \rightarrow \quad VC\ dim(H) = 1$

ex.

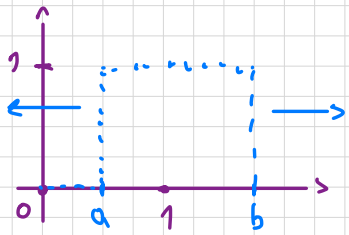
$$H = \{ h_{a,b} : a < b \in \mathbb{R} \}, \quad h_{a,b}(x) = 1 \iff x \in [a,b], \quad x \in \mathbb{R}$$

$\rightarrow$

- SHOW THAT $\{0,1\}$ IS SHATTERED :

  - SHOW THAT $\{0,1\} \mapsto \{\pm 1\}^{|C|=2}$ $\rightarrow$

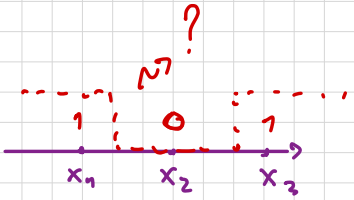
0	0
0	1
1	0
1	1



  - $(0,0)$: CHOOSE $a > 1$ $\rightarrow$ ex. $h_{2,3}$
  - $(0,1)$: CHOOSE $a \in [0,1]$, $b > 1$ $\rightarrow$ ex. $h_{0.5,3}$
  - $(1,0)$: CHOOSE $a < 0$, $b \in [0,1]$ $\rightarrow$ ex. $h_{-1,0.5}$
  - $(1,1)$: CHOOSE $a < 0$, $b > 1$ $\rightarrow$ ex. $h_{-1,2}$

- SHOW THAT ANY 3 POINTS CANNOT BE SHATTERED :

$\rightarrow$ THE TRIPLET $(1,0,1)$ IS NOT POSSIBLE



$\rightarrow VC\ dim(H) \geq 2, \quad VC\ dim(H) < 3 \quad \rightarrow VC\ dim = 2$

ex. VC DIM OF HALFSPACES

(SEE LESSONS NOTES, PAG. 19)

- **THE FUNDAMENTAL THEOREM OF PAC LEARNING:**

LET:

- H BE A FINITE HYPOTHESIS CLASS OF FUNCTIONS $X \longrightarrow \{0,1\}$
- LOSS FUNCTION: 0-1 LOSS

$\longrightarrow$ TFAE: ~> THE FOLLOWING 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
5. ANY ERM RULE IS A SUCCESSFUL PAC LEARNER FOR H
6. H HAS A FINITE VC-DIM $= d < \infty$

-> $VC\,DIM(H) = d$ -> H IS PAC LEARNABLE

- THEOREM:

LET:

- H BE A FINITE HYPOTHESIS CLASS OF FUNCTIONS $X \longrightarrow \{0,1\}$
- LOSS FUNCTION: 0-1 LOSS
- $VC\,DIM = d < \infty$

-> $\exists\ C_1, C_2$ CONSTANTS, SUCH THAT:

$$C_1 \cdot \frac{d + \log(1/\delta)}{\varepsilon} \leq m_H(\varepsilon, \delta) \leq C_2 \frac{d \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}$$

AND $m_H(\varepsilon, \delta)$ IS ACHIEVED BY THE ERM LEARNING RULE

(NO PROOF)

- LEMMA:

LET H BE AN HYPOTHESIS CLASS $/ VC\,DIM(H) = d < \infty$

-> $\forall C \subset X \,/\, |C| = m > d+1$ -> $|H_C| \leq \left(\frac{e\,m}{d}\right)^d \approx K \cdot m^d$

GENERAL LEARNING MODEL AND BIAS COMPLEXITY TRADEOFF :

THE GENERAL PAC MODEL :

RELAXING THE REALIZABILITY ASSUMPTION - AGNOSTIC PAC LEARNING :

SO FAR WE ASSUMED THAT Y (LABELS) GENERATED BY $f \in H$

-> THIS ASSUMPTION IS TOO STRONG :

"TARGET LABELING FUNCTION" ⟼ "DATA-LABELS GENERATING DISTRIBUTION"

	PAC	Agnostic PAC
Distribution	$\mathcal{D}$ over $\mathcal{X}$	$\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$
Truth	$f \in \mathcal{H}$	not in class or doesn't exist
Risk	$L_{\mathcal{D},f}(h) = \mathcal{D}(\{x : h(x) \neq f(x)\})$	$L_{\mathcal{D}}(h) = \mathcal{D}(\{(x,y) : h(x) \neq y\})$
Training set	$(x_1,\ldots,x_m) \sim \mathcal{D}^m$ $\forall i,\ y_i = f(x_i)$	$((x_1,y_1),\ldots,(x_m,y_m)) \sim \mathcal{D}^m$
Goal	$L_{\mathcal{D},f}(A(S)) \leq \epsilon$	$L_{\mathcal{D}}(A(S)) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$

AGNOSTIC PAC LEARNING :

- D OVER $X \times Y$
- RISK : $L_D(h) \overset{def}{=} P_{(x,y)\sim D}[h(x) \neq y] \overset{def}{=} D(\{(x,y) : h(x) \neq y\})$ (GENERIC LOSS OF CLASS)
- "APPROXIMATELY CORRECT" NOTION : $L_D(A(S)) \leq \min_{h \in H} L_D(h) + \epsilon$ (-> MINIMUM LOSS AMONG ALL FUNCTIONS)
  - -> A: LEARNER, S: TRAINING SET

LOSS FUNCTION :

LET $Z = X \times Y$. GIVEN $h \in H$ AND AN EXAMPLE $(\bar{x}, y) \in Z$,

-> THE LOSS FUNCTION CAN TELL HOW GOOD IS h ON $(\bar{x}, y)$:

- $\ell : H \times Z \longmapsto \mathbb{R}^+$

- EXAMPLES :
  - 0-1 LOSS : $\ell(h,(x,y)) = \begin{cases} 1, & \text{IF } h(x) \neq y \\ 0, & \text{IF } h(x) = y \end{cases}$
  - SQUARED LOSS : $\ell(h,(x,y)) = (h(x) - y)^2$
  - ABSOLUTE VALUE LOSS : $\ell(h,(x,y)) = |h(x) - y|$
  - COST-SENSITIVE LOSS : $\ell(h,(x,y)) = C_{h(x),y}$

/ C IS A $|Y| \times |Y|$ MATRIX

- **GENERAL PAC LEARNING PROBLEM:**

$\frac{1}{m}\sum_{i}^{m} \ell(h, z_i)$

WE WISH TO PA SOLVE: $\min_{h \in H} L_D(h)$ / $L_D(h) := \underset{z \sim D}{E}[\ell(h, z)]$

- LEARNER KNOWS H, $Z = X \times Y$ AND ℓ, AND RECEIVES ε (ACCURACY), δ (CONFIDENCE) AS PARAMETERS
- LEARNER CAN DECIDE ON TRAINING SET SIZE m, BASED ON ε, δ
- LEARNER DOESN'T KNOW D, BUT CAN SAMPLE $S \sim D^m$

→ USING S, LEARNER OUTPUTS AN HYPOTHESIS $A(S)$

→ WE WANT TO HOLD: $L_D(A(S)) \leq \min_{h \in H} L_D(h) + \varepsilon$, WITH $p \geq 1 - \delta$

- **A MORE FORMAL DEFINITION:**

AN HYPOTHESIS CLASS H IS AGNOSTIC PAC LEARNABLE TO A SET $Z = X \times Y$ AND A LOSS FUNCTION $\ell : H \times Z \longmapsto \mathbb{R}^+$,

IF $\exists$ A FUNCTION $m_H : (0,1)^2 \longmapsto \mathbb{N}$ AND A LEARNING ALGORITHM A WITH THE PROPERTY:

$\forall \varepsilon, \delta \in (0,1)$, $m \geq m_H(\varepsilon, \delta)$, AND A DISTRIBUTION D OVER Z (THEORETICAL TRUE DISTRIBUTION OF Z).

$$L_{S \sim D^m}\left(\left\{ S \in Z^m : L_D(A(S)) \leq \min_{h \in H} L_D(h) + \varepsilon \right\} \right) \geq 1 - \delta$$

- S: DATASET
- Z^m: ALL POSSIBLE m-TUPLES IN Z ($\sim Z \times \dots \times Z$)
- $A(S)$: LEARNER A ON S
- $L_D(A(S))$: EXPECTED LOSS IS AT MOST ε-LARGER THAN THE EXPECTED LOSS OF THE BEST h
- $\min_{h \in H} L_D(h)$: LOSS ON BEST h
- $1 - \delta$: C.I.
- D^m: CARTESIAN PRODUCT → $D \times \dots \times D$ / EACH $Z \sim D$

LEARNING VIA UNIFORM CONVERGENCE:

REPRESENTATIVE SAMPLE:

- A TRAINING SET S IS ε-REPRESENTATIVE SAMPLE IF

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

- LEMMA:

  ASSUME THAT A TRAINING SET S IS $\frac{\varepsilon}{2}$-REPRESENTATIVE

  → ANY OUTPUT OF $ERM_H(S)$: $h_S \in \underset{h \in H}{\arg\min}\, L_S(h)$

  SATISFIES: $L_D(h_S) \leq \min_{h \in H} L_D(h) + \varepsilon$

- → IF S IS $\frac{\varepsilon}{2}$-REPRESENTATIVE → $L_D(h_S) \leq \min_{h \in H} L_D(h) + \varepsilon$ (OUTPUT OF $ERM_H(S)$ → $h_S = ERM_H(S)$)
- PROOF ($\forall h \in H$):

$$L_D(h_S) \leq L_S(h_S) + \frac{\varepsilon}{2} \overset{\text{LEMMA}}{\leq} L_S(h) + \frac{\varepsilon}{2} \leq \underbrace{L_D(h) + \frac{\varepsilon}{2}}_{L_S(h)} + \frac{\varepsilon}{2} \leq L_D(h) + \varepsilon$$

UNIFORM CONVERGENCE: → IF UC → H CAN LEARN WELL FROM S

H HAS THE UNIFORM CONVERGENCE PROPERTY

IF $\exists$ FUNCTION $m_H^{UC}: (0,1)^2 \longmapsto \mathbb{N}$ / $\forall \varepsilon, \delta \in (0,1)$ AND $\forall D$.

$$\text{↳ } D^m\left(\left\{S \in Z^m : S \text{ IS } \varepsilon\text{-REPRESENTATIVE}\right\}\right) \geq 1 - \delta$$

- → IF $\exists\, m_H^{UC}$ / $D^m\left(\left\{S \in Z^m : S \text{ IS } \varepsilon\text{-REPRESENTATIVE}\right\}\right) \geq 1 - \delta$ → UC

COROLLARY: → IF UC → ∃ BOUND (UPPER) FOR $m_H(\varepsilon, \delta)$

- IF H HAS UC PROPERTY WITH A FUNCTION m_H^{UC}

  → H IS AGNOSTICALLY PAC LEARNABLE WITH $m_H(\varepsilon, \delta) \leq m_H^{UC}\left(\frac{\varepsilon}{2}, \delta\right)$

- IN THAT CASE, ERM_H PARADIGM IS A SUCCESSFUL AGNOSTIC PAC LEARNER FOR H

- **THEOREM** (FINITE CLASS -> AGNOSTIC PAC LEARNABLE): (NO PROOF, FOR NOW)

  LET H BE FINITE AND $\ell \in [0,1]$

  -> H IS AGNOSTICALLY PAC LEARNABLE WITH:

  $$m_H(\varepsilon, \delta) \leq \left\lceil \frac{2 \log\left(2|H|/\delta\right)}{\varepsilon^2} \right\rceil$$

  ~> $\lceil .. \rceil$: CEILING FUNCTION
  ↳ eg. $\lceil 2.4 \rceil = 3$

- **DISCRETIZATION TRICK:**

  SUPPOSE H PARAMETRIZED BY d NUMBERS / EACH NUMBER REPRESENTED BY b BITS

  -> $|H| \leq 2^{d \cdot b}$

  -> $m_H(\varepsilon, \delta) \leq \left\lceil \frac{2db + 2\log\left(2/\delta\right)}{\varepsilon^2} \right\rceil$

LINEAR REGRESSION AND LEAST SQUARE:

- $X \subset \mathbb{R}^d$, $Y \subset \mathbb{R}$
- $H = \{ \bar{x} \mapsto \langle \bar{w}, \bar{x} \rangle : w \in \mathbb{R}^d \}$
- LOSS $\longrightarrow$ SQUARED LOSS: $\ell\left(h, (\bar{x}, y)\right) = \left(h(\bar{x}) - y\right)^2$
- ERM PROBLEM: $\min_{\bar{w} \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} \left(\langle \bar{w}, \bar{x}_i \rangle - y_i \right)^2$ $\rightarrow$ MINIMIZE MSE

  OR, EQUIVALENTLY: $\min_{\bar{w} \in \mathbb{R}^d} \| \underset{m \times d}{X^T} \cdot \underset{d \times 1}{\bar{w}} - \underset{m \times 1}{\bar{y}} \|^2$

- GRADIENT AND OPTIMIZATION:

$$\nabla f(\bar{x}) = \left[\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_d} \right] \quad / \quad f : \mathbb{R}^d \mapsto \mathbb{R}$$

$\rightarrow$ IF $\bar{x}$ MINIMIZES $f(\bar{x})$ $\rightarrow$ $\nabla f(\bar{x}) = (0, \dots, 0)$

- JACOBIAN:

$$\bar{f} : \mathbb{R}^n \mapsto \mathbb{R}^m$$

$$J_x(\bar{f}) = \begin{pmatrix} \nabla f_1(\bar{x}) \\ \vdots \\ \nabla f_m \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

$m \times n$

ex. $f(x, y) = (\sin x + y, xy) \rightarrow J_x(\bar{f}) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} \cos x & 1 \\ y & x \end{pmatrix}$

- IF $\bar{f}(\bar{w}) = \underset{m \times n}{A} \bar{w}$ $\rightarrow$ $J_x(\bar{f}) = A$

- CHAIN RULE:

  WHEN $f : \mathbb{R}^n \mapsto \mathbb{R}^m$, $y : \mathbb{R}^k \longrightarrow \mathbb{R}^n$:

$$J_x(f \circ y) = J_{y(x)}(f) \cdot J_x(y)$$

$f(y(x))$ $\quad \frac{df}{dy} \cdot \frac{dy}{dx}$

$\rightsquigarrow$ ANALOG OF $\left(f(y(x))\right)' = f'(y(x)) \cdot y'(x)$

• LEAST SQUARES : →

PROBLEM: $\min_{\bar{w} \in \mathbb{R}^d} \frac{1}{2} \| X^T \cdot \bar{w} - \bar{v} \|^2$ $\quad \langle \bar{w}, \bar{x} \rangle - \bar{v}$

