UNSUPERVISED LEARNING

└ REPRESENTATION LEARNING.

Goal: Given a set of "data points", "understand" something "useful" about them.

Data points → vectors in $\mathbb{R}^d$ $\begin{bmatrix} \text{height} \\ \text{weight} \\ \text{age} \end{bmatrix} \in \mathbb{R}^3$

Running theme: "COMPREHENSION IS COMPRESSION" (George Chaitin)

↳ understanding

└↳ learning

Problem: Input: $\{x_1, x_2, \cdots, x_n\}$ $x_i \in \mathbb{R}^d$ ← # of features

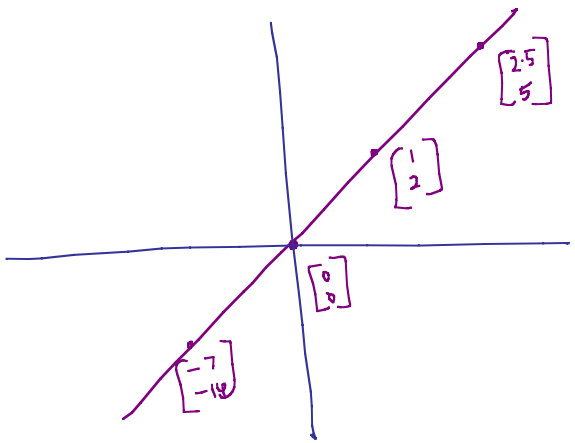
Output: Some "compressed" representation of the dataset

Example: $\{ \underset{x_1}{\begin{bmatrix} -7 \\ -14 \end{bmatrix}}, \underset{x_2}{\begin{bmatrix} 2.5 \\ 5 \end{bmatrix}}, \underset{x_3}{\begin{bmatrix} 0.5 \\ 1 \end{bmatrix}}, \underset{x_4}{\begin{bmatrix} 0 \\ 0 \end{bmatrix}} \}$

Question: How many real numbers are needed to store this dataset 8

Representative $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \in \mathbb{R}^2$ | **co-efficients** $\{-7, 2.5, 0.5, 0\}$ 6

NOTE: using representative & co-effecients can "RECONSTRUCT" the dataset exactly.



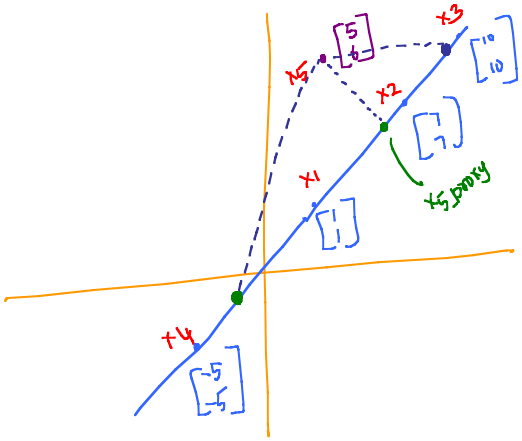
Rep: $\begin{bmatrix}1\\2\end{bmatrix}$ Coeff $\{-7, 2.5, 0.5, 0\}$

$\begin{bmatrix}1/\sqrt{5}\\2/\sqrt{5}\end{bmatrix}$ $\{-7\sqrt{5}, 2.5\sqrt{5}, 0.5\sqrt{5}, 0\}$

NOTE: Any vector along the purple line can be chosen as a representative [except $\begin{bmatrix}0\\0\end{bmatrix}$]

Original: # real numbers = dn

real numbers in Compressed representation = $d + n$



Rep

$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$

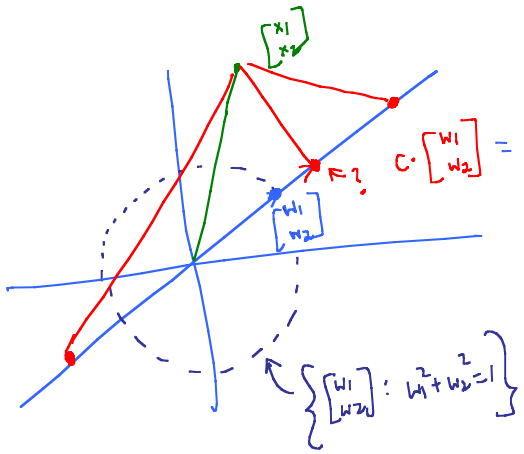
Co-eff

$\{(-5,-5), (1,1), (7,7), (5,6), (10,10)\}$

$x_3 = 10 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 10 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$

Q: Who can "pretend" to be a proxy for x_5 along the blue line?

Ans: Projection of x_5 onto the blue line.



$$c \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \left(\frac{x_1 w_1 + x_2 w_2}{w_1^2 + w_2^2} \right) \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$= \left(\frac{x^T w}{\|w\|^2} \right) \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$\min_c \quad \text{length}^2(\text{error vector})$$

$$\hookrightarrow \begin{bmatrix} x_1 - c w_1 \\ x_2 - c w_2 \end{bmatrix}$$

$$\min_c \quad (x_1 - c w_1)^2 + (x_2 - c w_2)^2$$

$$c^* = \left(\frac{x_1 w_1 + x_2 w_2}{w_1^2 + w_2^2} \right) \quad \text{(scalar)}$$

Inner product / dot product of x and w.

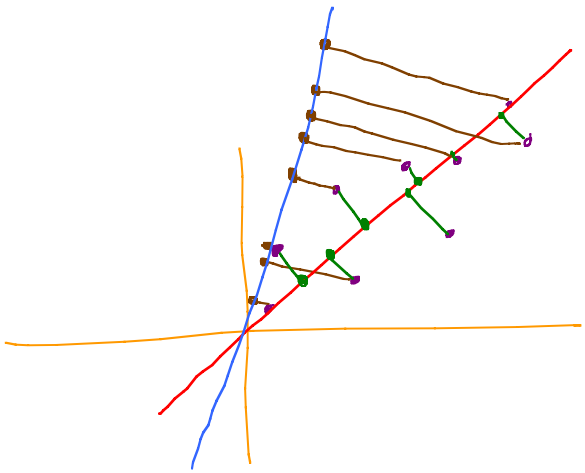
$$\text{length}^2 \left(\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \right) \leftarrow$$

NOTE: Can always pick $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ s.t. $\text{length}\left(\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \right) = 1$

$$\Rightarrow \quad c^* = (x^T w) \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

Goal: Develop a way to find a "compressed" representation of data when data points not-necessarily fall on line

↳ RULE, NOT EXCEPTION



Goal: Find the line that has the least "reconstruction" error.

Dataset: $\{x_1, x_2, \cdots, x_n\}$ $\quad x_i \in \mathbb{R}^d$

$\|z\|^2 = z^T z$

$$\text{ERROR}(\text{line}, \text{dataset}) = \sum_{i=1}^{n} \text{error}(\text{line}, x_i)$$

↑ Represented using w s.t. $\|w\|^2 = 1$

$$= \sum_{i=1}^{n} \text{length}^2\left(x_i - (x_i^T w)w\right)$$

$$= \sum_{i=1}^{n} \| x_i - (x_i^T w)\cdot w \|^2$$

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \| \underbrace{x_i}_{\in \mathbb{R}^d} - \underbrace{(x_i^T w)}_{\in \mathbb{R}} \cdot \underbrace{w}_{\in \mathbb{R}^d} \|^2$$

$$= \frac{1}{n}\sum_{i=1}^{n} \left(x_i - (x_i^T w)\cdot w\right)^T \left(x_i - (x_i^T w)\cdot w\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left[x_i^T x_i - (x_i^T w)^2 - (x_i^T w)^2 + (x_i^T w)^2 \cdot 1 \right]$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\underbrace{x_i^T x_i}_{} - (x_i^T w)^2 \right)$$

$$\min_{\substack{w: \\ \|w\|^2=1}} \quad g(w) = \frac{1}{n}\sum_{i=1}^{n} -(x_i^T w)^2$$

$$\max_{\substack{w: \\ \|w\|^2=1}} \quad \frac{1}{n}\sum_{i=1}^{n}(x_i^T w)^2 = \frac{1}{n}\sum_{i=1}^{n}\underbrace{(w^T x_i)}_{1\times d}\underbrace{(x_i^T w)}_{d\times 1}$$

$$= \frac{1}{n}\sum_{i=1}^{n} w^T (x_i x_i^T) w$$

$$= w^T \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} x_i x_i^T \right)}_{C \quad d\times d \text{ matrix}} w$$