↳ SOL: $\bar{w} = (X X^T)^{-1} X v$ ↳ IF EXISTS

RECALL: WE'D LIKE TO SOLVE THE ERM PROBLEM $\min_{\bar{w} \in \mathbb{R}^d} \frac{1}{2} \| X^T \cdot \bar{w} - \bar{v} \|^2$

- LET $\bar{y}(\bar{w}) = X^T \cdot \bar{w} - \bar{v}$, $\bar{f}(\bar{v}) = \frac{1}{2} \| \bar{v} \|^2 = \frac{1}{2} \sum_{i=1}^{m} v_i^2$

↳ ERM PROBLEM $\mapsto \min_{\bar{w}} \bar{f}(\bar{y}(\bar{w}))$ ↳ $\begin{cases} J_{\bar{w}}(\bar{y}) = X^T \\ J_{\bar{v}}(\bar{f}) = (v_1, \dots, v_m) \end{cases}$

$$\rightarrow J_{\bar{w}}(\bar{f} \circ \bar{y}) \overset{\text{CHAIN RULE}}{=} J_{\bar{y}(\bar{w})}(\bar{f}) \cdot J_{\bar{w}}(\bar{y}) = \bar{y}(\bar{w})^T \cdot X^T = (X^T \bar{w} - v)^T X^T$$

- OPTIMIZATION → SET $J_{\bar{w}}(\bar{f} \circ \bar{y}) = \bar{0}$

$(X^T w - v)^T \cdot X^T = 0$
$(X^T w)^T - v^T) X^T = 0$
$(w^T X - v^T) X^T = 0$
$w^T X X^T - v^T X^T = 0$
$w^T X X^T = v^T X^T$
$X X^T w = X v$

$$\rightarrow \left(X^T \bar{w} - \bar{v} \right)^T \cdot X^T = \bar{0}^T \rightarrow X \cdot X^T \bar{w} = X \bar{v}$$ (UNKNOWN)

→ IF $\exists (X \cdot X^T)^{-1}$ → $\bar{w} = (X X^T)^{-1} \cdot X \bar{v}$

↳ WHAT IF $\nexists (X X^T)^{-1}$ → SOL: MOORE-PENROSE PSEUDO INVERSE

- INTERPRETATION AS PROJECTION;

IF $\bar{w}$ IS THE LEAST SQUARE SOLUTION:

→ $\hat{\bar{v}} = X^T \bar{w}$ IS THE VECTOR IN $C = \{ X^T \bar{w} : \bar{w} \in \mathbb{R}^d \}$ CLOSEST TO $\bar{v}$

↳ PROJECTION OF $\bar{v}$ ONTO C

- WE CAN FIND $\hat{\bar{v}} = V \cdot V^T \cdot v$ / $V_{m \times d}$: COLUMNS ARE ORTHONORMAL BASIS OF THE RANGE OF X^T

POLYNOMIAL FITTING :

- 1-DIMENSIONAL POLYNOMIAL FUNCTION OF DEGREE n :

$$p(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \dots + a_n x^n$$

- GOAL : GIVEN DATA $S = \left((x_1, y_1), \dots, (x_m, y_m) \right)$ -> FIND ERM WITH RESPECT TO THE CLASS OF POLYNOMIALS OF DEGREE n

- REDUCTION TO LINEAR REGRESSION :
  - DEFINE $\psi : \mathbb{R} \longmapsto \mathbb{R}^{n+1}$, $\psi(x) = (1, x, x^2, \dots, x^n)$
  - DEFINE $\bar{a} = (a_0, a_1, \dots, a_n)$ AND OBSERVE :

$$p(x) = \sum_{i=0}^{n} a_i x^i = \langle \bar{a}, \psi(x) \rangle$$ -> HALF SPACE / LIN. REGRESSION

-> TO FIND $\bar{a}$ WE CAN SOLVE LS (EASY SOLVERS) W.R.T. $\left((\psi(x_1), y_1), \dots, (\psi(x_m), y_m) \right)$

BIAS - COMPLEXITY TRADE OFF :

ERROR DECOMPOSITION :

LET $h_S = ERM_H(S)$ → BEST HYPOTHESIS FUNCTION BASED ON S. WE CAN DECOMPOSE THE RISK OF h_S AS :

$$L_D(h_S) = \underbrace{\varepsilon_{APP}}_{\text{APPROXIMATION ERROR}} + \underbrace{\varepsilon_{EST}}_{\text{ESTIMATION ERROR}}$$

- **APPROXIMATION ERROR** → (!) H → $\varepsilon_{APP} = \min_{h \in H} L_D(h)$

  ex. CHANGING POLYNOMIAL DEGREE

  → IT IS THE RISK WE MAKE DUE TO RESTRICTING TO H → ex. FIXING n AS POLYNOMIAL DEGREE

  $\uparrow n \rightarrow \downarrow \varepsilon_{APP}$ → IT DOES NOT DEPEND ON S

  → $\uparrow$ (COMPLEXITY OF H / VC dim) $\rightarrow \downarrow \varepsilon_{APP}$

- **ESTIMATION ERROR** → (!) S → $\varepsilon_{EST} = L_D(h_S) - \varepsilon_{APP}$

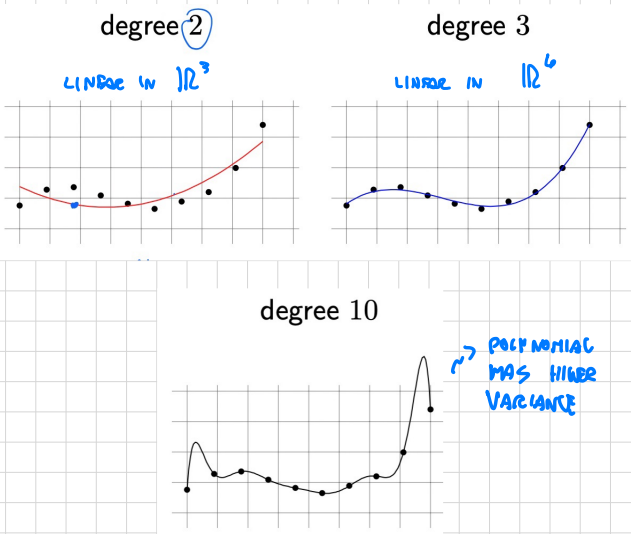
  → IT TELLS HOW GOOD AM I IN EXPLOITING H ON MY PROBLEM

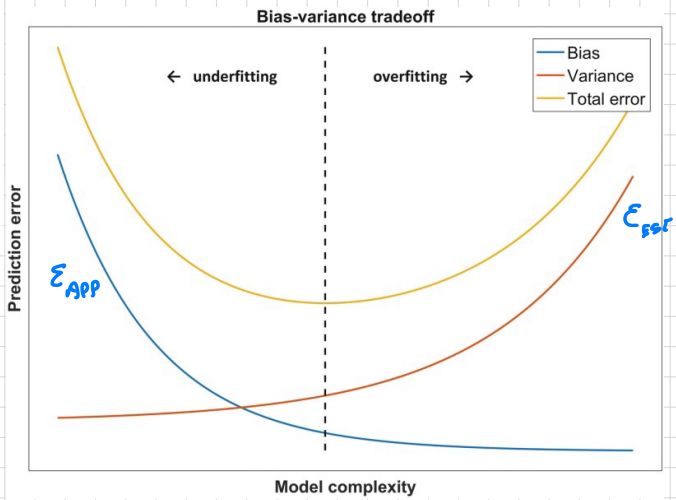
  → IT IS THE RESULT OF L_S BEING ONLY AN ESTIMATE OF L_D

  → $\uparrow$ SIZE OF S → $\downarrow \varepsilon_{EST}$

  → $\uparrow$ COMPLEXITY OF H → $\downarrow$ IT CANNOT FIT WELL OTHER DATA NOT IN TRAINING SET → $\uparrow \varepsilon_{EST}$ ∵ $\uparrow$ COMPLEXITY → $\downarrow \varepsilon_{APP}$ → $\uparrow \varepsilon_{EST}$

- HOW TO CHOOSE H ? → BIAS - COMPLEXITY TRADE OFF





VALIDATION AND MODEL SELECTION :

· VALIDATION :

SUPPOSE WE ACTUALLY WANT SOME HYPOTHESIS h, AND WE WANT TO ESTIMATE HOW GOOD IT IS

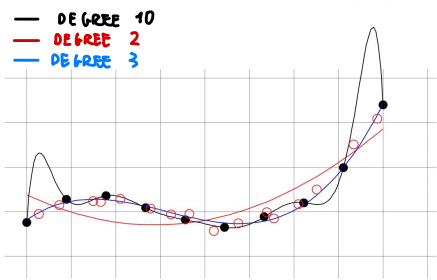
-> TAKE NEW iid SAMPLE $V = (x_1, y_1), \dots, (x_{m_v}, y_{m_v})$ (VALIDATION; NOT INVOLVED DURING TRAINING PROCESS OF h) / m_v : # SAMPLE IN VALIDATION

-> IT OUTPUTS $L_V(h)$ AS AN ESTIMATOR OF $L_D(h)$

· USING HOEFDING'S INEQUALITY, IF $\ell \in [0,1]$:

$$|L_V(h) - L_D(h)| \leq \sqrt{\frac{\log(2/\delta)}{2 \cdot m_v}}$$

· VALIDATION FOR MODEL SELECTION :



ex.

LET : o VALIDATION POINTS, ● TRAINING POINTS

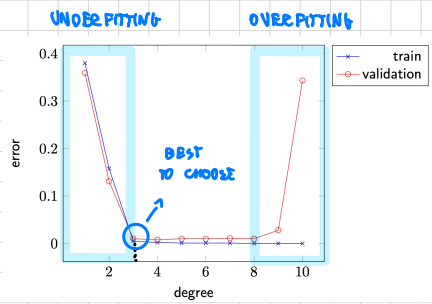
-> DEGREE 3 POLYNOMIAL HAS MINIMAL VALIDATION ERROR

LET $H = \{h_1, \dots, h_r\}$ BE THE OUTPUT PREDICTORS OF APPLYING ERM W.R.T. THE DIFFERENT CLASSES ON S

· LET V BE A NEW VALIDATION SET, CHOOSE $h^* \in ERM_H(V)$

$$\rightarrow L_D(h^*) \leq \min_{h \in H} L_D(h) + \sqrt{\frac{2\log(2|H|/\delta)}{|V|}}$$

· TRAIN - VALIDATION - TEST SPLIT :



· TRAINING SET : APPLY LEARNING ALGORITHM WITH DIFFERENT PARAMETERS ON TRAINING SET TO PRODUCE $H = \{h_1, \dots, h_n\}$

· VALIDATION SET : CHOOSE h^* FROM H BASED ON VALIDATION SET ~> TUNING AND SELECTION

· TEST SET : ESTIMATE ERROR OF h^* USING TEST SET

-> K-FOLD IS A GOOD PROCEDURE

SVM AND KERNEL METHODS:

SVM:

HALF-SPACE CLASSIFIER:

$$x \longmapsto \text{sign}\left(\langle \bar{w}, \bar{x} \rangle + b\right) = \pm 1 \quad , \quad \bar{w} \in \mathbb{R}^d$$

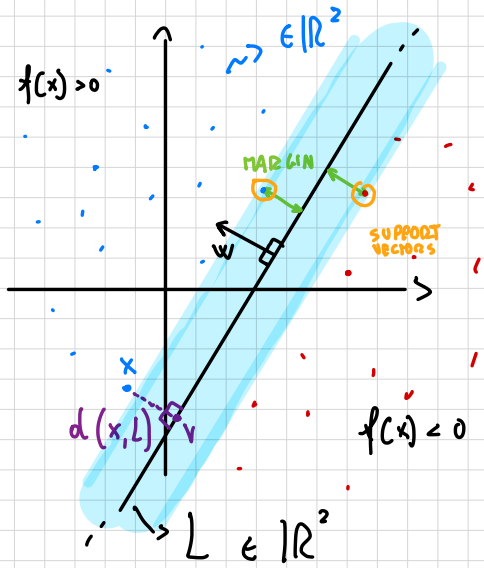
- MARGIN:

GIVEN THE HYPERPLANE $L = \{v : \langle w, v \rangle + b = 0\}$ (SET OF POINTS OF HYPERPLANE)

- GIVEN $\bar{x}$ (→ IN ES. $\in \mathbb{R}^2$, IN GENERAL $\in \mathbb{R}^d$), DISTANCE OF x TO L:

$$d\left(\bar{x}, L\right) = \min\left\{\| \bar{x} - \bar{v} \| : \bar{v} \in L\right\}$$

↳ IF $\|w\| = 1$ → $d\left(\bar{x}, L\right) = \left| \langle \bar{w}, \bar{x} \rangle + b \right|$



es. IN $\mathbb{R}^3$ ($\bar{w} = (a, b, c)$)

PLANE $\tilde{\pi}: ax + by + cz + d = 0$, POINT $P = (x_p, y_p, z_p)$

$$\rightarrow d\left(P, \tilde{\pi}\right) = \frac{\left| \langle \bar{w}, \bar{P} \rangle + d \right|}{\|w\|} = \frac{\left| a x_p + b y_p + c z_p + d \right|}{\sqrt{a^2 + b^2 + c^2}}$$

- MARGIN: $\min_i \left| \langle \bar{w}, \bar{x}_i \rangle + b \right|$ ⇝ CLOSEST DISTANCE FROM L TO x_i, $\forall i$

- SUPPORT VECTORS:

A SEPARATING HYPERPLANE IS DEFINED BY $\bar{w}, b$ / $\forall i, y_i \cdot \left(\langle \bar{w}, x_i \rangle + b\right) > 0$

$f(x)$: >0 → CLASS $+1$; <0 → CLASS -1

→ THE CLOSEST EXAMPLES x_i ARE CALLED SUPPORT VECTORS

· HARD SVM :

SEEK FOR THE SEPARATING HYPERPLANE WITH LARGEST MARGIN :

FIND $\bar{w}, b$ TO MAXIMIZE min DISTANCE

$$\underset{(\bar{w},b):\|\bar{w}\|=1}{\arg\max} \left(\underbrace{\min_{i\in[m]} \underbrace{|\langle \bar{w}, \bar{x}_i \rangle + b|}_{\geq 0}}_{\text{CLOSEST DISTANCE } L \text{ TO } x_i} \right) \Big/ \underbrace{\forall i, \; y_i \left(\langle \bar{w}, x_i \rangle + b \right) > 0}_{\text{2 CLASSES SEPARATION}}$$

· OR EQUIVALENTLY :

$$\underset{(\bar{w},b):\|\bar{w}\|=1}{\arg\max} \left(\min_{i\in[m]} \underbrace{y_i \left(\langle \bar{w}, \bar{x}_i \rangle + b \right)}_{\geq 0} \right)$$

· OR EQUIVALENTLY : BEST PLANE : $\langle \bar{w}_0, \bar{x} \rangle + b_0$

$$(\bar{w}_0, b_0) = \underset{\bar{w}, b}{\arg\min} \left(\|\bar{w}\|^2 \right) \Big/ \forall i, \; y_i \left(\langle \bar{w}, x_i \rangle + b \right) \geq 1$$

$\rightarrow$ MARGIN OF $\left(\frac{\bar{w}_0}{\|\bar{w}_0\|}, \frac{b}{\|\bar{w}_0\|} \right) = \frac{1}{\|w_0\|}$ $\rightsquigarrow$ MAXIMAL MARGIN

EX.

$$\tilde{\pi} : \frac{1}{\sqrt{13}} (2x + 3y - 3) = 0 \rightarrow \left(\frac{(2,3)}{\sqrt{13}}, \frac{-3}{\sqrt{13}} \right)$$

$\rightarrow$ MAXIMAL MARGIN $= \frac{1}{\|w_0\|} = \frac{1}{\sqrt{13}} \left(= \frac{1}{\sqrt{2^2+3^2}} \right)$

· MARGIN IS SCALE SENSITIVE :

IF $(\bar{w}, b)$ SEPARATES x_i WITH MARGIN γ $\rightarrow$ $\alpha \cdot x_i$ SEPARABLE BY $\alpha \cdot \gamma$

· MARGIN OF DISTRIBUTION :

D IS SEPARABLE WITH A (γ, ρ) - MARGIN, IF $\exists \; (\bar{w}^*, b^*) / \|\bar{w}^*\| = 1$

AND :

$$D\left(\{ (\bar{x}, y) : \underbrace{\|\bar{x}\| \leq \rho}_{\text{POINTS IN A FINITE SPACE}} \cap \underbrace{y \left(\langle \bar{w}^*, x \rangle + b^* \right) \geq 1}_{\text{2 CLASS SEPARATION}} \right) = 1$$

$\rightarrow$ THEOREM :

IF D SEPARABLE WITH (γ, ρ) - MARGIN $\rightarrow$ $m(\varepsilon, \delta) \leq \frac{8}{\varepsilon^2} \left(2 \left(\frac{\rho}{\gamma} \right)^2 + \log\left(\frac{2}{\delta} \right) \right)$ SAMPLE COMPLEXITY

$\hookrightarrow$ UNLIKE VC BOUNDS, m (.) $\frac{\rho}{\gamma}$, NOT ON d

- **SOFT SVM :**

  - HARD SVM ASSUMES DATA ARE LINEARLY SEPARABLE

  -> IF NOT SEPARABLE -> SOFT-SVM : RELAX CONSTRAINT

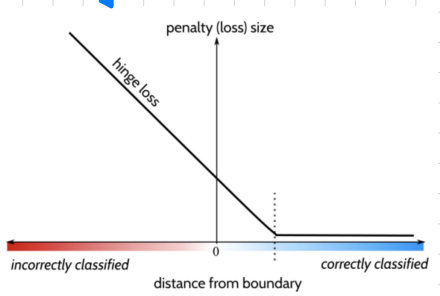
$$\underset{\bar{w}, b, \xi}{\operatorname{argmin}} \left(\lambda \cdot \|\bar{w}\|^2 + \underbrace{\frac{1}{m} \sum_{i=1}^{m} \xi_i}_{L_S^{\text{HINGE}}} \right) \Big/ \forall i,\ y_i \left(\langle \bar{w}, x_i \rangle + b \right) \geq 1 - \xi_i \ , \ \xi_i \geq 0$$

  - GIVEN THE HINGE LOSS $\ell^{\text{HINGE}}\left((\bar{w}, b), (\bar{x}, y) \right) = \max \left\{ 0, \underbrace{1 - y \left(\langle \bar{w}, \bar{x} \rangle + b \right)}_{\xi} \right\}$

  , IT CAN BE WRITTEN AS REGULARIZED LOSS MINIMIZATION :

$$\underset{\bar{w}, b}{\operatorname{argmin}} \left(\lambda \cdot \|\bar{w}\|^2 + L_S^{\text{HINGE}}\left((\bar{w}, b) \right) \right)$$

↳ FOR WHOLE DATASET



  - REGULARIZED LOSS MINIMIZATION (RLM):

    GIVEN A REGULARIZATION FUNCTION $R : \mathbb{R}^d \longmapsto \mathbb{R}$:

$$\text{RLM RULE} : A(S) = \underset{\bar{w}}{\operatorname{argmin}} \left(L_S(\bar{w}) + R(\bar{w}) \right)$$

b TERM CAN BE REMOVED IF A NEW FEATURE $x_{d+1} = 1$ IS ADDED

    - TIKHONOV / RIDGE REGULARIZATION : $R(\bar{w}) = \lambda \|\bar{w}\|^2$

- **STABILITY :**

INFORMALLY, AN ALGORITHM A IS STABLE IF A SMALL CHANGE ON THE INPUTS S WILL LEAD TO A SMALL CHANGE OF ITS OUTPUTS HYPOTHESIS

  - REPLACE 1 SAMPLE : LET $S^{(i)} = (z_1, \dots, z_{i-1}, z', z_{i+1}, \dots, z_m)$ (REPLACED SAMPLE, POSITION i) $\Big/ S = (z_1, \dots, z_m)$ (ORIGINAL DATA)

  - ON-AVERAGE-REPLACE-ONE-STABLE DEFINITION :

LET $\varepsilon : \mathbb{N} \longmapsto \mathbb{R}$ BE A MONOTONICALLY DECREASING FUNCTION.

  - ALGORITHM A IS ON-AVERAGE-REPLACE-ONE-STABLE WITH RATE $\varepsilon(m)$ IF $\forall$ DISTRIBUTION D :

$$\underset{\substack{(S, z') \sim D^{m+1} \\ i \sim U(m)}}{E} \left[\ell\left(A(S^{(i)}), z_i \right) - \ell\left(A(S), z_i \right) \right] \leq \varepsilon(m)$$

- **THEOREM:**

IF A IS ON-AVERAGE-REPLACE-ONE-STABLE WITH RATE $\varepsilon(m)$:

$$\rightarrow \mathop{E}_{S\sim D^m}\left[L_D\left(A(S)\right) - L_S\left(A(S)\right)\right] \leq \varepsilon(m)$$

(L_D: LOSS BY TRUE DISTRIBUTION D OF DATA; L_S: LOSS BY TRAINING DATA)

- **THEOREM:**

TIKHONOV REGULARIZATION AS STABILIZER: f PROPERTY: $|f(x_1) - f(x_2)| \leq \rho \cdot |x_1 - x_2|$

LIKE: ↑

ASSUME THAT THE LOSS FUNCTION IS CONVEX AND ρ-LIPSCHITZ.

→ RLM RULE, WITH REGULARIZER $\lambda \|w\|^2$, IS ON-AVERAGE-REPLACE-ONE-STABLE WITH RATE $\varepsilon(m) = \frac{2\rho^2}{\lambda m}$, AND:

$$\mathop{E}_{S\sim D^m}\left[L_D\left(A(S)\right) - L_S\left(A(S)\right)\right] \leq \underbrace{\frac{2\rho^2}{\lambda m}}_{\text{RATE } \varepsilon(m)}$$

λ: RAPID CHANGES ↗ (.) β

- SIMILARLY FOR CONVEX, β-SMOOTH, NON-NEGATIVE LOSS

RATE $\varepsilon(m) = \frac{48\beta C}{\lambda m}$ / C: UPPER BOUND ON $\max_z \ell(\bar{0}, z)$

- **FITTING-STABILITY TRADEOFF:**

$$\mathop{E}_{S}\left[L_D\left(A(S)\right)\right] = \underbrace{\mathop{E}_{S}\left[L_S\left(A(S)\right)\right]}_{1^{ST}\text{ TERM: HOW GOOD } A \text{ FITS } S} + \underbrace{\mathop{E}_{S}\left[L_D\left(A(S)\right) - L_S\left(A(S)\right)\right]}_{2^{ND}\text{ TERM: OVERFITTING}}$$

(L_D: TRUE RISK; L_S: TRAINING ERROR)

→ λ CONTROLS TRADEOFF BETWEEN THE 2 TERMS: $\leq \varepsilon(m) = \frac{2\rho^2}{\lambda m}$

$\uparrow \lambda \rightarrow \downarrow \varepsilon(m) \rightarrow \downarrow$ OVERFITTING

- FIX SOME ARBITRARY VECTOR $\bar{w}^*$:

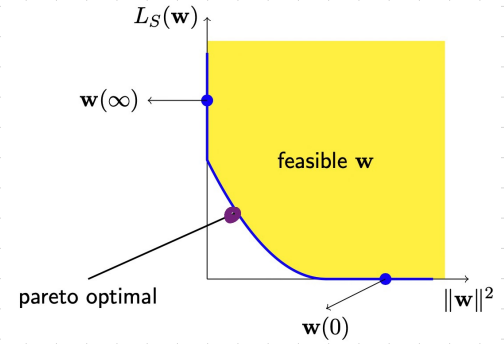
$$L_S\left(A(S)\right) \leq L_S\left(A(S)\right) + \lambda\|A(S)\|^2 \leq L_S(\bar{w}^*) + \lambda\|\bar{w}^*\|^2$$

TAKING $E[\cdot]$ ON BOTH SIDES? / $E_S[L_S(\bar{w}^*)] = L_D(\bar{w}^*)$

→ WE CAN OBTAIN: $\mathop{E}_{S}\left[L_S\left(A(S)\right)\right] \leq L_D(\bar{w}^*) + \lambda\|\bar{w}^*\|^2$

$$\rightarrow \mathop{E}_{S}\left[L_D\left(A(S)\right)\right] \leq L_D(\bar{w}^*) + \lambda\|\bar{w}^*\|^2 + \frac{2\rho^2}{\lambda m}$$

- **THE REGULARIZATION PATH :**



  - RLM RULE :

$$\bar{w}(\lambda) = \underset{\bar{w}}{\operatorname{argmin}} \left(L_S(\bar{w}) + \lambda \|\bar{w}\|^2 \right)$$

  - → WE WANT TO MINIMIZE BOTH $L_S(\bar{w})$ AND $\|\bar{w}\|^2$:
    - $\bar{w}(\lambda = 0)$: MINIMIZE JUST $L_S(\bar{w})$
    - $\bar{w}(\lambda = \infty)$: MINIMIZING $\|\bar{w}\|^2$ AS AN HIGHER IMPACT : $L_S(\bar{w}) \xrightarrow{\lambda \|\bar{w}\| \gg 0} 0$

  - HOW TO CHOOSE λ ?
    - → BOUND MINIMIZATION : CHOOSE λ ACCORDING TO THE BOUND ON $L_D(\bar{w})$ USUALLY FAR FROM OPTIMAL, AS THE BOUND IS WORST CASE
    - → VALIDATION : CALCULATE SEVERAL PARETO OPTIMAL POINTS ON THE REGULARIZATION PATH (BY VARYING λ) AND USE VALIDATION SET TO CHOOSE THE BEST ONE

- **SAMPLE COMPLEXITY OF SOFT-SVM.**

  - SOFT-SVM $\equiv$ RLM
  - HINGE-LOSS : $\bar{w} \mapsto \max\{0, 1 - y\langle \bar{w}, \bar{x} \rangle\}$ IS $\|\bar{x}\|$-LIPSHITZ : ($\rho = \|\bar{x}\|$)

$$\|\bar{x}\|\text{-LIPSHITZ} \rightarrow \left| \max\{0, 1 - y\langle \bar{w}, x \rangle\} - \max\{0, 1 - y\langle \bar{w}', \bar{x} \rangle\} \right| \leq \|\bar{x}\| \cdot \|\bar{w} - \bar{w}'\|$$

  - WITH MIN-MAX SCALER : $\dfrac{\bar{x} - \min(\bar{x})}{\max(\bar{x}) - \min(\bar{x})} \rightarrow$ DATA $\subset [0,1]^d \rightarrow \exists \rho \;/\; \|\bar{x}\| < \rho$

  - ASSUME D SUCH THAT $\|\bar{x}\| < \rho$ WITH PROBABILITY $= 1$

→ WE OBTAIN A CONVEX-LIPSHITZ LOSS, AND $\forall \bar{u}$:

$$\underset{S \sim D^m}{E} \left[L_D^{\text{HINGE}}(A(S)) \right] \leq L_D^{\text{HINGE}}(\bar{u}) + \lambda \|\bar{u}\|^2 + \frac{2\rho^2}{\lambda m}$$

  - IF WE SET $\lambda = \frac{1}{B}\sqrt{\frac{2\rho^2}{m}}$ ($\forall B > 0$, CONST.), AND SINCE ℓ^{HINGE} IS AN UPPER BOUND FOR 0-1 LOSS :

$$\rightarrow \underset{S \sim D^m}{E} \left[L_D^{0-1}(A(S)) \right] \leq \min_{\bar{w} : \|\bar{w}\| < B} \left\{ L_D^{\text{HINGE}}(\bar{w}) \right\} + \sqrt{\frac{8\rho^2 B^2}{m}}$$

- MARGIN / NORM VS DIMENSIONALITY :
  - VC DIMENSION OF LEARNING HALFSPACES (·) DIMENSION OF $\sim$ (SEE GPT BOOK MARKS FOR EXAMPLES -> NOTION)
    -> $\uparrow$ of -> $\uparrow$ SAMPLE COMPLEXITY
  - FOR SVMs : SAMPLE COMPLEXITY (·) $\left(\frac{B}{\gamma}\right)^2 = \left(B^2 \rho^2\right)$

SGD for solving Soft-SVM (NOT COVERED ?)

goal: Solve $\operatorname{argmin}_{\mathbf{w}} \left(\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m} \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}_i\rangle\}\right)$
parameter: T
initialize: $\boldsymbol{\theta}^{(1)} = \mathbf{0}$
for $t = 1, \ldots, T$
  Let $\mathbf{w}^{(t)} = \frac{1}{\lambda t}\boldsymbol{\theta}^{(t)}$
  Choose i uniformly at random from $[m]$
  If $(y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle < 1)$
    Set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + y_i\mathbf{x}_i$
  Else
    Set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$
output: $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}^{(t)}$

KERNELS :

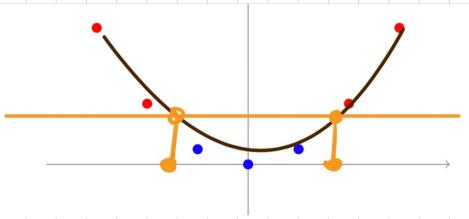
EMBEDDING INTO FEATURE SPACE :

ex.

→ THIS SAMPLE $\in \mathbb{R}^1$ IS NOT SEPARABLE BY HALFSPACES

BUT...