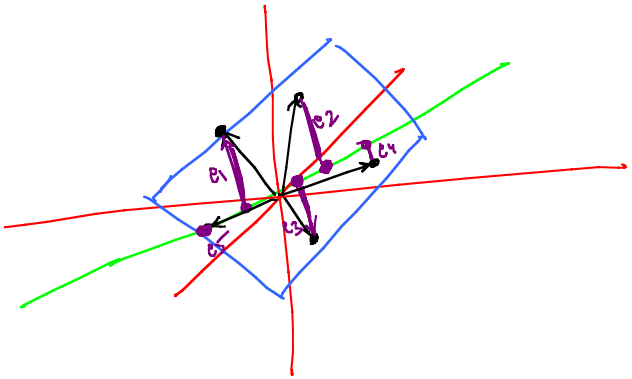
Equivalently

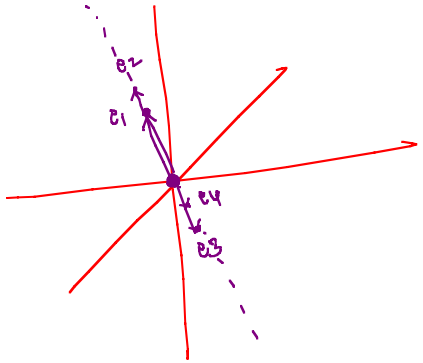
$$\max_{w:\ \|w\|^2 = 1} \quad w^T C w$$

$$C = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$$

↳ Co-variance matrix

Soln: w is the eigenvector corresponding to the maximum eigen value of C





$x \in \mathbb{R}^d$

↓ Find w

$(x^T w) \cdot w$

↓ Residue / error

$x - (x^T w) \cdot w$

Might not be error but has "information"

POSSIBLE ALGORITHM

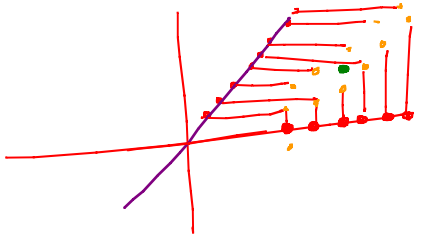
Input: $\{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$

→ $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ $\quad x_i = x_i - \mu \ \forall i$

→ Find "best" line $w_1 \in \mathbb{R}^d$

→ Replace $x_i \leftarrow x_i - (x_i^T w) w$

→ Repeat to obtain w_2



Issue: Data may not be centered.

Questions

- How to solve $\max_{w \atop \|w\|^2=1} w^T C w$?
- How many times to repeat the procedure?
- Where exactly is "compression" is happening?
- What "representations" are we learning?

Note Title 05-08-2022

$$D = \{x_1, x_2, \cdots, x_n\} \qquad x_i \in \mathbb{R}^d.$$

$$w_1 = \underset{\substack{w: \\ \|w\|^2 = 1}}{\operatorname{argmax}} \; w^T C w \qquad C = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$$

$$x_1' \rightarrow x_1 - (x_1^T w_1) w_1$$

$$x_2' \rightarrow x_2 - (x_2^T w_1) w_1$$

$$\vdots$$

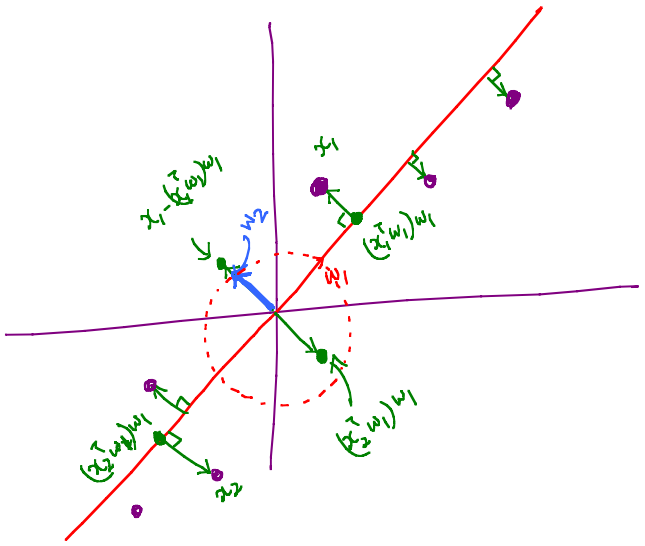
$$x_n' \rightarrow x_n - (x_n^T w_1) w_1$$

$$w_2 = \underset{\substack{w: \\ \|w\|_2^2 = 1}}{\operatorname{argmax}} \; w^T C' w$$

$$C' = \frac{1}{n} \sum_{i=1}^{n} x_i' x_i'^T$$

Note Title 05-08-2022

Question: What can we say about w_1 & w_2?



Observation:

→ All residues are orthogonal to w_1

→ Any line which minimizes sum of errors w.r.t residues must also be orthogonal to w_1 [ARGUE WHY]

$\Rightarrow$ $\boxed{w_2^T w_1 = 0}$

Note Title 05-08-2022

By continuing this procedure, we get $\{w_1, w_2, \cdots, w_d\}$ s.t $\|w_k\|_2^2 = 1 \ \forall k$ and $w_i^T w_j = 0 \ \forall i \neq j$

↓

ORTHONORMAL VECTORS

Residue after round 1

$$\{x_1 - (x_1^T w_1) w_1, \cdots, x_n - (x_n^T w_1) w_1\}$$

all vectors in $\in \mathbb{R}^d$

- w_2 → Best line that fits residues.
- $w_1^T w_2 = 0$

Note Title 05-08-2022

Residues after round 2

$$\left\{ x_1 - (x_1^T w_1) w_1 \quad - \left(\left(x_1 - (x_1^T w_1) w_1 \right)^T w_2 \right) w_2 \quad , \quad \cdots \right\}$$

$$= \left\{ x_1 - (x_1^T w_1) w_1 - \left(x_1^T w_2 - (x_1^T w_1) \cdot \underbrace{w_1^T w_2}_{=0} \right) w_2 \quad , \quad \cdots \right\}$$

$$= \left\{ x_1 - (x_1^T w_1) w_1 - (x_1^T w_2) w_2 \quad , \quad \cdots \right\}$$

Residues after d-rounds.

$\forall i \qquad x_i - \left((x_i^T w_1) w_1 + (x_i^T w_2) w_2 + \cdots + (x_i^T w_d) w_d \right) = \vec{0} \in \mathbb{R}^d$

Note Title 05-08-2022

$$\forall i \quad x_i = (x_i^T w_1) w_1 + (x_i^T w_2) w_2 + \cdots + (x_i^T w_d) w_d.$$

What have we gained?

- If data lives in a "low" dimensional linear sub-space, then residues become 0 much earlier than d rounds.