... IF WE MAP $x \longmapsto (x, x^2)$: ✓



GENERAL APPROACH :

1. DEFINE $\psi : X \longmapsto F \;/\; F \overset{\sim}{\to}$ SUBSET OF HILBERT SPACE : GENERALIZATION OF EUCLIDEAN SPACE IS SOME FEATURE SPACE
2. TRAIN A HALFSPACE OVER $(\psi(\bar{x}_1), y_1), \dots, (\psi(\bar{x}_m), y_m)$

POLYNOMIAL MAPPINGS :

DEGREE k

$$p(x) = \sum_{j=0}^{k} w_j x^j = \langle \bar{w}, \psi(x) \rangle \;/\; \psi(x) = (1, x, x^2, \dots, x^k)$$

- MORE GENERALLY, A DEGREE k MULTIVARIATE POLYNOMIAL FROM $\mathbb{R}^m \mapsto \mathbb{R}$:

$$p(\bar{x}) = \sum_{\substack{J \in [n]^r \\ r \le k}} w_J \cdot \prod_{i=1}^{r} x_{J_i} = \langle \bar{w}, \psi(\bar{x}) \rangle \;/\; \psi : \mathbb{R}^m \mapsto \mathbb{R}^d$$

ex. $J \in [n]^r$, $n = 3$, $r = 2$:

$\to [3]^2 \to [3] \times [3] =$

	x_1	x_2	x_3
x_1	$(1,1)$	$(1,2)$	$(1,3)$
x_2	$(2,1)$	$(2,2)$	$(2,3)$
x_3	$(3,1)$	$(3,2)$	$(3,3)$

$\to (x_1, x_2, x_3) \longmapsto (x_1^2, x_2^2, x_3^2, x_1x_2, x_2x_3, x_1x_3, x_1, x_2, x_3)$

KERNEL TRICK :

$$K(\bar{x}_1, \bar{x}_2) = \langle \psi(\bar{x}_1), \psi(\bar{x}_2) \rangle$$

IT CAN HELP IN AVOID TO EXPLICITLY COMPUTE $\psi(x), \psi(y)$ AND USE THE SHORTCUT $K(\bar{x}_1, \bar{x}_2)$

→ SOMETIMES CALCULATING $K(\cdot)$ IS EASIER AND MORE EFFICIENT

ex.

Simple Example: $x = (x1, x2, x3)$; $y = (y1, y2, y3)$. Then for the function $f(x) = (x1x1, x1x2, x1x3, x2x1, x2x2, x2x3, x3x1, x3x2, x3x3)$, the kernel is $K(x, y) = (\langle x, y \rangle)^2$.

EXPLICIT CALCULUS →

Let's plug in some numbers to make this more intuitive: suppose x = (1, 2, 3); y = (4, 5, 6). Then:
f(x) = (1, 2, 3, 2, 4, 6, 3, 6, 9)
f(y) = (16, 20, 24, 20, 25, 30, 24, 30, 36)
<f(x), f(y)> = 16 + 40 + 72 + 40 + 100+ 180 + 72 + 180 + 324 = 1024

USING $K(x, y)$ →

Now let us use the kernel instead:
K(x, y) = (4 + 10 + 18) ^2 = 32² = 1024
Same result, but this calculation is so much easier.

• THE REPRESENTER THEOREM:

CONSIDER ANY LEARNING RULE, $f : \mathbb{R}^m \mapsto \mathbb{R}$, OF THE FORM:

$$\bar{w}^* = \arg\min_{\bar{w}} \left\{ f\left(\langle \bar{w}, \psi(\bar{x}_1) \rangle, \dots, \langle \bar{w}, \psi(\bar{x}_m) \rangle \right) + \lambda \|\bar{w}\|^2 \right\}$$

$$\rightarrow \exists\, \bar{\alpha} \in \mathbb{R}^m \;/\; \bar{w}^* = \sum_{i=1}^{m} \alpha_i \cdot \psi(\bar{x}_i) \rightsquigarrow \bar{w}^* \text{ IS A LINEAR COMBINATION OF } \psi(\bar{x}_i)$$

• IMPLICATIONS OF THEOREM:

• DENOTE BY G (GRAM MATRIX) THE MATRIX / $G_{i,j} = \langle \psi(\bar{x}_i), \psi(\bar{x}_j) \rangle$, $\forall i$: (POSITIVE $(\lambda_i > 0)$, SEMI-DEFINED $(\lambda_i \geq 0)$)

$$\rightarrow \langle \bar{w}, \psi(\bar{x}_i) \rangle \overset{\text{REPR. THEOREM}}{=} \left\langle \sum_j \alpha_j \psi(\bar{x}_j), \psi(\bar{x}_i) \right\rangle =$$

$$\langle \bar{w}, \psi(\bar{x}_i) \rangle = (G\bar{\alpha})_i = \sum_{j=1}^{m} \alpha_j \cdot \langle \psi(\bar{x}_j), \psi(\bar{x}_i) \rangle = \underbrace{(G \cdot \bar{\alpha})_i}_{\in \mathbb{R}}$$

$G_{i,j} = K(\bar{x}_i, \bar{x}_j)$

ex. $\{x_1, x_2\}$

$$G = \begin{pmatrix} \langle \psi(\bar{x}_1), \psi(\bar{x}_1) \rangle & \langle \psi(\bar{x}_1), \psi(\bar{x}_2) \rangle \\ \langle \psi(\bar{x}_2), \psi(\bar{x}_1) \rangle & \langle \psi(\bar{x}_2), \psi(\bar{x}_2) \rangle \end{pmatrix} = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix}$$

$\langle \psi(\bar{x}_1), \psi(\bar{x}_1) \rangle = \|\psi(\bar{x}_1)\|^2$ → ANALOG FOR $i,j = 2$

$\langle \bar{w}, \bar{w} \rangle =$

$$\rightarrow \|\bar{w}\|^2 = \left\langle \sum_j \alpha_j \psi(\bar{x}_j), \sum_j \alpha_j \psi(\bar{x}_j) \right\rangle =$$

$$= \sum_{i,j=1}^{m} \alpha_i \cdot \alpha_j \cdot \langle \psi(\bar{x}_i), \psi(\bar{x}_j) \rangle = \bar{\alpha}^T \cdot G \cdot \bar{\alpha}$$

$\|\bar{w}\|^2 = \bar{\alpha}^T G \bar{\alpha}$

$$\left(= \sum_{i=1}^{m} \alpha_i G_{i,i} + 2 \sum_{i<j} \alpha_i \alpha_j G_{i,j} \right) \rightsquigarrow \text{QUADRATIC FORM}$$

→ SINCE $\langle \bar{w}, \psi(\bar{x}) \rangle = G \cdot \bar{\alpha}$, WE CAN OPTIMIZE OVER $\bar{\alpha}$:

$$\arg\min_{\bar{\alpha} \in \mathbb{R}^m} \left\{ f(G \cdot \bar{\alpha}) + \lambda\, \bar{\alpha}^T G \bar{\alpha} \right\}$$

• <u>N.B.</u>: G ONLY DEPENDS ON $\langle \cdot, \cdot \rangle$ → IT CAN BE COMPUTED USING $K(\cdot, \cdot)$ ALONE

→ SUPPOSE WE FOUND $\bar{\alpha}$, THEN GIVEN A NEW INSTANCE $\bar{x}$:

$$\langle \bar{w}, \psi(\bar{x}) \rangle \overset{\bar{x}: \text{NEW INSTANCE}}{=} \left\langle \sum_j \psi(\bar{x}_j), \psi(\bar{x}) \right\rangle = \sum_j \langle \psi(\bar{x}_j), \psi(\bar{x}) \rangle =$$

$$= \sum_j K(\bar{x}_j, \bar{x}) \rightsquigarrow \text{TRAINING AND PREDICTIONS CAN BE DONE USING } K(\cdot,\cdot) \text{ ALONE}$$

- REPRESENTER THEOREM FOR SVM:

  - SOFT-SVM: $\min_{\bar{\alpha} \in \mathbb{R}^m} \left\{ \underbrace{\lambda \|\bar{w}\|^2}_{\lambda \bar{\alpha}^T G \bar{\alpha}} + \underbrace{L_S^{\text{HINGE}}}_{\frac{1}{m}\sum_{i=1}^{m} \max\{0, 1 - y_i (G\bar{\alpha})_i\}} \right\}$

  - HARD-SVM: $\min_{\bar{\alpha} \in \mathbb{R}^m} \left\{ \underbrace{\|\bar{w}\|^2}_{\bar{\alpha}^T G \bar{\alpha}} \right\} \; / \; \forall i, \; y_i (G\bar{\alpha})_i \geq 1$

- POLYNOMIAL KERNELS:

  K DEGREE POLYNOMIAL KERNEL: $K(\bar{x}_1, \bar{x}_2) = \left(1 + \langle \bar{x}_1, \bar{x}_2 \rangle\right)^K$ (K: DEGREE OF POLYNOMIAL USED TO TRANSFORM $\bar{x}_1, \bar{x}_2$)

  ex. $K = 2$

  $$K(\bar{x}_1, \bar{x}_2) = \left(1 + \langle \bar{x}_1, \bar{x}_2 \rangle\right)^2 = \left(1 + \underbrace{x_{1,1} \cdot x_{2,1}}_{a} + \underbrace{x_{1,2} \cdot x_{2,2}}_{b}\right)^2 =$$

  ($x_{1,1}$: ELEMENT OF VECTOR $\bar{x}_1$, POSITION 1)

  $$\left(= 1^2 + a^2 + b^2 + 2 \cdot 1 \cdot a + 2 \cdot 1 \cdot b + 2 \cdot a \cdot b = 1 + a^2 + b^2 + 2a + 2b + 2ab\right)$$

  $$= 1 + \left(x_{1,1} \cdot x_{2,1}\right)^2 + \left(x_{1,2} \, x_{2,2}\right)^2 + 2 \cdot x_{1,1} x_{2,1} + 2 \cdot x_{1,2} x_{2,2} + 2 \cdot x_{1,1} x_{2,1} \cdot x_{1,2} x_{2,2}$$

  - N.B.: CALCULATING $K(\bar{x}_1, \bar{x}_2)$ TAKES $O(n)$ TIME, WHILE DIMENSION OF $\psi(\bar{x})$ IS n^K ~> n: # SAMPLES

- GAUSSIAN KERNELS (RBF):

  $$K(\bar{x}_1, \bar{x}_2) = e^{-\frac{\|\bar{x}_1 - \bar{x}_2\|^2}{2\sigma}}$$ → IT CAN LEARN ANY POLYNOMIAL

  - LET ORIGINAL INSTANCE SPACE BE $\mathbb{R}$, MAPPING ψ / $\exists \; \psi(x)_n = \frac{1}{\sqrt{n!}} e^{-\frac{x^2}{2}} \cdot x^n$ (SINGLE ELEMENT, $n \in \mathbb{Z}$, $n \geq 0$)

  $$\rightarrow \langle \psi(x_1), \psi(x_2) \rangle = \sum_{n=0}^{\infty} \left(\frac{1}{\sqrt{n!}} e^{-\frac{x_1^2}{2}} \cdot x_1^n\right)\left(\frac{1}{\sqrt{n!}} e^{-\frac{x_2^2}{2}} \cdot x_2^n\right) =$$

  $$= e^{-\frac{x_1^2 + x_2^2}{2}} \cdot \sum_{n=0}^{\infty} \frac{(x_1 x_2)^n}{n!} = e^{-\frac{(x_1 - x_2)^2}{2}}$$

  - LEMMA (MERCER'S CONDITION):

    A SYMMETRIC FUNCTION $K: X \times X \mapsto \mathbb{R}$ IMPLEMENTS AN INNER PRODUCT IN SOME HILBERT SPACE $\iff \forall \bar{x}_1, \ldots, \bar{x}_m$: $G_{i,j} = K(\bar{x}_i, \bar{x}_j)$ IS A POSITIVE, SEMI-DEFINITE MATRIX

- **FRITZ-JOHN CONDITION :**

THEOREM 15.8 *Let $\mathbf{w}_0$ be as defined in Equation (15.3) and let $I = \{i : |\langle \mathbf{w}_0, \mathbf{x}_i \rangle| = 1\}$. Then, there exist coefficients $\alpha_1, \ldots, \alpha_m$ such that*

$$\mathbf{w}_0 = \sum_{i \in I} \alpha_i \mathbf{x}_i .$$

The examples $\{\mathbf{x}_i : i \in I\}$ are called *support vectors*.

The proof of this theorem follows by applying the following lemma to Equation (15.3).

LEMMA 15.9 (Fritz John) *Suppose that*

$$\mathbf{w}^\star \in \operatorname*{argmin}_{\mathbf{w}} f(\mathbf{w}) \quad s.t. \quad \forall i \in [m],\ g_i(\mathbf{w}) \leq 0,$$

where $f, g_1, \ldots, g_m$ are differentiable. Then, there exists $\boldsymbol{\alpha} \in \mathbb{R}^m$ such that $\nabla f(\mathbf{w}^\star) + \sum_{i \in I} \alpha_i \nabla g_i(\mathbf{w}^\star) = \mathbf{0}$, where $I = \{i : g_i(\mathbf{w}^\star) = 0\}$.

- **DUALITY :**

  - FOR HARD-SVM, FOR EXAMPLE :

$$\min_{\bar{w}} \|\bar{w}\|^2 \quad / \quad \forall i,\ y_i \langle \bar{w}, \bar{x}_i \rangle \geq 1$$

IT CAN BE WRITTEN AS

$$\min_{\bar{w}} \left\{ \max_{\substack{\bar{\alpha} \in \mathbb{R}^m \\ / \bar{\alpha} \geq \bar{0}}} \left\{ \frac{1}{2}\|\bar{w}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \langle \bar{w}, \bar{x}_i \rangle\right) \right\} \right\}$$

LAGRANGIAN MULTIPLIERS — CONDITION

-> WE CAN OBTAIN THE WEAK DUALITY INEQUALITY :

$$\min_{\bar{w}} \left\{ \max_{\substack{\bar{\alpha} \in \mathbb{R}^m \\ / \bar{\alpha} \geq \bar{0}}} \left\{ \frac{1}{2}\|\bar{w}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \langle \bar{w}, \bar{x}_i \rangle\right) \right\} \right\} \geq \max_{\substack{\bar{\alpha} \in \mathbb{R}^m \\ / \bar{\alpha} \geq \bar{0}}} \left\{ \min_{\bar{w}} \left\{ \frac{1}{2}\|\bar{w}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \langle \bar{w}, \bar{x}_i \rangle\right) \right\} \right\}$$

  - WE CAN SOLVE ANALYTICALLY THE OPTIMIZATION $\min \{ \ldots \}$:

↳ SOLUTION : $\bar{w} = \sum_{i=1}^{m} \alpha_i y_i \bar{x}_i$

-> WHICH LEADS TO : ~> PLUGGING IT BACK

$$\max_{\substack{\bar{\alpha} \in \mathbb{R}^m \\ / \bar{\alpha} \geq 0}} \left\{ \frac{1}{2} \left\| \sum_{i=1}^{m} \alpha_i y_i \bar{x}_i \right\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left\langle \sum_{j} \alpha_j y_j \bar{x}_j, \bar{x}_i \right\rangle\right) \right\}$$

- **CURSE OF DIMENSIONALITY:**

GIVEN n THE SPACE DIMENSION: $X \in \mathbb{R}^n$ → ex. $n=3$: $\bar{x} = (\overset{x_1}{a}, \overset{x_2}{b}, \overset{x_3}{c})$

① ↑ n → ↑ # OPERATIONS REQUIRED TO COMPUTE DISTANCES:

$$d(\bar{x}, \bar{r}) = \| \bar{x} - \bar{r} \| = \sqrt{\sum_i (x_i - r_i)^2} \quad / \quad \bar{x}, \bar{r} \in \mathbb{R}^n$$

ex. $n=3$: $d(\bar{x}, \bar{r}) = \left((x_1 - r_1)^2 + (x_2 - r_2)^2 + (x_3 - r_3)^2 \right)$ → 3 SUMS

ex. $n=100$: $d(\bar{x}, \bar{r}) = \left((x_1 - r_1)^2 + \dots + (x_{100} - r_{100})^2 \right)$ → 100 SUMS

② ↑ n → ↑ DISTANCES LENGTH:

$Q_d = [0,1]^d$, MAX DISTANCE $= \| (1, \dots, 1) - (0, \dots, 0) \| = \sqrt{d}$

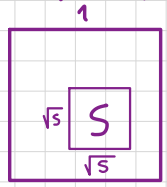
ex. $d=2$ ex. $d=3$ ex. $d=100$

..... → $\sqrt{100}$

→ ① AND ② CAN CAUSE PROBLEMS IN MANY MACHINE LEARNING PROBLEMS: DISTANCES FOR K-NN, ESTIMATION IN REGRESSIONS, ECC.

- **HIGH DIMENSIONAL SPACES ARE EMPTY:**

ex. $\mathbb{R}^2$: $p=2$ → $s^{1/2} = \sqrt{s}$

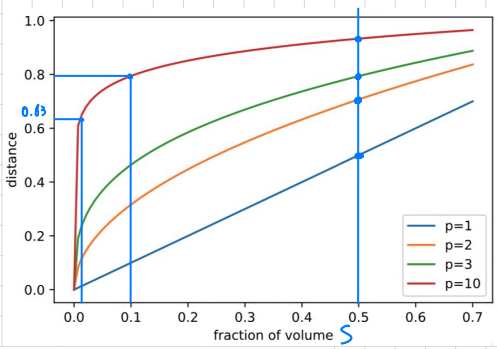


→ s IS A FRACTION OF $[0,1]^2$

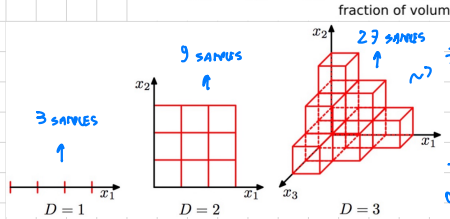
ASSUME DATA LIVES IN $[0,1]^p$.

TO CAPTURE A NEIGHBOURHOOD WHICH IS A FRACTION s OF THE HYPERCUBE VOLUME, EDGE LENGTH (OF HYPERCUBE) $= s^{1/p}$

- $s = 0.1$ (10% OF VOLUME), $p = 10$ → $s^{1/p} = 0.80$ (/1.26)
- $s = 0.01$ (1% OF VOLUME), $p = 10$ → $s^{1/p} = 0.63$ (/1.26)



- IN ORDER TO CAPTURE HIGHER s → AS ↑ p, A "LONGER" EDGE IS NEEDED, WHICH GROWS EXPONENTIALLY



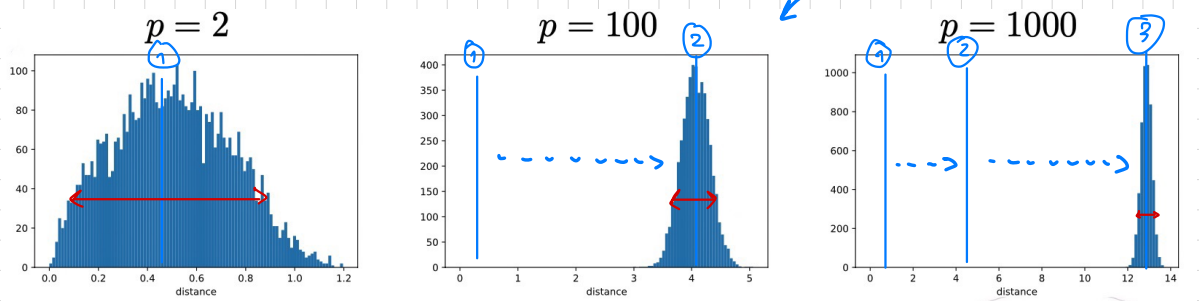
SAMPLES NEEDS TO GROW EXPONENTIALLY AS p GROWS LINEARLY IN ORDER TO CAPTURE SAME FRACTION OF THE SPACE

- **NEAREST NEIGHBOURS** :

GIVE 2 R.V. $X, Y \sim U([0,1]^p)$ (# DIMENSIONS)

- MEAN SQUARE DISTANCE $(MSD) = \|X - Y\|^2$:
  - $E[MSD] = \alpha = \frac{1}{6} p$
  - $STD[MSD] = \beta \simeq 0.2\sqrt{p}$

SCALED STD : $\frac{\beta}{\alpha} = \frac{\sigma}{\mu} \approx p^{-\frac{1}{2}}$



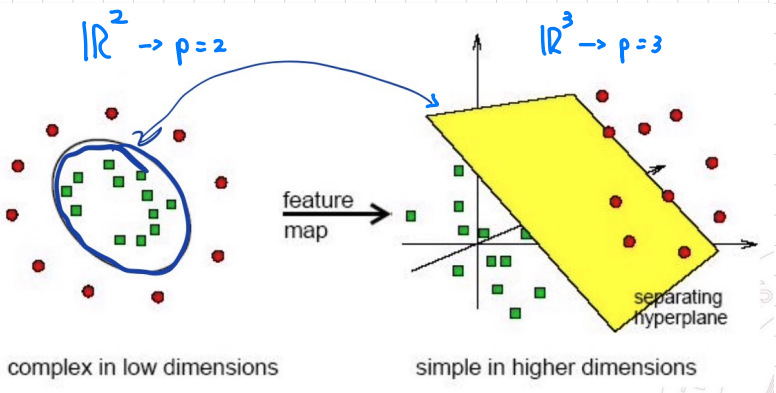
- $\uparrow p \longrightarrow \uparrow E[MSD]$, $\downarrow STD[MSD]$

  $\longrightarrow \uparrow E[MSD]$: ALL POINTS TENDS TO BE AT HIGHER DISTANCES FROM EACH OTHERS

  $\longrightarrow \downarrow STD[MSD]$: ALL POINTS TENDS TO HAVE ALMOST THE SAME DISTANCE FROM EACH OTHER

$\longrightarrow$ THE NOTION OF NN VANISHES : DIFFICULTY TO HAVE MEANINGFUL DISTANCES AND USE THEM

$\longrightarrow$ BUT $p >> 0$ CAN BE USEFUL TO SEPARATE GROUPS BETTER ;



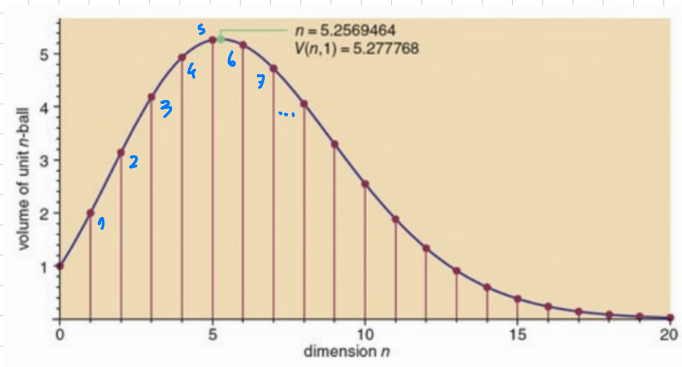
CONCENTRATION PHENOMENA :

VOLUME OF HYPER-BALL, RADIUS r :

$$V_p(r) = r^p \frac{\pi^{p/2}}{\Gamma\left(\frac{p}{2}+1\right)}$$

GAMMA FUNCTION :

$\Gamma(n+1) = n!$, $n \in \mathbb{N}$, $\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t}\, dt$, $z \in \mathbb{C}$



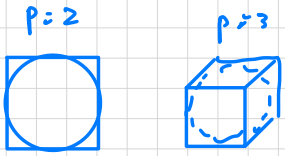
$r = 1$

-> STRANGE PHENOMENA :

AFTER $n = 5.25\ldots$

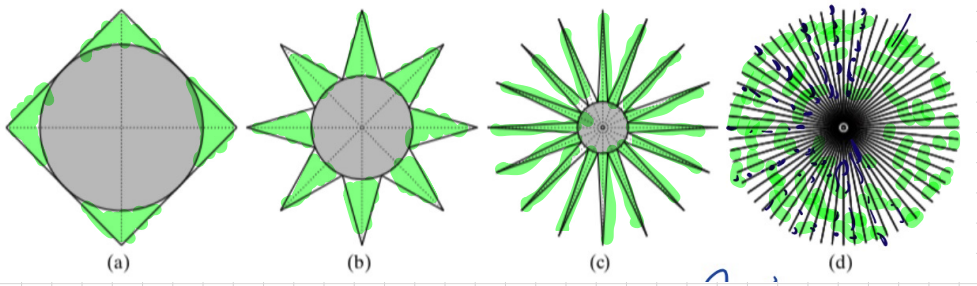
VOLUME STARTS DECREASING !

-> IF YOU WANT TO COVER $[0,1]^p$: ~>



. . .

MOST OF POINTS



- ASSUME YOU DRAW n SAMPLES WITH $\sim U$

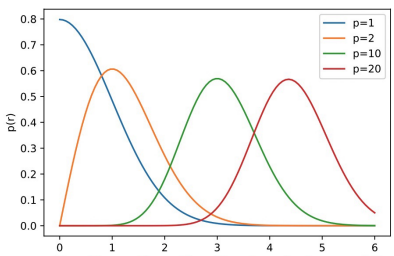
-> MOST SAMPLES WILL BE IN THE CORNERS OF HYPERCUBE

- PROBABILITY THAT $X \sim U([0,1]^p)$ BELONGS TO SHELL BETWEEN $r = 0.9$ AND $r = 1$:

$$P\left(X \in S_{0.9}(p)\right) = 1 - 0.9^p \xrightarrow[p \to \infty]{} 1$$

~> AS $\uparrow p$, IT'S ALMOST SURE THAT THE POINT WILL BE IN $0.9 < r < 1$

- FOR GAUSSIAN DISTRIBUTION :



Most of the mass of a Gaussian distribution is located in areas where the density is extremely small compared to its maximum value.

• DIMENSIONALITY REDUCTION :

DEF. : TAKING DATA IN HIGH DIMENSIONAL SPACE AND MAPPING IT INTO A LOW DIMENSIONAL SPACE

→ REDUCES TRAINING AND TESTING TIME

→ REDUCES ESTIMATION ERROR

→ INTERPRETABILITY OF DATA

$$\begin{bmatrix} \tilde{x} \end{bmatrix}_{d\times 1} \longmapsto \begin{bmatrix} W \end{bmatrix}_{n\times d,\ n<d} \cdot \begin{bmatrix} \tilde{x} \end{bmatrix} = \begin{bmatrix} \tilde{v} \end{bmatrix}_{n\times 1}$$

• LINEAR DIMENSIONALITY REDUCTION :

$$\tilde{x} \longmapsto W\tilde{x} = \tilde{v} \ / \ W \in \mathbb{R}^{n,d},\ n<d$$

• PCA (PRINCIPAL COMPONENT ANALYSIS) :

$$\tilde{x} \xrightarrow[n\times d]{W} W\tilde{x} = \tilde{v} \qquad / \qquad \tilde{x} = [x_1, x_2, \dots, x_d]^T$$

$\rightsquigarrow O(nd^2 + d^3)$

- NATURAL CRITERION : WE WANT TO BE ABLE TO APPROXIMATELY RECOVER $\tilde{x}$ FROM $\tilde{v} = W\tilde{x}$

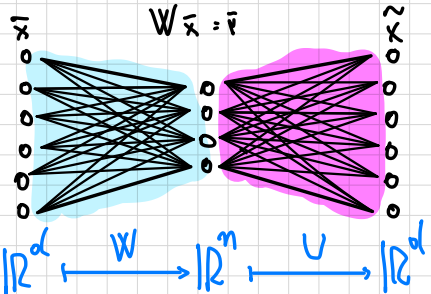
  - IN PCA : $\tilde{\tilde{x}} = U\tilde{v} = UW\tilde{x}$ (U: $d\times n$, W: $n\times d$) — LINEAR RECOVERY
  - MEASURES "APPROXIMATE RECOVERY", SOLVE :

$$\underset{W\in\mathbb{R}^{n,d},\ U\in\mathbb{R}^{d,n}}{\arg\min} \left\{ \sum_{i=1}^{m} \| \underbrace{\tilde{x}_i}_{\text{ORIGINAL FEATURES}} - \underbrace{UW\tilde{x}_i}_{\text{RECOVERED FEATURES}} \|_2^2 \right\} \rightsquigarrow m : \#\ \text{SAMPLES}$$

- IT IS A LOSSY PROCESS : $\tilde{\tilde{x}} \neq \tilde{x}$
- $U \cdot W \approx I_d$
- $\tilde{x}_{d\times 1} \xrightarrow{W} W\tilde{x} = \tilde{v}_{n\times 1} \xrightarrow{U} U\cdot\tilde{v} = \tilde{\tilde{x}}_{d\times 1}$ (W: $n\times d$, $n<d$; U: $d\times n$)

  ENCODING | DECODING

- IT CAN BE SEEN AS A MLP :



- **SOLVING THE PCA PROBLEM:**

$$\underset{W \in \mathbb{R}^{d,d},\, U \in \mathbb{R}^{d,n}}{\arg\min} \left\{ \sum_{i=1}^{m} \| \bar{x}_i - U W \bar{x}_i \|_2^2 \right\}$$

- **THEOREM:**

IF $\mu = 0$: LET $A = \sum_{i=1}^{m} \bar{x}_i \bar{x}_i^T$ ($d \times d$), LET $\bar{u}_1, \dots, \bar{u}_m$ BE THE

A (·) COVARIANCE MATRIX

n LEADING EIGENVECTORS OF A:

$$U = \left[\begin{matrix} | & | & & | \\ \bar{u}_1 & \bar{u}_2 & \cdots & \bar{u}_n \\ | & | & & | \end{matrix} \right]$$

→ SOLUTION OF PCA: $\begin{cases} \text{COLUMNS OF } U \text{ SET TO } \bar{u}_1, \dots, \bar{u}_n \\ W = U^T \end{cases}$

EIGENVALUE λ OF A: $\det(A - \lambda I)$

EIGENVECTORS $\bar{v}$ OF A: $(A - \lambda I) \cdot \bar{v} = 0$ (CORRESPONDING EIGENVALUE OF $\bar{v}$)