Example: Say Dataset is such that after 3-rounds, residues become 0.

$$\text{Dataset} = \{x_1, \cdots, x_n\} \qquad x_i \in \mathbb{R}^{100}$$

Note Title 05-08-2022

$\forall i \quad x_i = (x_i^T w_1) w_1 + (x_i^T w_2) w_2 + (x_i^T w_3) w_3$

$\{w_1, w_2, w_3\} \in \mathbb{R}^{100}$

Rep

$\{w_1, w_2, w_3\}$

↳ Common for dataset

Co-efficients

$x_i \rightarrow [x_i^T w_1 \quad x_i^T w_2 \quad x_i^T w_3] \in \mathbb{R}^3$

↳ Data point specific

original: $100 \times n$

$100 \times 100 = 10000$

$d \times n$

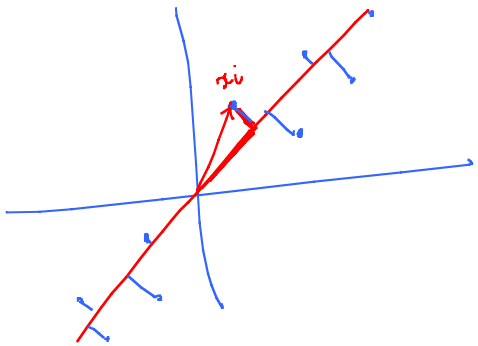
now: $3 \times 100 + 3n$

$3 \times 100 + 3 \times 100 = 600$

$d \times k + k \times n$

Note Title 05-08-2022

Question: What if data "approximately" lies in a low-dimensional space?



PYTHAGOROUS

For any $w \in \mathbb{R}^d$, s.t $\|w\|_2^2 = 1$

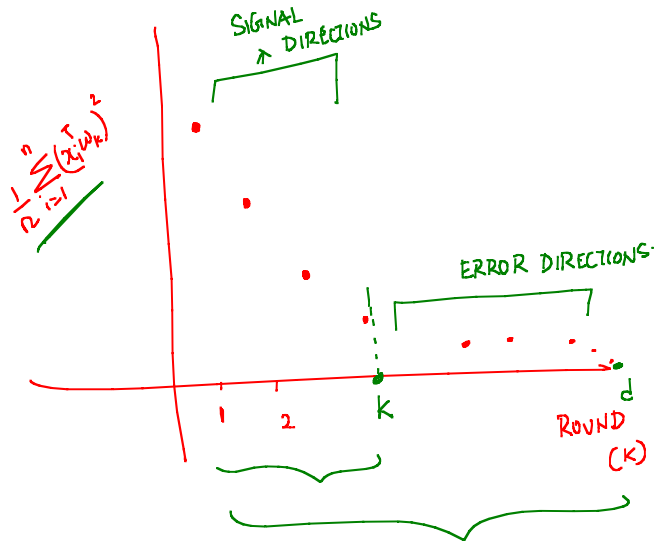
$$\text{ie } \|x_i\|^2 = \|x_i - (x_i^T w)w\|^2 + \|(x_i^T w)w\|^2$$

$$\frac{1}{n}\sum_{i=1}^{n}\|x_i\|^2 = \frac{1}{n}\sum_{i=1}^{n}\|x_i - (x_i^T w)w\|^2 + \frac{1}{n}\sum_{i=1}^{n}\|(x_i^T w)w\|^2$$

↑ As large as possible.

"Larger the value of $\frac{1}{n}\sum_{i=1}^{n}(x_i^T w)^2$, the better the fit"

Note Title 05-08-2022



ENTER LINEAR ALGEBRA

$$\max_{w:\ \|w\|_2^2=1} w^T C w \qquad C = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$$

$C \rightarrow$ Covariance matrix.

Soln: w_1 is eigenvector corresponding to the "largest" eigenvalue of C [HILBERT's min-max theorem]

In fact $\{w_1, \ldots, w_d\}$, the eigenvectors of C form an orthonormal basis.

$w_k \rightarrow$ Best line one can obtain in round k

Note Title 05-08-2022

What do eigenvalues of C mean?

we know

$$C w_1 = \lambda_1 w_1$$

$$w_1^T C w_1 = w_1^T (\lambda_1 w_1) = \lambda_1$$

$$\lambda_1 = w_1^T C w_1 = w_1^T \left(\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T \right) w_1$$

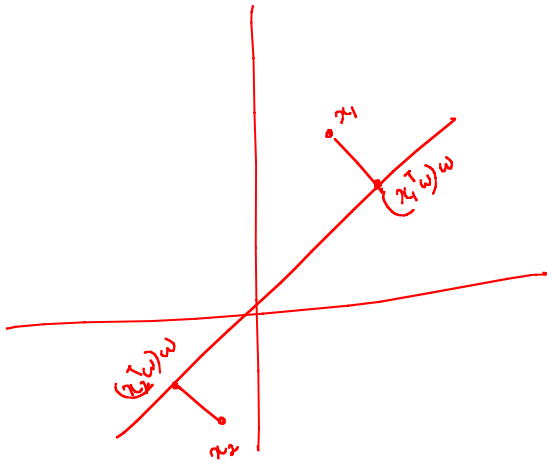
$$\boxed{\lambda_1 = \frac{1}{n} \sum_{i=1}^{n} (x_i^T w_1)^2}$$ ← term we used earlier

Rule of thumb for # dimensions

$$\left(\sum_{i=1}^{L} \lambda_i(C) \Big/ \sum_{i=1}^{d} \lambda_i(C) \right) \geq 0.95$$

↳ usually in practice

Note Title 05-08-2022



$\{(x_1^T w), \cdots, (x_n^T w)\}$

Average?

$$\frac{1}{n}\sum_{i=1}^{n}(x_i^T w) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^T w$$

$0^T w$ [for a centered dataset].

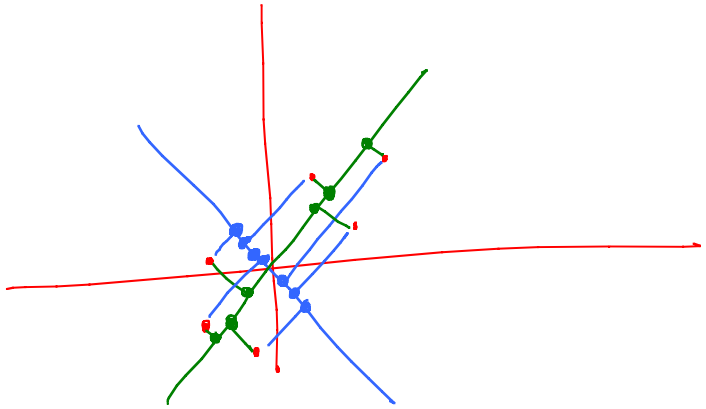
$= 0$

Variance

$$\frac{1}{n}\sum_{i=1}^{n}(x_i^T w - \underbrace{\text{mean}}_{=0})^2 = \boxed{\frac{1}{n}\sum_{i=1}^{n}(x_i^T w)^2}$$

ERROR MINIMIZATION on CENTERED DATASET $\iff$ VARIANCE MAXIMIZATION

Note Title 05-08-2022



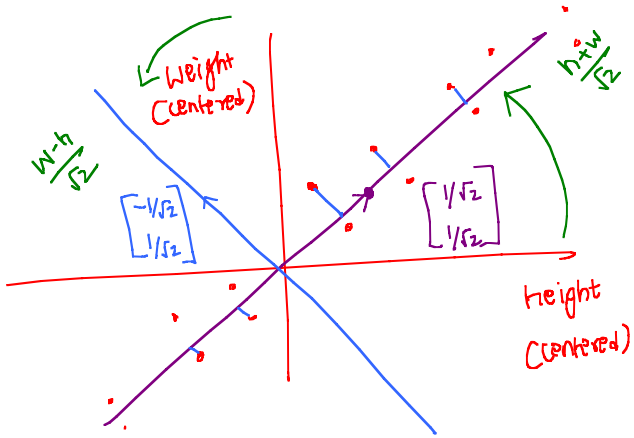
Want directions where

Projections don't "crowd up"