$(C \cdot D)^T = D^T C^T$

(i) PROPERTY: $A^T = \left(\sum_i \bar{x}_i x_i^T \right)^T = \sum_i \left(\bar{x}_i \cdot \bar{x}_i^T \right)^T = \sum_i \bar{x}_i \cdot \bar{x}_i^T = A \rightarrow A$ SYMMETRIC

FROM (i)

(ii) PROPERTY: IF A SYMMETRIC $\rightarrow \lambda_i \in \mathbb{R}$

PROPERTY: $\lambda_1 \geq \dots \geq \lambda_d \geq 0 \rightarrow A$ IS POSITIVE SEMI-DEFINITE

- **N.B.:** $\frac{1}{m-1} (\bar{x} - \mu)^T (\bar{x} - \mu) = \Sigma$ = COVARIANCE MATRIX

- **N.B.:** $\| \bar{x} - U U^T \bar{x} \|^2 = \dots = \| \bar{x} \|^2 - \text{trace}\left(U^T \bar{x} \bar{x}^T U \right)$ ([III] ≡ (22/80))

SUM $(a_{i,i})$: SUM OF ALL ELEMENTS ON DIAGONAL

→ EQUIVALENT PCA PROBLEM:

$$\underset{U \in \mathbb{R}^{d,n};\, U^T U = I}{\arg\max} \left\{ \text{trace}\left(U^T \left(\sum_{i=1}^{m} \bar{x}_i \bar{x}_i^T \right) U \right) \right\}$$

→ SOLUTION IS TO SET U TO BE THE LEADING EIGENVECTORS OF $A = \sum_{i=1}^{m} \bar{x}_i \bar{x}_i^T$

PCA AND SVD:

https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca

- PCA CAN BE PERFORMED BY < FINDING EIGENVECTORS OF Σ (COVARIANCE MATRIX) / SVD ON X ~> DATA MATRIX

RECALL: $XX^T = [||]$ $(X^TX)^T = [||]$

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

/ U: COLUMNS ARE EIGENVECTORS OF $X\cdot X^T$

S: DIAGONAL MATRIX, $S_{ii} \geq 0$ ($= \sqrt{\lambda_i}$ OF X^TX)

V: ROWS ARE EIGENVECTORS OF X^TX

- IN PCA: $\underset{d\times d}{\Sigma} = \frac{1}{n-1}X^TX$, ASSUMING DATA IN X ARE ZERO-CENTERED ($X \mapsto X - \bar{\mu}$)

- Σ IS SYMMETRIC ~> Σ CAN BE DIAGONALIZED. $\Sigma = \underset{d\times d}{V}\cdot\underset{d\times d}{L}\cdot\underset{d\times d}{V^T}$

, WHERE: EIGENVECTOR $\bar{V}_i$, EIGENVALUE λ_i

$$V = \begin{bmatrix} | & & | \\ \bar{V}_1 & \cdots & \bar{V}_d \\ | & & | \end{bmatrix}, \quad L = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_d \end{bmatrix} \quad (\text{DIAGONAL MATRIX}) \quad / \ \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$$

- $\bar{V}_j$: PRINCIPAL AXES / PROJECTION OF DATA ON $\bar{V}_j$ ~> PC_j

↳ $\underset{n\times 1}{PC_j} = [X]\cdot\begin{bmatrix} | \\ \bar{V}_j \\ | \end{bmatrix}$

↳ $\underset{1\times d}{X_{i,PCA}} = [x_i][V]$ ~> COORDINATES OF i-TH POINT ON NEW PC SPACE

- RELATION BETWEEN SVD AND PCA:

$$\Sigma = \frac{1}{n-1}X^TX = \frac{1}{n-1}VSU^T\cdot USV^T = V\cdot\frac{S^2}{n-1}V^T \quad ((ABC)^T = C^TB^TA^T)$$

$$/ \ \lambda_i = \frac{1}{n-1}S_i^2$$

$\rightarrow$

$$\min_{W \in \mathbb{R}^{n,d},\, U \in \mathbb{R}^{d,n}} \left\{ \sum_{i=1}^{m} \| \bar{x}_i - UW\bar{x}_i \|_2^2 \right\} = \sum_{i=n+1}^{d} \lambda_i(A)$$

// SUM

$\lambda_d, \lambda_{d-1}, \dots, \lambda_{n+1}, \lambda_n, \dots, \lambda_1$

- **PCA PROCEDURE:**

1. CENTER SAMPLES: $X \longmapsto X - \bar{\mu}$ / $\bar{\mu} = \frac{1}{m} \sum_{i=1}^{m} \bar{x}_i$ $\rightsquigarrow$ MEAN OF EACH FEATURE OF DATASET
   (X: $m \times d$, $\bar{\mu}$: $1 \times d$)
2. APPLY PCA ON VECTORS $(\bar{x}_1 - \bar{\mu}), \dots, (\bar{x}_m - \bar{\mu})$

- PROPORTION OF VARIANCE EXPLAINED: $PVE = \frac{\lambda_i}{\sum_{j=1}^{d} \lambda_j} = \frac{\lambda_i}{\text{trace}(\Sigma)}$

- EFFICIENT IMPLEMENTATION OF PCA, FOR $d \gg m$:

LET $B = X \cdot X^T \rightarrow B_{i,j} = \langle \bar{x}_i, \bar{x}_j \rangle$

- IF $B\bar{u} = \lambda\bar{u} \rightarrow A(X^T\bar{u}) = X^T X X^T \bar{u} = X^T B \bar{u} = \lambda(X^T\bar{u})$

$\rightarrow \frac{X^T\bar{u}}{\| X^T\bar{u} \|}$ IS AN EIGENVECTOR OF A, WITH EIGENVALUE λ

$\rightarrow$ WE CAN PCA CALCULATING EIGENVALUES OF B INSTEAD OF A $\rightsquigarrow O(m^3 + m^2 d)$

m: # POINTS
d: # FEATURES

ex. PCA $d = 4$

f1	f2	f3	f4
1	2	3	4
5	5	6	7
1	4	2	3
5	3	2	1
8	1	2	2

$n = 5$

$\bar{\mu}_1 = \frac{1+5+1+5+8}{5} = 4$
$\bar{\mu}_2 = \dots = 3$
$\bar{\mu}_3 = \dots = 3$
$\bar{\mu}_4 = \dots = 3.4$

1. DATA ARE STANDARDIZED:

$$X \longmapsto \frac{X - \mu}{\sigma} \longrightarrow$$

$\rightarrow$ HERE ALSO σ IS CONSIDERED

f1	f2	f3	f4
-1	-0.63246	0	0.26062
0.33333	1.26491	1.73205	1.56374
-1	0.63246	-0.57735	-0.17375
0.33333	0	-0.57735	-1.04249
1.33333	-1.26491	-0.57735	-0.60812

2. COMPUTE Σ.

$$\Sigma(f_i, f_j) = \frac{\sum_{i=1}^{n} (f_i - \bar{\mu}_i)(f_j - \bar{\mu}_j)}{n}$$

↳ n-1 FOR SAMPLES
↳ WHOLE n USED

STANDARDIZED DATASET

f1	f2	f3	f4
-1	-0.63246	0	0.26062
0.33333	1.26491	1.73205	1.56374
-1	0.63246	-0.57735	-0.17375
0.33333	0	-0.57735	-1.04249
1.33333	-1.26491	-0.57735	-0.60812

ex: $\rightarrow Cov(f_1, f_2) = \frac{1}{5}\left[(-1-0)(-0.63246-0) + \ldots + (1.33\bar{3}-0)(-1.26491-0)\right] = -0.25298$

· Σ:

	f1	f2	f3	f4
f1	0.8	-0.25298	0.03849	-0.14479
f2	-0.25298	0.8	0.51121	0.4945
f3	0.03849	0.51121	0.8	0.75236
f4	-0.14479	0.4945	0.75236	0.8

3. CALCULATE λ_i AND $\bar{V}_i$:

EIGENVALUE, EIGENVECTOR

$$A\bar{v} = \lambda\bar{v} \rightarrow (A - \lambda I)\bar{v} = 0 \iff A - \lambda I = 0$$

$A = \Sigma$

· EIGENVALUES: $A - \lambda I = 0 \iff \det(A - \lambda I) = 0$

↳ SOL: $\lambda = [2.5757_{\lambda_1}, 1.0653_{\lambda_2}, 0.3439_{\lambda_3}, 0.0250_{\lambda_4}]$

· EIGENVECTORS:

$$\bar{v}_i: (A - \lambda_i I)\bar{v}_i = 0 \rightarrow \begin{pmatrix} 0.800000-\lambda & -(0.252982) & 0.038490 & -(0.144791) \\ -(0.252982) & 0.800000-\lambda & 0.511208 & 0.494498 \\ 0.038490 & 0.511208 & 0.800000-\lambda & 0.752355 \\ -(0.144791) & 0.494498 & 0.752355 & 0.800000-\lambda \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = 0$$

↳ SOL:

$\bar{V}_1$	$\bar{V}_2$	$\bar{V}_3$	$\bar{V}_4$
0.161960	-0.917059	-0.307071	0.196162
-0.524048	0.206922	-0.817319	0.120610
-0.585896	-0.320539	0.188250	-0.720099
-0.596547	-0.115935	0.449733	0.654547

$\rightsquigarrow$ V MATRIX

4. SORT EIGENVECTORS BY THEIR CORRESPONDING λ_i

→ IN THIS CASE THEY ARE ALREADY SORTED

K = 2 PRINCIPAL COMPONENTS

5. TRANSFORM DATA:

f1	f2	f3	f4
-1.000000	-0.632456	0.000000	0.260623
0.333333	1.264911	1.732051	1.563740
-1.000000	0.632456	-0.577350	-0.173749
0.333333	0.000000	-0.577350	-1.042493
1.333333	-1.264911	-0.577350	-0.608121

(5,4)

*

e1	e2
0.161960	-0.917059
-0.524048	0.206922
-0.585896	-0.320539
-0.596547	-0.115935

(4,2)

=

nf1	nf2
0.014003	0.755975
-2.556534	-0.780432
-0.051480	1.253135
1.014150	0.000239
1.579861	-1.228917

(5,2)

$\rightarrow \bar{X}_{i,PCA} = \bar{X}_i \cdot V_{[:,:K]}$ $\rightsquigarrow$ MULTIPLY $\bar{X}_i$ BY FIRST K EIGENVECTORS

- **RANDOM PROJECTIONS**: $\rightsquigarrow O(nd\,k)$ / k: TARGET DIMENSIONALITY

IN SOME CASE, RECONSTRUCTION OF $\bar{x}$ FROM $\bar{y} = W\bar{x}$ IS NOT IMPORTANT, BUT MAYBE IT IS IMPORTANT TO KEEP SOME PROPERTIES, SUCH AS:

① DO NOT DISTORT DISTANCES: $\|\bar{x}_i - \bar{x}_j\| \approx \|\bar{y}_i - \bar{y}_j\|$, $\forall i,j$

$\rightarrow \dfrac{\|W\bar{x}_i - W\bar{x}_j\|}{\|\bar{x}_i - \bar{x}_j\|} \approx 1$, $\forall i,j$

② $\forall\, \bar{x} \in Q$, $Q = \{\bar{x}_i - \bar{x}_j : i,j \in [m]\}$, WE HAVE $\dfrac{\|W\bar{x}\|}{\|\bar{x}\|} \approx 1$

RANDOM PROJECTION

- $\bar{x} \in \mathbb{R}^{d\times 1} \longmapsto W\bar{x}$ ($W \in \mathbb{R}^{k\times d}$, $\bar{x} \in \mathbb{R}^{d\times 1}$, $W\bar{x} \in \mathbb{R}^{k\times 1}$) → DATA PROJECTED IN k-DIMENSIONAL SPACE.

/ W IS A RANDOM MATRIX: $W_{i,j} \sim N\left(0, \frac{1}{n}\right)$

- LET $\bar{w}_i$ BE THE i^{TH} ROW OF W:

$$\rightarrow \mathbb{E}\left[\|W\bar{x}\|^2\right] = \sum_{i=1}^{n} \mathbb{E}\left[\left(\langle \bar{w}_i, \bar{x}\rangle\right)^2\right] = \sum_{i=1}^{n} \bar{x}^T \mathbb{E}\left[\bar{w}_i \bar{w}_i^T\right] \bar{x} =$$

$$= n\,\bar{x}^T \left(\frac{1}{n} I\right) \bar{x} = \|\bar{x}\|^2 \rightarrow \text{PROPERTY ② SATISFIED}$$

→ IN FACT: $\|W\bar{x}\|^2 \sim \chi^2_n$ ⇝ CHI-SQUARED DISTRIB.

$$\rightarrow P\left[\left|\frac{\|W\bar{x}\|^2}{\|\bar{x}\|^2} - 1\right| > \varepsilon\right] \leq 2e^{-\varepsilon^2 \frac{n}{6}}$$ HOEFFDING'S INEQUALITY

- **LEMMA (JOHNSON-LINDENSTRAUSS LEMMA)**: ⇝ OBTAINED APPLYING THE UNION BOUND OVER ALL VECTORS IN Q

LET Q BE A FINITE SET OF VECTORS IN $\mathbb{R}^d$

LET $\delta \in (0,1)$ AND $n \in \mathbb{Z}$ / $\varepsilon = \sqrt{\dfrac{6 \cdot \log\left(2|Q|/\delta\right)}{n}} \leq 3$ ⇝ REDUCED DIMENTION

→ WITH $p \geq 1-\delta$ (PROBABILITY) OVER A CHOICE OF A RANDOM MATRIX $W \in \mathbb{R}^{n,d}$, $W_{i,j} \sim N(0, \frac{1}{n})$

, WE HAVE: $\max_{\bar{x}\in Q} \left|\dfrac{\|W\bar{x}\|^2}{\|\bar{x}\|^2} - 1\right| < \varepsilon$

↳ SETTING $\varepsilon = \sqrt{\dfrac{6 \cdot \log\left(2|Q|/\delta\right)}{n}} \leq 3$ → PROPERTY ② STATISFIED, WITH C.I $1-\delta$

COMPRESSED SENSING :