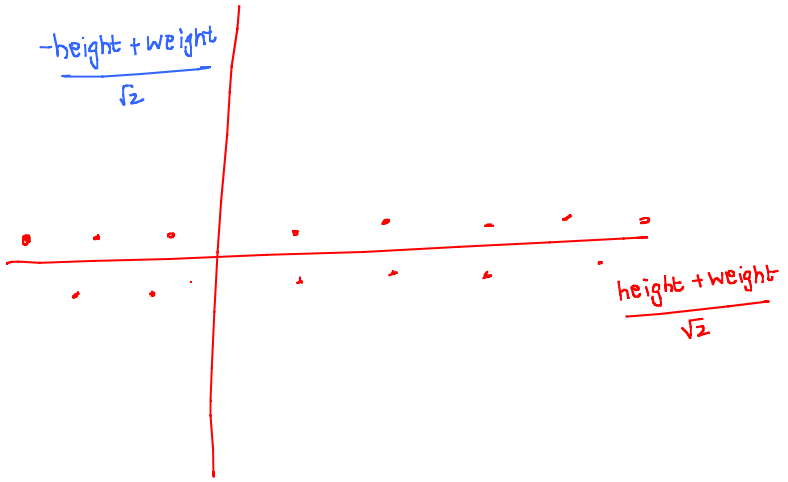
i.e., Variance is not small.

ONE MORE EXAMPLE

Note Title 05-08-2022



height gives some information about weight.



(h+w)/√2 does not give any information about (w-h)/√2.

"de-correlated"

Note Title 05-08-2022

PRINCIPAL COMPONENT ANALYSIS – $\{w_1, \ldots, w_k\}$

PRINCIPAL COMPONENT

"DIMENSIONALITY REDUCTION"

PCA Finds combination of features that are de-correlated. [loosely speaking independent of each other].

"EIGENFACES"

Note Title 8/10/2022

ISSUES / CONCERNS with PCA

- TIME COMPLEXITY - Finding the Eigen vectors and Eigen values.

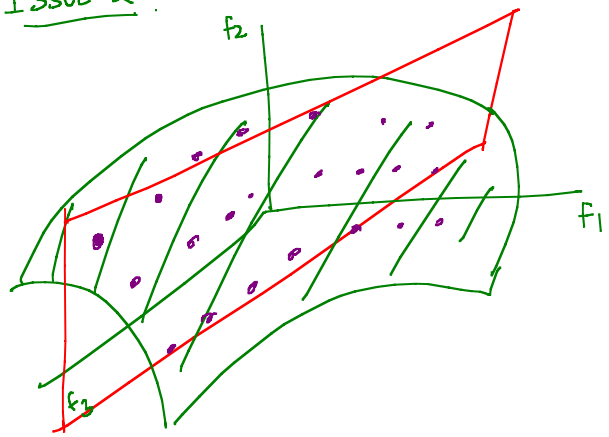
  $C \in \mathbb{R}^{d \times d}$ Typically $O(d^3)$

- Issue when d is large

  Example: Face recognition (EigenFaces)

Note Title 8/10/2022

- ISSUE 2 :



Data may not necessarily live in a low-dimensional LINEAR subspace.

SURPRISING RESULT : Same solution to both issues!

Note Title 8/10/2022

Issue 1: Large d $\quad [d >> n]$

d: # features, n: # datapoints.

$$X = \begin{bmatrix} | & | & | & & | \\ x_1 & x_2 & x_3 & \cdots & x_n \\ | & | & | & & | \end{bmatrix}$$

$$X \in \mathbb{R}^{d \times n}$$

$$C = \frac{1}{n}\left(\sum_{i=1}^{n} x_i x_i^T\right)$$

$$XX^T = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \sum_{i=1}^{n} x_i x_i^T$$

Exercise: [Show this]

$$\Rightarrow \quad \boxed{C = \frac{1}{n} XX^T}$$

Note Title 8/10/2022

Let w_k be the eigenvector corresponding to the k^{th} largest eigenvalue of C (λ_k) ← e-value

$$C w_k = \lambda_k w_k \quad \text{[by definition]}$$

$$\left(\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T \right) w_k = \lambda_k w_k$$

$$w_k = \sum_{i=1}^{n} \left(\frac{x_i^T w_k}{n \lambda_k} \right) \cdot x_i$$

w_k is a LINEAR COMBINATION of data points!

Note Title 8/10/2022

$$\begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix} \begin{bmatrix} \alpha_{k1} \\ \vdots \\ \alpha_{kn} \end{bmatrix} = \sum_{i=1}^{n} \alpha_{ki} x_i$$

$$w_k = X \alpha_k \quad \text{for some } \alpha_k \in \mathbb{R}^n$$

How to get α_k ?

Some Algebra:

$$w_k = X \alpha_k \qquad \forall k$$

$$C w_k = \lambda_k w_k$$

$$\left(\frac{1}{n} X X^T\right)(X \alpha_k) = \lambda_k X \alpha_k$$

Note Title 8/10/2022

$$(XX^T)\, X\alpha_k = n\lambda_k\, X\alpha_k$$

Pre multiply by X^T

$$X^T\big((XX^T)\, X\alpha_k\big) = X^T\big(n\lambda_k X\alpha_k\big)$$

$$(X^TX)(X^TX)\alpha_k = n\lambda_k (X^TX)\alpha_k$$

$X \in \mathbb{R}^{d\times n}$

$X^TX \in \mathbb{R}^{n\times n}$

Call $X^TX := K$

$$K^2\alpha_k = n\lambda_k K\alpha_k$$

Note Title 8/10/2022

If we can find α_k that satisfies

$$K\alpha_k = (n\lambda_k)\alpha_k$$

↳ Eigen equation

We know $w_k = X\alpha_k$

$$w_k^T w_k = (X\alpha_k)^T (X\alpha_k) = \alpha_k^T (X^T X)\alpha_k$$

$$1 = \alpha_k^T K \alpha_k$$

Note Title 8/10/2022

LINEAR ALGEBRA FACT: The non-zero eigenvalues of XX^T and X^TX are exactly the same!

($XX^T \in \mathbb{R}^{d \times d}$, $X^TX \in \mathbb{R}^{n \times n}$)

$C = \frac{1}{n} XX^T$

Eigenvectors $= \{w_1, \ldots, w_\ell\}$ $\forall k, \|w_k\|^2 = 1$

Eigenvalues $= \{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\ell\}$

$XX^T = nC$

Eigenvectors $= \{w_1, \ldots, w_\ell\}$

Note Title 8/10/2022

Eigenvalues $= \{ n\lambda_1 \geq n\lambda_2 \geq \cdots \geq n\lambda_p \}$

$\frac{X^T X}{k}$

Eigenvectors $= \{ \beta_1, \ldots, \beta_p \}$

$\|\beta_k\|^2 = 1 \quad \forall k$

Eigenvalues $= \{ n\lambda_1 \geq n\lambda_2 \cdots \geq n\lambda_p \}$

$K\beta_k = (n\lambda_k)\beta_k$ IS $\beta_k = \alpha_k$?

Note Title 8/10/2022

$$\beta_k^T K \beta_k = \beta_k^T (n\lambda_k \beta_k) = n\lambda_k \beta_k^T \beta_k$$

$$= n\lambda_k$$

Set $\forall k \quad \alpha_k := \frac{\beta_k}{\sqrt{n\lambda_k}}$

Now ① $K\alpha_k = (n\lambda_k)\alpha_k$

② $\alpha_k^T K \alpha_k = 1$

$$= \frac{\beta_k^T K \beta_k}{n\lambda_k} = \frac{n\lambda_k}{n\lambda_k} = 1$$

Note Title 8/10/2022

Input: $D = \{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$ [$d \gg n$]

$\forall i,j$

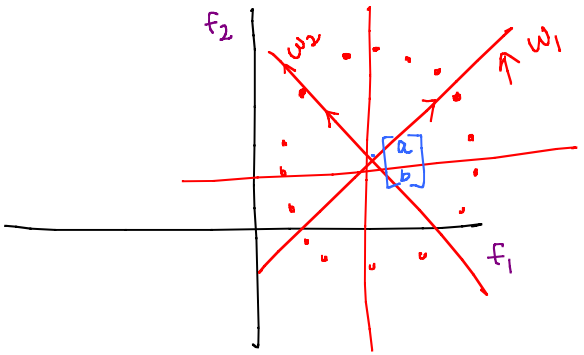
$k_{ij} = x_i^T x_j$

- Step 1: Compute $K = X^T X$ $K \in \mathbb{R}^{n \times n}$
- Step 2: Compute eigendecomposition of K

  eigenvectors $\{\beta_1, \ldots, \beta_l\}$ corresponding to e.values $\{n\lambda_1, \ldots, n\lambda_l\}$

  $O(n^3)$
- Step 3: set $\alpha_k = \frac{\beta_k}{\sqrt{n\lambda_k}}$ $\forall k = 1, \ldots, l$
- Step 4 $w_k = X\alpha_k$ $\forall k$.

Issue 2 → Non-linear relationships



What would PCA give? $\{w_1, w_2\}$ and both important.

Relation between features

$$(f_1 - a)^2 + (f_2 - b)^2 = r^2$$

$$\Rightarrow \boxed{f_1^2 + a^2 - 2f_1 a + f_2^2 + b^2 - 2f_2 b - r^2 = 0} \leftarrow (*)$$

$x \in \mathbb{R}^2$ (b) $\phi(x)$

$[f_1 \quad f_2] \longrightarrow [1 \quad f_1^2 \quad f_2^2 \quad f_1 f_2 \quad f_1 \quad f_2]$

$\mathbb{R}^2$ $\mathbb{R}^6$

Let $u \in \mathbb{R}^6$ $[a^2+b^2-r^2 \quad 1 \quad 1 \quad 0 \quad -2a \quad -2b]$

Each datapoint satisfies $\boxed{\phi(x)^T u = 0}$ $\equiv$ (*) $\Rightarrow$ the datapoints lie in a LINEAR subspace of $\mathbb{R}^6$!

IDEA: Transform features from low dimension $\mathbb{R}^d$ to high dimension $\mathbb{R}^D$

$x \in \mathbb{R}^d \xrightarrow{\phi} \phi(x) \in \mathbb{R}^D$

$\boxed{\phi(x)^T \phi(x)}$ instead of $\phi(x)\phi(x)^T$

- We already know how to handle the case when $D >> n$

ISSUE – $\phi(x) \in \mathbb{R}^D$ may be too hard to compute.

$[f_1 \quad f_2 \quad f_3 \quad f_4]$

ϕ → Cubic relations

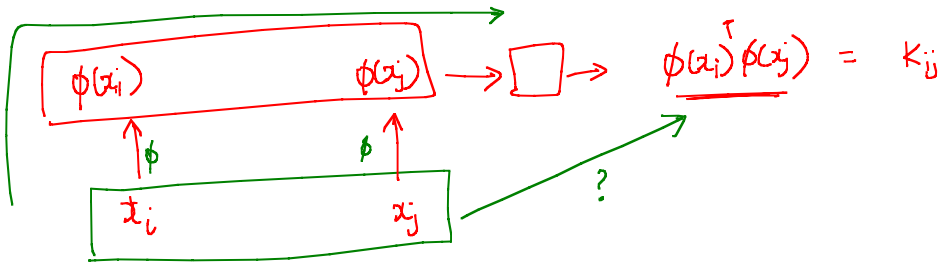
$[1 \quad f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_1f_2 \quad f_1f_3 \quad \cdots \quad f_3f_4 \quad f_1f_2f_3 \quad f_1f_2f_4 \quad \cdots]$

$1 \quad (4C_0) \qquad 4 \quad (4C_1) \qquad 6 \quad (4C_2) \qquad 4 \quad (4C_3)$

$1 + 4 + 6 + 4$

In general, d features, $\leq p^{-th}$ power

$$\sum_{i=0}^{p} {}^dC_i \approx O(d^p)$$



Example.

$x = [f_1 \quad f_2]$ $\qquad x' = [g_1 \quad g_2]$

Consider the function

$$(x^T x' + 1)^2 = \left([f_1 \quad f_2] \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} + 1 \right)^2 = (f_1 g_1 + f_2 g_2 + 1)^2$$

$$= f_1^2 g_1^2 + f_2^2 g_2^2 + 1 + 2 f_1 g_1 f_2 g_2 + 2 f_1 g_1 + 2 f_2 g_2$$

$$(x^T x' + 1)^2$$

$[f_1 \; f_2]$ $[g_1 \; g_2]$

$$= \underbrace{\begin{bmatrix} f_1^2 & f_2^2 & 1 & \sqrt{2} f_1 f_2 & \sqrt{2} f_1 & \sqrt{2} f_2 \end{bmatrix}} \begin{bmatrix} g_1^2 \\ g_2^2 \\ 1 \\ \sqrt{2} g_1 g_2 \\ \sqrt{2} g_1 \\ \sqrt{2} g_2 \end{bmatrix}$$

$$= \phi(x)^T \phi(x')$$

where

$$\phi(x) = \phi\left(\begin{bmatrix} a \\ b \end{bmatrix}\right) = \begin{bmatrix} a^2 \\ b^2 \\ 1 \\ \sqrt{2} ab \\ \sqrt{2} a \\ \sqrt{2} b \end{bmatrix}$$

INSIGHT:

$(x^T x' + 1)^2$ Computes the dot-product in a "transformed space".

$$[f_1 \; f_2] \rightarrow [\underbrace{f_1^2 \; f_2^2 \; 1 \; \sqrt{2} f_1 f_2 \; \sqrt{2} f_1 \; \sqrt{2} f_2}]$$

$$[g_1 \; g_2] \rightarrow [\underbrace{g_1^2 \; g_2^2 \; 1 \; \sqrt{2} g_1 g_2 \; \sqrt{2} g_1 \; \sqrt{2} g_2}]$$

$x \in \mathbb{R}^2$

$\downarrow$

$\phi(x) \in \mathbb{R}^6$

WE MANAGED TO COMPUTE $\phi(x)^T \phi(x')$ WITHOUT EXPLICITLY COMPUTING $\phi(x)$

MORE EXAMPLES.

$x \in \mathbb{R}^d$:

$$k(x, x') = (x^T x' + 1)^p$$

for some $p \geq 1$

→ Can be shown to be a "valid" function

i.e., $\exists\ \phi : \mathbb{R}^d \to \mathbb{R}^D$ such that

$$k(x, x') = \phi(x)^T \phi(x')$$

EXERCISE:

Compute the explicit ϕ for $p=3$ and $p=4$

ONE MORE EXAMPLE

$$K(x, x') = \exp\left(- \frac{\|x - x'\|^2}{2\sigma^2} \right) \quad \text{for some } \sigma > 0$$

RADIAL BASIS FUNCTION.

→ Can be shown to be a "valid" map.

→ Interestingly, ϕ in this case maps x to an "infinite" dimensional space.

[Technicalities aside, can think of this as mapping a data point to a "function" and dot-products between functions become integrals]

KERNEL FUNCTION

Any function $k: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ which is a "valid" map is called a kernel function

$$k(x, x') = (x^T x' + 1)^p \Rightarrow \text{POLYNOMIAL KERNEL}$$

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \Rightarrow \text{RADIAL BASIS / GAUSSIAN KERNEL}$$

Given a function $k: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, how can we say its a valid kernel ?