- PRIOR ASSUMPTION :
  - $\bar{x} \approx U \bar{\alpha}$ / U IS ORTHONORMAL $\rightsquigarrow U^{-1} = U^T$, $\langle \bar{u}_i, \bar{u}_j \rangle = 0$, $\|\bar{u}_i\|_2^2 = 1$
  - $\|\bar{\alpha}\|_0 := |\{ i : \alpha_i \neq 0 \}| \leq S$, $S \ll d$
- HOW TO STORE $\bar{x}$:
  - FIND $\bar{\alpha} = U^T \bar{x}$, THEN SAVE NON-ZERO ELEMENTS OF $\bar{\alpha}$
  - REQUIRES ORDER OF $S \cdot \log(d)$ STORAGE
    - CAN'T WE JUST MEASURE THE PART THAT WON'T BE THROWN AWAY? → COMPRESSED SENSING
- COMPRESSED SENSING MAIN RESULTS :
  - ANY SPARSE SIGNAL CAN BE FULLY RECONSTRUCTED IF IT WAS COMPRESSED BY $\bar{x} \longmapsto W\bar{x}$ / W SATISFIES RESTRICTED ISOPERIMETRIC PROPERTY (RIP)
    - → IF A MATRIX HAS THIS PROPERTY → IT HAS A LOW DISTORTION OF THE NORM OF ANY SPARSE REPRESENTABLE VECTOR
  - RECONSTRUCTION CAN BE CALCULATED IN POLYNOMIAL TIME BY SOLVING A LINEAR PROGRAM
  - A RANDOM $n \times d$ MATRIX IS LIKELY TO SATISFY RIP CONDITION IF n IS GREATER THEN THE ORDER OF $S \cdot \log(d)$
- RESTRICTED ISOPERIMETRIC PROPERTY :

  A MATRIX $W \in \mathbb{R}^{n,d}$ IS (ε, S)-RIP IF $\forall \bar{x} \neq 0 \,/\, \|\bar{x}\|_0 \leq S$:

  $$\rightarrow \left| \frac{\|W\bar{x}\|_2^2}{\|\bar{x}\|_2^2} - 1 \right| \leq \varepsilon \quad \rightarrow \quad 1 - \varepsilon \leq \frac{\|W\bar{x}\|_2^2}{\|\bar{x}\|_2^2} \leq 1 + \varepsilon$$

- RIP MATRICES YIELD LOSSLESS COMPRESSION FOR SPARSE VECTORS:
  - THEOREM:

    LET $\varepsilon < 1$, W BE A $(\varepsilon, 2s)$-RIP MATRIX,
    , $\bar{x}$ BE A VECTOR / $\|\bar{x}\|_0 \le s$, $\bar{r} = W\bar{x}$,
    , $\hat{x} \in \underset{\bar{v}: W\bar{v} = \bar{r}}{\operatorname{argmin}} \|\bar{v}\|_0$:

    $\rightarrow \hat{x} = \bar{x}$ $\rightsquigarrow$ $\bar{x}$ CAN BE PERFECTLY RECONSTRUCTED

- EFFICIENT RECONSTRUCTION:

  IF WE FURTHER ASSUME THAT $\varepsilon < \frac{1}{1+\sqrt{2}}$: LINEAR PROGRAMMING PROBLEM

  $\rightarrow \bar{x} = \underset{\bar{v}: W\bar{v} = \bar{r}}{\operatorname{argmin}} \|\bar{v}\|_0 = \underset{\bar{v}: W\bar{v} = \bar{r}}{\operatorname{argmin}} \|\bar{v}\|_1$

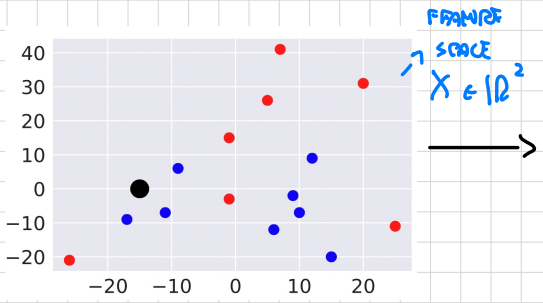
  - SUMMARY: WE CAN RECONSTRUCT AN SPARSE VECTOR EFFICIENTLY BASED ON $O(s \log(d))$ MEASUREMENTS

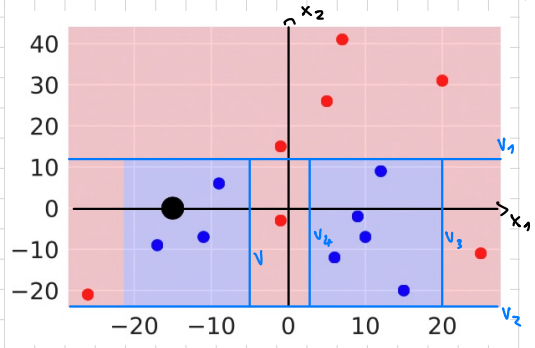
- PCA VS RANDOM PROJECTIONS:
  - RANDOM PROJECTIONS GUARANTEE PERFECT RECOVERY $\forall$ $O\left(\frac{n}{\log(d)}\right)$-SPARSE VECTORS
  - PCA GUARANTEE PERFECT RECOVERY IF ALL EXAMPLES ARE IN AN n-DIMENSIONAL SUBSPACE

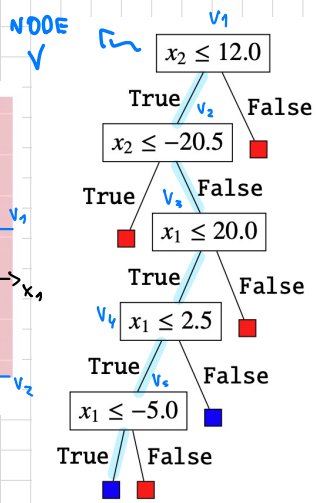
  $\rightarrow$
  - IF DATA IS $\bar{e}_1, \dots, \bar{e}_d$:
    - RANDOM PROJECTIONS $\rightarrow$ PERFECT
    - PCA $\rightarrow$ FAIL
  - IF $d \gg 0$, DATA EXACTLY IN n-DIM SUBSPACE (I ONLY SELECT FIRST n PC):
    - RANDOM PROJECTIONS $\rightarrow$ MIGHT FAIL
    - PCA $\rightarrow$ PERFECT
      - e.g. DATA IN $\mathbb{R}^2$ AND I TAKE $n = 2$ PC

  IF LINEAR DIMENSIONALITY REDUCTION: $\bar{x} \mapsto W\bar{x}$ $\rightarrow$
  - PCA: OPTIMAL IF LINEAR RECONSTRUCTION AND ERROR IS SQUARED DISTANCE
  - RANDOM PROJECTIONS: PRESERVE DISTANCES, EXACT RECONSTRUCTION FOR SPARSE VECTORS (BUT WITH NON-LINEAR RECONSTRUCTION)

- **DECISION TREES**:







- HOW TO CLASSIFY $\bullet = \bar{x} = [-15, 0]^T$:

  -> DEFINE A FUNCTION $y(\bar{x})$: $\bar{x} = [x_1, x_2] \mapsto$ RED / BLUE

- $\forall$ NODE v_1 -> REGION R_v OF FEATURE SPACE X SPLITTED INTO 2 SUBREGIONS
- $\forall$ LEAF NODE w -> REGIONAL PREDICTION FUNCTION y^w ON R_w
- $\forall$ TREE T -> # REGIONS = $|W|$ / W : SET OF LEAF NODES

- **PREDICTION FUNCTION**:

$$y(\bar{x}) = \sum_{w \in W} y^w(\bar{x}) \cdot \mathbb{1}_{[\bar{x} \in R_w]}$$

- **TRAINING LOSS**:

GIVEN $y(\bar{x})$ AS ABOVE, TRAINING SET $\tilde{T} = \{(\bar{x}_i, y_i)\}_{i=1}^n$:

$$l_{\tilde{T}}(y) = \sum_{w \in W}^{\forall \text{ REGION}} \frac{1}{n} \sum_{i=1}^{n \text{ } \forall \text{ POINT } \in \tilde{T}} \mathbb{1}_{[\bar{x}_i \in R_w]}^{\begin{cases} 1, \text{ IF } \bar{x}_i \in R_w \\ 0, \text{ OTHERWISE} \end{cases}} \cdot \text{Loss}\left(y_i, y^w(\bar{x}_i)\right)$$

CONTRIBUTION OF REGIONAL y^w TO OVERALL TRAINING LOSS

- **CONSTRUCTION OF DECISION TREES:**

IT IS NEEDED TO SPECIFY A SPLITTING RULE $\forall$ NODE v_i:

$$S : X \longmapsto \{0, 1\}$$

↳ IT IS A CONDITION

e.g. ROOT NODE OF PREVIOUS EXAMPLE: $\bar{x} \longmapsto \mathbb{1}_{[x_1 \leq 12]}$

- USING A SPLITTING RULE S WE CAN DIVIDE ANY SUBSET $\sigma \subseteq \tilde{\imath}$ OF THE TRAINING SET IN:

$$\sigma \begin{cases} \sigma_T := \{(\bar{x}, y) \in \sigma : S(\bar{x}) = \text{TRUE}\} \\ \sigma_F := \{(\bar{x}, y) \in \sigma : S(\bar{x}) = \text{FALSE}\} \end{cases}$$

→ FINAL TREE CAN BE BUILT USING:

$$T = \text{CONSTRUCT\_SUBTREE}(v_0, \tilde{\imath})$$

(ROOT NODE)

T_v : SUBTREE OF T FROM NODE v

$$g^v(\bar{x}) = g^{T_v}(\bar{x}) \mathbb{1}_{[\bar{x} \in R_{v_T}]} + g^{T_F}(\bar{x}) \mathbb{1}_{[x \in R_{v_F}]}$$

Algorithm 1: `Construct_Subtree`

Input: A node v and a subset of the training data: $\sigma \subseteq \tau$.
Output: A (sub) decision tree $\mathbb{T}_v$.

```
if termination criterion is met then        // v is a leaf node
    Train a regional prediction function g^v using training data σ.
else                                        // split the node
    Find the best splitting rule s_v for node v.
    Create successors v_T and v_F of v.
    σ_T ← {(x, y) ∈ σ : s_v(x) = True}
    σ_F ← {(x, y) ∈ σ : s_v(x) = False}
    T_{v_T} ← Construct_Subtree (v_T, σ_T)      // left branch
    T_{v_F} ← Construct_Subtree (v_F, σ_F)      // right branch
return T_v
```

- **REGIONAL PREDICTION FUNCTION FOR CLASSIFICATION:**

GIVEN LABELS $z = 0, \dots, C-1$, n_w : # FEATURE VECTORS IN R_w,

LET $p_z^w = \frac{1}{n_w} \sum_{\{(\bar{x}, y) \in \hat{\imath} : \bar{x} \in R_w\}} \mathbb{1}_{[y = z]}$ BE THE PROPORTION OF FEATURE VECTORS IN R_w WITH LABEL $z = 0, \dots, C-1$

→ REGIONAL PREDICTION FUNCTION FOR NODE w: $g^w(\bar{x}) = \underset{z \in \{0, \dots, C-1\}}{\text{argmax}} \{p_z^w\}$

- **REGIONAL PREDICTION FUNCTION FOR REGRESSION:**

g^w IS USUALLY CHOSEN AS THE MEAN RESPONSE IN THAT REGION:

$$g^w(\bar{x}) = \bar{y}_{R_w} := \frac{1}{n_w} \sum_{\{(\bar{x}, y) \in \hat{\imath} : \bar{x} \in R_w\}} y$$

- **OPTIMAL SPLITTING RULES:**

GIVEN $S(\bar{x}) = \mathbb{1}_{[x_j \leq \xi]}$ WE WANT TO CHOOSE j, ξ TO MINIMIZE:

$$\frac{1}{m} \sum_{j=1}^{m} \mathbb{1}_{[(\bar{x}_i, y) \in G_T]} \cdot \text{Loss}\left(y_i, \psi^T(\bar{x}_i)\right) + \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}_{[(\bar{x}_i, y) \in G_F]} \cdot \text{Loss}\left(y_i, \psi^F(\bar{x}_i)\right)$$

- FOR **REGRESSION**, WE WANT TO MINIMIZE:

$$\frac{1}{m} \sum_{(\bar{x}, y) \in \tilde{I} \,:\, x_j \leq \xi} \left(y - \bar{y}_T\right)^2 + \frac{1}{m} \sum_{(\bar{x}, y) \in \tilde{I} \,:\, x_j > \xi} \left(y - \bar{y}_F\right)^2$$

LET $\{x_{j,k}\}_{k=1}^{m}$ DENOTE THE POSSIBLE VALUES OF x_j, $j = 1, \dots, p$ WITHIN THE TRAINING SUBSET G ($m \leq n$ ELEMENTS)

-> TO MINIMIZE THE LOSS: EVACUATE IT $\forall\, m \times p$ VALUES $x_{j,k}$

-> THEN TAKE THE MINIMIZING PAIR $(j, x_{j,k})$

- FOR **CLASSIFICATION**, WE WANT TO MINIMIZE:

$$\frac{1}{m} \sum_{(\bar{x}, y) \in G_T} \mathbb{1}_{[y \neq y_T^*]} + \frac{1}{m} \sum_{(\bar{x}, y) \in G_F} \mathbb{1}_{[y \neq y_F^*]} \qquad / \; y^*: \text{MOST RELEVANT CLASS (MAJORITY VOTE)}$$

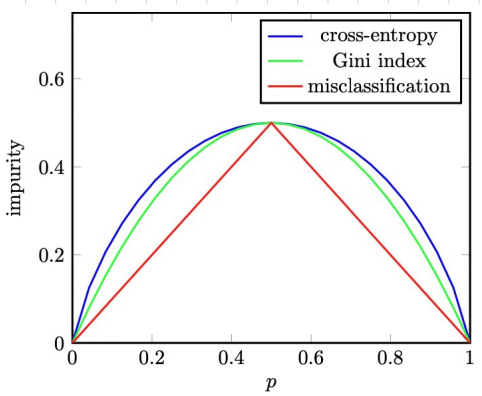
- **IMPURITIES:**

MINIMIZING LOSS CAN BE SEEN AS MINIMIZING A WEIGHTED AVERAGE OF "IMPURITIES" OF NODES G_T AND G_F:

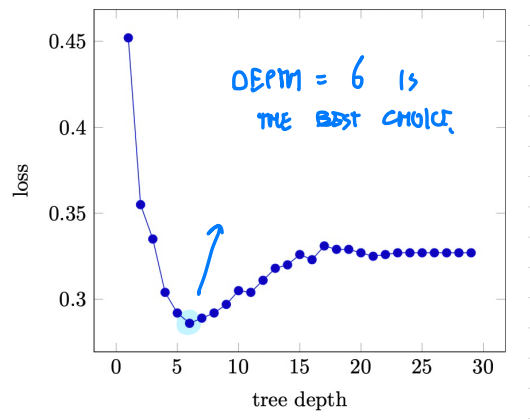
$$\to \frac{1}{|G|} \sum_{(\bar{x}, y) \in G} \mathbb{1}_{[y \neq y^*]} = 1 - \frac{1}{|G|} \sum_{(\bar{x}, y) \in G} \mathbb{1}_{[y = y^*]} = 1 - p_{y^*} = \underbrace{1 - \max_{z \in \{0, \dots, C-1\}} \{p_z\}}_{\text{MISCLASSIFICATION IMPURITY}}$$

- OTHER IMPURITY MEASURES:
  - ENTROPY: $-\sum_{z=0}^{C-1} p_z \cdot \log_2(p_z)$
  - GINI: $\frac{1}{2}\left(1 - \sum_{z=0}^{C-1} p_z^2\right)$