METHOD 1: Exhibit a map ϕ explicitly.

[might be hard sometimes]

METHOD 2: MERCER'S THEOREM (Informal)

$k: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a valid kernel

if and only if

$\phi(x)^T \phi(x)$

$\phi(x)\phi(x)^T$

(a) k is symmetric i.e., $k(x, x') = k(x', x)$.

(b) For any dataset $\{x_1, \ldots, x_n\}$, the matrix

$K \in \mathbb{R}^{n \times n}$ where $K_{ij} = k(x_i, x_j)$ is POSITIVE SEMI DEFINITE

All e-values of K are non-negative. $\longrightarrow$ (b)

KERNEL PCA

- Input - $\{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$; kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$.

- **Step 1 :** Compute $K \in \mathbb{R}^{n \times n}$ where

  $K_{ij} = k(x_i, x_j)$ $\forall i,j$

  → "Center the kernel"

- **Step 2 :** Compute $\beta_1, \ldots, \beta_l$ eigen vectors & $n\lambda_1 \geq \ldots n\lambda_l$ eigen values of K.

  and normalize to get

  $\forall u \quad \alpha_u = \frac{\beta_u}{\sqrt{n\lambda_u}}$

~~X~~ • Step 3 : $w_k = \phi(x)\,\alpha_k$ → Defeats the purpose because it needs $\phi(x)$.

- We cannot "reconstruct" the eigenvectors of the co-variance matrix.
- But we can still compute the "compressed" representation.

$$\phi(x_i)^T w_k = \phi(x_i)^T \left(\sum_{j=1}^{n} \phi(x_j)\,\alpha_{kj} \right)$$

$$= \sum_{j=1}^{n} \alpha_{kj}\, \phi(x_i)^T \phi(x_j) = \sum_{j=1}^{n} \alpha_{kj}\, K_{ij}$$

For downstream tasks, this is usually good enough.

- Modified Step 3 : Compute $\sum_{j=1}^{n} \alpha_{kj} K_{ij} \quad \forall \; k$

$$x_i \in \mathbb{R}^d \rightarrow \left[\sum_{j=1}^{n} \alpha_{1j} K_{ij} \;,\; \sum_{j=1}^{n} \alpha_{2j} K_{ij} \;\cdots\; \sum_{j=1}^{n} \alpha_{kj} K_{ij} \right]$$

DETAILS : (Centering the kernel)

Given $K \in \mathbb{R}^{n \times n}$ $\quad K_{ij} = k(x_i, x_j) \;\; \forall i,j$

$\{x_1, \ldots, x_n\}$

$\downarrow$

$\{\phi(x_1) \ldots, \phi(x_n)\}$

$\{\phi(x_1) - \mu, \ldots, \phi(x_n) - \mu\}$

$\mu = \frac{1}{n} \sum_i \phi(x_i)$

$(\phi(x_i) - \mu)^T (\phi(x_j) - \mu)$

$\phi(x_i)^T \phi(x_j)$

$-\mu^T \phi(x_j)$

$-\mu^T \phi(x_i)$

$+\mu^2$

Create a new kernel K^c ← centered.

$$K^c_{ij} = K_{ij} - \theta_i \mathbb{1}_j^T - \mathbb{1}_i \theta_j^T + P \, \mathbb{1}_i \mathbb{1}_j^T$$

where

$$\theta_i = \frac{1}{n} \sum_{k=1}^{n} K_{ik} \quad \forall i$$

$$P = \frac{1}{n^2} \sum_{ij} K_{ij}$$

Note Title 8/23/2022

Unsupervised Learning

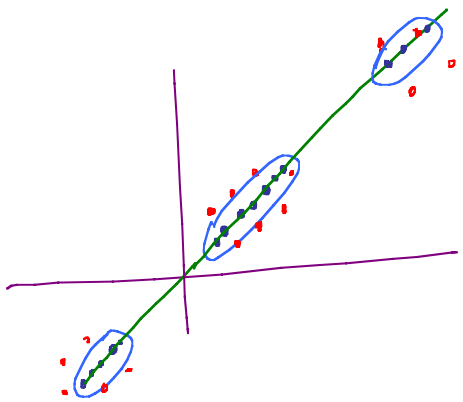
└ Representation learning

└ PCA

└ kernel PCA

└

Clustering (today)



"Clustered together"

Note Title 8/23/2022

Goal: $\{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$

Partition the data k different clusters

Example: $\{x_1, x_2, x_3, x_4, x_5\}$ $k = 3$

$[x_1, x_2, x_5]$ $[x_3]$ $[x_4]$

$[x_1, x_4]$ $[x_2]$ $[x_3, x_5]$

3^5 possibilities

Note Title 8/23/2022

$x_1, x_2, \cdots, x_n$ ← DATA POINTS

$z_1, z_2, \cdots, z_n$ ← CLUSTER INDICATOR $\quad z_i \in \{1, \cdots, K\}$

Question: Given a cluster assignment, how good is it?

$$F(z_1, \cdots, z_n) = \sum_{i=1}^{n} \| x_i - \mu_{z_i} \|_2^2$$

↳ Mean/average of z_i^{th} cluster

$$\mu_k = \frac{\sum_{i=1}^{n} x_i \, \mathbb{1}(z_i = k)}{\sum_{i=1}^{n} \mathbb{1}(z_i = k)}$$

$$\mathbb{1}(u) = \begin{cases} 1 & \text{if } u \text{ is true} \\ 0 & \text{o/w} \end{cases}$$

Note Title 8/23/2022

Example: x_1 x_2 x_3 x_4 x_5 $k=2$

$z_1=1$ $z_2=2$ $z_3=1$ $z_4=1$ $z_5=2$

$\mu_1 = \frac{x_1+x_3+x_4}{3}$; $\mu_2 = \frac{x_2+x_5}{2}$

Goal $\min_{\{z_1,\dots,z_n\}} \sum_{i=1}^{n} \|x_i - \mu_{z_i}\|^2$

Too many possibilities! (k^n)

NP-HARD

Note Title 8/23/2022

LLYOD'S ALGORITHM / K-MEANS ALGORITHM

INITIALIZATION $z_1^0, z_2^0, \ldots, z_n^0 \in \{1, \ldots, k\}$ (0 ← ITERATION)

→ UNTIL CONVERGENCE

- COMPUTE MEANS $\forall k \quad \mu_k^t = \dfrac{\sum_{i=1}^{n} x_i \mathbb{1}(z_i^t = k)}{\sum_{i=1}^{n} \mathbb{1}(z_i^t = k)}$

- RE-ASSIGNMENT STEP $\forall i \quad z_i^{t+1} = \arg\min_k \| x_i - \mu_k^t \|_2^2$

[IF the current assignment is smallest, then don't reassign] (mean of)

Note Title 8/23/2022

FACT: LLYOD'S ALGORITHM CONVERGES. ← Good news.

- Converged solution may not be "optimal"
- But produces "reasonable" clusters in practice.

QUESTIONS

- CONVERGENCE ?
- NATURE OF CLUSTERS ?
- INITIALIZATION ?
- CHOICE OF K ?

Note Title 8/29/2022

LLYOD'S ALGORITHM

QUESTIONS:

1. CONVERGENCE
2. NATURE OF CLUSTERS
3. INITIALIZATION
4. CHOICE OF K.

Note Title 8/29/2022

CONVERGENCE

- Does Lloyd's Algorithm converge? YES.