-> MAXIMAL $\Leftrightarrow$ LABEL PROPORTION IS $\frac{1}{2}$



- **TERMINATION CRITERIA:**
  - # DATA POINTS IN TREE NODE $\leq$ THRESHOLD
  - MAXIMAL TREE DEPTH
  - NO ADVANTAGE (OF TRAINING LOSS) IN SPLITTING REGIONS



- **ENSEMBLE METHODS:**

- **BOOTSTRAP AGGREGATION:**

GOAL: TRY TO REDUCE TENDENCY OF DTs TO OVERFIT DATA

- SUPPOSE B iid COPIES OF A TRAINING DATASET $\tilde{T}$: $\tilde{T}_1, \dots, \tilde{T}_B$
- WE CAN TAKE TRAIN B SEPARATE DTs, WITH AVERAGE PREDICTION:

$$y_{AVG}(\bar{x}) = \frac{1}{B} \sum_{b=1}^{B} y_{\tilde{T}_b}(\bar{x})$$

EXPECTED TRAINER

$$\rightarrow \text{BY LLN}: \quad y_{AVG}(\bar{x}) \xrightarrow[B \mapsto \infty]{} y^E := E[y_{\tilde{T}}]$$

- **THEOREM:**

EXPECTED SQUARED-ERROR GENERALIZATION RISK:

LET $\tilde{T}$ BE A RANDOM TRAINING SET AND LET $\bar{X}, Y$ BE A RANDOM FEATURE VECTOR AND RESPONSE THAT ARE INDEPENDENT OF $\tilde{T}$:

$$E^2\left(Y - y_{\tilde{T}}(X)\right) \geq E^2\left(Y - y^E(X)\right)$$

- MULTIPLE INDEPENDENT DATASETS ARE RARELY AVAILABLE

$\rightarrow$ WE CAN SUBSTITUTE THEM WITH BOOTSTRAPPED ONES:

WE CAN OBTAIN $\tilde{T}^*_1, \dots, \tilde{T}^*_B$ BY RESAMPLING FROM A SINGLE FIXED TRAINING SET $\tilde{T}$, AND USE THEM TO TRAIN B SEPARATE MODELS

$\sim$ = BOOTSTRAPPED

$\rightarrow$ WE OBTAIN A BAGGED ESTIMATOR OF THE FORM:

$$y_{bag}(\bar{x}) = \frac{1}{B} \sum_{b=1}^{B} y_{\tilde{T}^*_b}(\bar{x})$$

LITTLE CHANGE IN INPUT...

- MOST EFFECTIVE FOR SENSITIVE MODELS

LESS VARIANCE

$$V\left(y_{bag}(\bar{x})\right) = \frac{1}{B} V\left(y_{\tilde{T}}(\bar{x})\right)$$

Algorithm 1: Bootstrap Aggregation Sampling

Input: Training set $\tau = \{(x_i, y_i)\}_{i=1}^n$ and resample size B.
Output: Bootstrapped data sets.

```
for b = 1 to B do
    T*_b ← ∅
    for i = 1 to n do
        Draw U ~ U(0, 1);
        I ← ⌈nU⌉          // select random index;
        T*_b ← T*_b ∪ {(x_I, y_I)}.
return T*_b, b = 1, ..., B.
```

- IT CAN BE SHOWN THAT, FOR LARGE SAMPLE SIZES, ON AVERAGE ABOUT $\frac{1}{3}$ ($e^{-1} \approx 0.37$ EXACTLY) OF ORIGINAL SAMPLE POINTS ARE NOT INCLUDED IN THE BOOTSTRAPPED SET $\tilde{I}_b^*$, $b = 1, \dots, B$

→ THESE SAMPLES ($\widetilde{OOB}$ → OUT OF BAG SAMPLES) CAN BE USED FOR THE LOSS ESTIMATION

ex. RUNNING CODE "BAGGING EXAMPLE.py"

```
DecisionTreeRegressor R^2 score =  0.575438224929718
Bagging R^2 score =  0.7612121189201985
Bagging OOB R^2 score =  0.7758253149069059
```

- <u>CORRELATED PREDICTIONS:</u>

LET $Z_b = \psi_{\tilde{I}_b}(\vec{x})$, $b = 1, \dots, B$ ARE i.i.d PREDICTION VALUES FROM $\tilde{I}_1, \dots, \tilde{I}_B$

- SUPPOSE $V(Z_b) = \sigma^2 \rightarrow V(\bar{Z}_b) = \frac{1}{B}\sigma^2$
- IF $\{\tilde{I}_b^K\}$ ARE USED INSTEAD, $\{Z_b\}$ WILL BE CORRELATED

→ Z_b ARE IDENTICALLY DISTRIBUTED, BUT NOT INDEPENDENT

→ $V(\bar{Z}_B) = \rho\sigma^2 + \sigma^2\frac{(1-\rho)}{B}$ (ρ: CORRELATION COEFF.)

$B \to \infty$: $\sigma^2\frac{(1-\rho)}{B} \to 0$

- **RANDOM FORESTS :**

BAGGING PROBLEM :

SUPPOSE $\exists$ FEATURE WITH GOOD SPLIT → IT WILL BE SELECTED AT ROOT LEVEL → HIGHLY CORRELATED PREDICTIONS $\forall \{ \psi_{\tilde{I}_b^*} \}_{b=1}^{B}$

↳ SOL : RANDOM FOREST → ONLY A SUBSET OF FEATURES CONSIDERED DURING TREE CONSTRUCTION

→ $\forall \tilde{I}_b^*$ → DT BUILT WITH $m \ll p$ FEATURES FOR SPLITTING RULES

- DEFAULT VALUES FOR m : $m = \lceil \frac{1}{3} p \rceil$ OR $m = \lfloor \sqrt{p} \rfloor$
  ↳ IN PRACTICE : m IS AN HYPERPARAMETER

→ WE LOSE INTERPRETABILITY
↳ SOL : FEATURE IMPORTANCE

- **FEATURE IMPORTANCE :**

$\forall$ INTERNAL NODE v → $\Delta_{loss}(v)$ (→ DECREASE LOSS) IN TRAINING LOSS

- FEATURE IMPORTANCE OF X_j :

IT TELLS HOW MUCH FEATURE X_j HAS IMPACT ON DECREASING THE LOSS DURING BUILDING OF T

$$I_T(X_j) = \sum_{v \text{ INTERNAL} \in T} \Delta_{loss}(v) \cdot \mathbb{1}_{[X_j \text{ ASSOCIATED WITH } v]}, \quad 1 \leq j \leq p$$

RANDOM FOREST

$$\rightarrow I_{RF}(X_j) = \frac{1}{B} \sum_{b=1}^{B} I_{T_b}(X_j), \quad 1 \leq j \leq p$$

- **BOOSTING**:

IT IMPROVES ACCURACY OF ANY LEARNING ALGORITHM

- **BOOSTING FOR REGRESSION**:

-> IN BOOSTING, PREDICTION FUNCTIONS ARE LEARNED SEQUENTIALLY:

1. START FROM A WEAK LEARNER g_0 FOR DATA $\tilde{\tau} = \{(\bar{x}_i, Y_i)\}_{i=1}^{n}$

2. "BOOST" g_0 TO A NEW LEARNER: $g_1 := g_0 + h_1$

$$h_1 = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \text{Loss}\left(Y_i, g_0(\bar{x}_i) + h(\bar{x}_i)\right) \right\}$$

3. REPEAT FOR $g_2, \dots, g_B$

4. UNTIL THE BOOSTING PREDICTION FUNCTION: $g_B(\bar{x}) = g_0(\bar{x}) + \sum_{b=1}^{B} h_b(\bar{x})$

-> OR USING A STEP PARAMETER γ: $g_b := g_{b-1} + \gamma h_b$

- BOOSTING PROCEDURE WITH SQUARED ERROR LOSS:

- START WITH: $g_0 := \frac{1}{n} \sum_{i=1}^{n} Y_i$

- $\forall b = 1, \dots, B$: $\tilde{\tau}_b := \left\{ \left(\bar{x}_i, \underbrace{Y_i - g_{b-1}(\bar{x}_i)}_{\text{RESIDUALS } e_i^{(b)}}\right) \right\}$

- FIT PREDICTION FUNCTION h_b ON $\tilde{\tau}_b$: $g_b(\bar{x}) = g_{b-1}(\bar{x}) + \gamma h_b(\bar{x})$

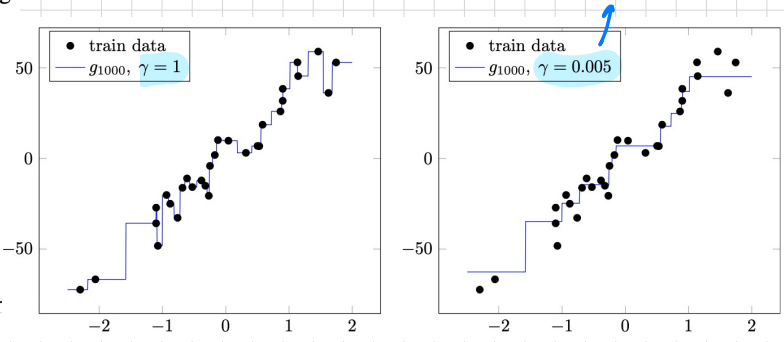
Algorithm 1: Regression Boosting with Squared-Error Loss

Input: Training set $\tau = \{(x_i, y_i)\}_{i=1}^{n}$, the number of boosting rounds B, and a shrinkage step-size parameter γ.

Output: Boosted prediction function.

1 Set $g_0(x) \leftarrow n^{-1} \sum_{i=1}^{n} y_i$.
2 **for** $b = 1$ **to** B **do**
3 Set $e_i^{(b)} \leftarrow y_i - g_{b-1}(x_i)$ for $i = 1, \dots, n$, and let $\tau_b \leftarrow \left\{\left(x_i, e_i^{(b)}\right)\right\}_{i=1}^{n}$.
4 Fit a prediction function h_b on the training data τ_b.
5 Set $g_b(x) \leftarrow g_{b-1}(x) + \gamma\, h_b(x)$.
6 **return** g_B.

NO OVERFITTING



GRADIENT BOOSTING :

- γ CAN BE VIEWED AS A STEP SIZE MADE IN THE OPPOSITE DIRECTION OF THE NEGATIVE GRADIENT OF SQUARED ERROR TRAINING LOSS :

$$z = \hat{y}_{b-1}(\bar{x}) \rightsquigarrow -\frac{\partial \text{Loss}(y_i, z)}{\partial z} = \frac{\partial \left(y_i - z\right)^2}{\partial z} =$$

$$= 2\left(y_i - \hat{y}_{b-1}(\bar{x}_i)\right) = 2\, e_i^{(b)}$$

- IT MIMICS A GRADIENT DESCENT ALGORITHM

Algorithm 2: Gradient Boosting

Input: Training set $\tau = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the number of boosting rounds B, a differentiable loss function $\text{Loss}(y, \widehat{y})$, and a step-size parameter γ.
Output: Gradient boosted prediction function.

1 Set $g_0(\boldsymbol{x}) \leftarrow 0$.
2 **for** $b = 1$ **to** B **do**
3 **for** $i = 1$ **to** n **do**
4 Evaluate the negative gradient of the loss at $(\boldsymbol{x}_i, y_i)$ via
$$r_i^{(b)} \leftarrow -\left.\frac{\partial\, \text{Loss}\,(y_i, z)}{\partial\, z}\right|_{z=g_{b-1}(\boldsymbol{x}_i)} \qquad i = 1, \ldots, n.$$
5 Approximate the negative gradient by solving
$$h_b = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=0}^{n} \left(r_i^{(b)} - [g_{b-1}(\boldsymbol{x}_i) + h(\boldsymbol{x}_i)]\right)^2.$$
6 Set $g_b(\boldsymbol{x}) \leftarrow g_{b-1}(\boldsymbol{x}) + \gamma\, h_b(\boldsymbol{x})$.
7 **return** g_B

ADABOOST : ⇝ FOR BINARY $(\{-1, 1\})$ CLASSIFICATION

- FINAL PREDICTION FUNCTION : $\hat{y}_B(\bar{x}) = \hat{y}_0(\bar{x}) + \sum_{b=1}^{B} h_b(\bar{x})$

$$h_b(\bar{x}) = \alpha_b\, c_b(\bar{x}), \quad \alpha_b \in \mathbb{R}_+, \quad c_b(\bar{x}) \in \{-1, 1\}$$

- AT BOOSTING OPERATION :

$$(\alpha_b, c_b) = \underset{\alpha \geq 0,\, c \in C}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \text{Loss}\left(y_i, \hat{y}_{b-1}(\bar{x}_i) + \alpha\, c(\bar{x}_i)\right) \right\}$$

$$\text{Loss}(y, \hat{y}) = e^{-y \cdot \hat{y}}, \quad \text{START} : \hat{y}_0 := 0$$

(FURTHER ANALYSIS : 16 ÷ 19)

- OPTIMAL α_b :
$$\alpha_b = \frac{1}{2} \ln\left(\frac{1 - \ell_\tau^{(b)}(c_b)}{\ell_\tau^{(b)}(c_b)}\right)$$

$$\ell_\tau^{(b)}(c) := \frac{\sum_{i=1}^n w_i^{(b)}\, \mathbb{I}\{c(\boldsymbol{x}_i) \neq y_i\}}{\sum_{i=1}^n w_i^{(b)}}$$

↳ WEIGHTED 0-1 LOSS, ITERATION b

- FINAL CLASSIFICATION :

$$\text{sign}\left(\sum_{b=1}^{B} \alpha_b\, c_b(\boldsymbol{x})\right)$$

Algorithm 3: AdaBoost

Input: Training set $\tau = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, and the number of boosting rounds B.
Output: AdaBoost prediction function.

1 Set $g_0(\boldsymbol{x}) \leftarrow 0$.
2 **for** $i = 1$ **to** n **do**
3 $w_i^{(1)} \leftarrow 1/n$
4 **for** $b = 1$ **to** B **do**
5 Fit a classifier c_b on the training set τ by solving
$$c_b = \underset{c \in C}{\operatorname{argmin}}\, \ell_\tau^{(b)}(c) = \underset{c \in C}{\operatorname{argmin}} \frac{\sum_{i=1}^n w_i^{(b)}\, \mathbb{I}\{c(\boldsymbol{x}_i) \neq y_i\}}{\sum_{i=1}^n w_i^{(b)}}.$$
6 Set $\alpha_b \leftarrow \frac{1}{2} \ln\left(\frac{1 - \ell_\tau^{(b)}(c_b)}{\ell_\tau^{(b)}(c_b)}\right)$. // Update weights
7 **for** $i = 1$ **to** n **do**
8 $w_i^{(b+1)} \leftarrow w_i^{(b)} \exp\{-y_i\, \alpha_b\, c_b(\boldsymbol{x}_i)\}$.
9 **return** $g_B(\boldsymbol{x}) := \sum_{b=1}^B \alpha_b\, c_b(\boldsymbol{x})$.