PROOF:

FACT 1: Let $x_1, x_2, \ldots, x_\ell \in \mathbb{R}^d$

$$v^* = \arg\min_{v \in \mathbb{R}^d} \sum_{i=1}^{\ell} \|x_i - v\|^2$$

[view this objective as a function of v, take derivative, set to 0 and solve]

Answer: $v^* = \frac{1}{\ell} \sum_{i=1}^{n} x_i$

Note Title 8/29/2022

- Say we are at iteration t of Lloyd's algorithm.

- CURRENT ASSIGNMENT.

$$Z_1^t, Z_2^t, \ldots, Z_n^t \in \{1, \ldots, k\}$$

$\mu_k^t \leftarrow$ Mean of cluster k in iteration t

- Say we update our assignments to

$$Z_1^{t+1}, Z_2^{t+1}, \ldots Z_n^{t+1} \in \{1, \ldots, k\}$$

Note Title 8/29/2022

$$\rightarrow \sum_{i=1}^{n} \| x_i - \mu^{t}_{z_i^{t+1}} \|^2 < \sum_{i=1}^{n} \| x_i - \mu^{t}_{z_i^{t}} \|^2$$

[By Algorithmic Choice]

$\sum_{i=1}^{n} \| x_i - \mu^{t}_{z_i^{t+1}} \|^2$: Mean of Cluster where x_i wants to go to

$F(z_1^t, \cdots, z_n^t)$

$\sum_{i=1}^{n} \| x_i - \mu^{t}_{z_i^{t}} \|^2$: Mean of current Cluster where x_i is assigned to.

$$\sum_{i=1}^{n} \| x_i - \mu^{t+1}_{z_i^{t+1}} \|^2 \leq F(z_1^{t+1}, \cdots, z_n^{t+1}) \leq \sum_{i=1}^{n} \| x_i - \mu^{t}_{z_i^{t+1}} \|^2 \leftarrow$$

Note Title 8/29/2022

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \| x_i - \mu_k^{t+1} \|^2 \cdot \mathbb{1}\left(z_i^{t+1} = k \right)$$

For every k

$$\sum_{i \in C_k} \| x_i - \mu_k^{t+1} \|^2 \leq \sum_{i \in C_k} \| x_i - v \|^2$$

[FACT 1]

$$\leq \sum_{i \in C_k} \| x_i - \mu_k^{t} \|^2$$

Note Title 8/29/2022

$\Rightarrow$ The objective function strictly reduces after each re-assignment.

$$F(z_1^{t+1}, \cdots z_n^{t+1}) < F(z_1^t, \cdots, z_n^t)$$

$\Rightarrow$ There are only "FINITE" number of assignments

$\Rightarrow$ Algorithm must converge!

Note Title 8/29/2022

NATURE OF CLUSTERS

$K=2$.

- Lloyd's algorithm produces 2 clusters with means μ_1 and μ_2

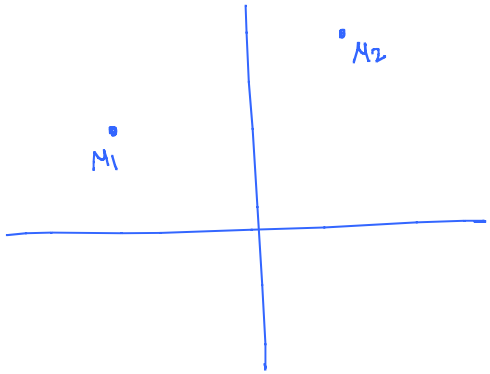
- What can we say about points assigned to cluster 1 ?

Note Title 8/29/2022

By Algorithm's construction

For points in Cluster 1,

$$\|x - \mu_1\|^2 \leq \|x - \mu_2\|^2$$



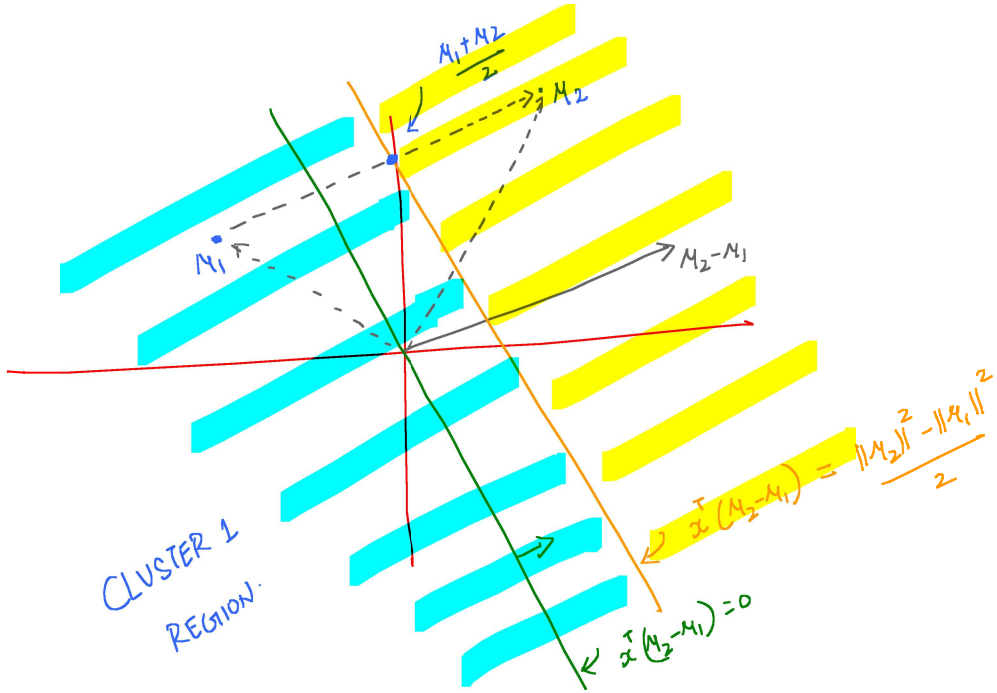
$$\|x\|^2 + \|\mu_1\|^2 - 2x^T\mu_1 \leq \|x\|^2 + \|\mu_2\|^2 - 2x^T\mu_2$$

$\forall$ x in cluster

$$x^T(\mu_2 - \mu_1) \leq \frac{\|\mu_2\|^2 - \|\mu_1\|^2}{2}$$

$$x^T w \leq b$$

Note Title 8/29/2022

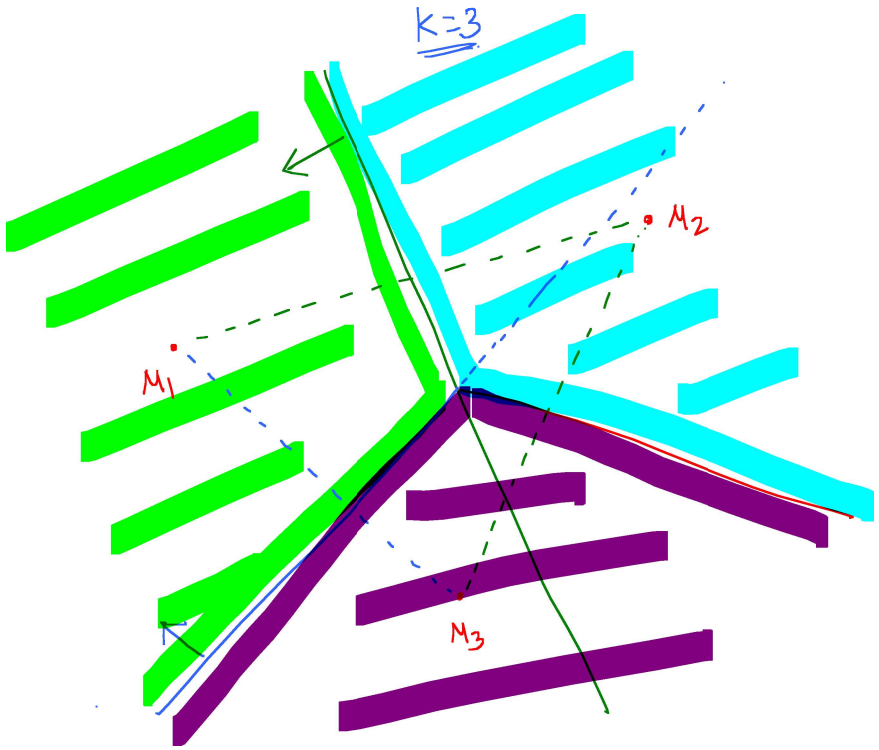


$$x^T(\mu_2-\mu_1) \le \frac{\|\mu_2\|^2-\|\mu_1\|^2}{2}$$

$$x = \frac{\mu_1+\mu_2}{2}$$

$$\left(\frac{\mu_1+\mu_2}{2}\right)^T(\mu_2-\mu_1) = \frac{\|\mu_2\|^2-\|\mu_1\|^2}{2}$$

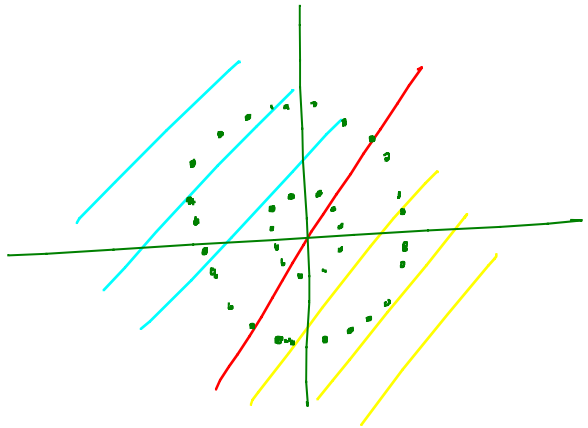
Note Title 8/29/2022



Cluster regions are intersection of Half Spaces

VORONOI REGIONS

Note Title 8/29/2022



How to fix?

- KERNELIZE K-MEANS
- SPECTRAL CLUSTERING.

Note Title 8/29/2022

INITIALIZATION

POSSIBILITIES

- Pick k-means uniformly at random from the dataset

K-MEANS ++

-> Choose first mean μ_1^0 (iteration) uniformly at random from $\{x_1, \ldots, x_n\}$

CHOICE OF K

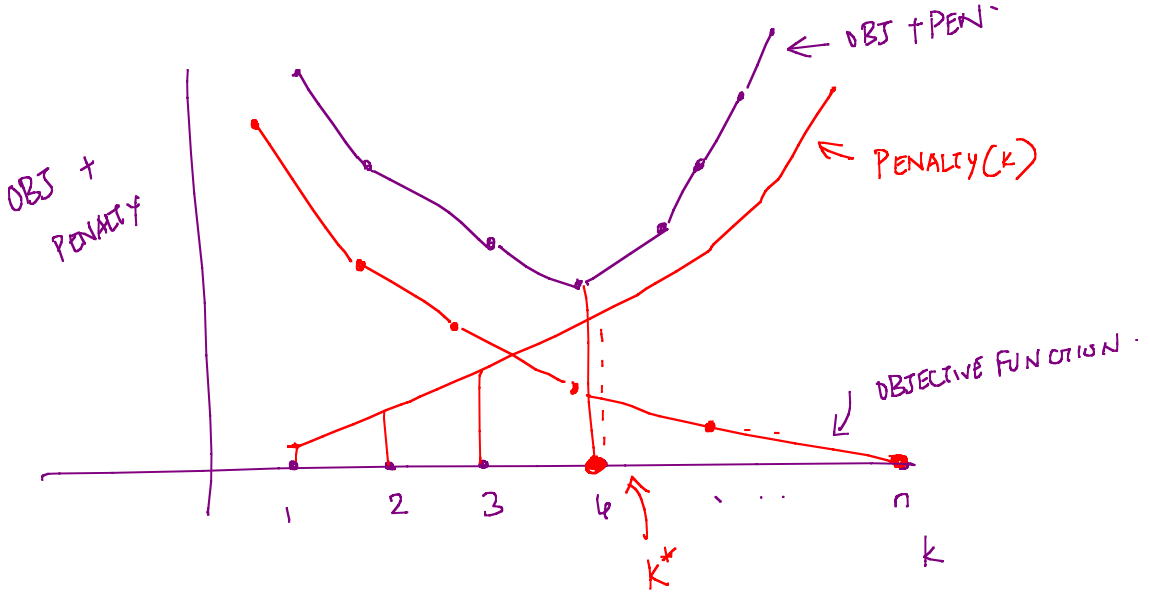
→ $F(z_1, \ldots, z_n) = \sum_{i=1}^{n} \|x_i - \mu_{z_i}\|^2$ $\qquad k = n$

→ want K to be as small as possible.

→ Penalize large values of K.

Find k that has the

Smallest Objective function value + Penalty(k)



Some common Criterion

A.I.C - Akaike Information Criterion

$$[2k - 2\log(L(\theta^*))]$$

B.I.C - Bayesian Information Criterion

$$[k\log(n) - 2\log(L(\theta^*))]$$

- CONVERGENCE - YES
- NATURE OF CLUSTERS - VORNOI REGIONS
- INITIALIZATION - K-MEANS++
- CHOICE OF K - OBJ + PENALTY(K).

Note Title 8/29/2022

→ for $\ell = 2, \dots, k$

Choose μ_ℓ^0 probabilistically proportional to score

$\forall x \quad S(x) = \min_{j=1,\dots,\ell-1} \|x - \mu_j^0\|^2$

x_1, x_2, x_3

10 20 30

1/6 1/3 1/2

GUARANTEE

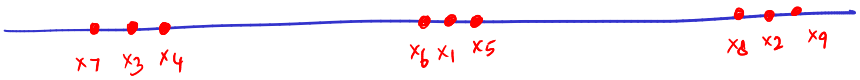
$$\mathbb{E}\left[\sum_{i=1}^{n} \|x_i - \mu_{z_i}\|^2\right] \leq O(\log k)\left[\min_{z_1,\dots,z_n} \sum_{i=1}^{n} \|x_i - \mu_{z_i}\|^2\right]$$

↑ over randomness of algorithm

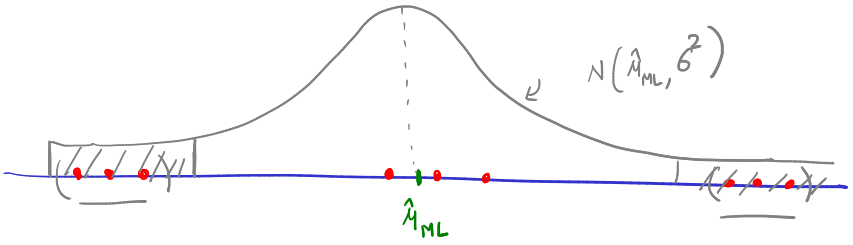
Unsupervised learning

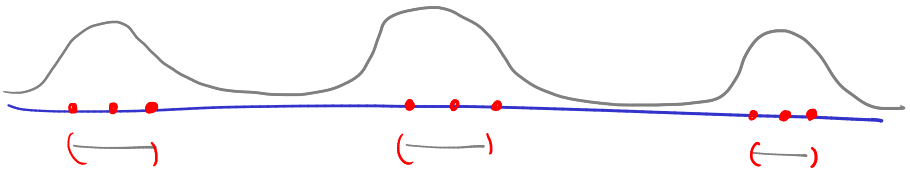
- Rep. learning – PCA / kernel PCA
- Clustering – LLyod's / k-means
- Estimation
  - Max. likelihood
  - Bayesian Modeling.

Today : Slightly complicated data Estimation for.



- What could be a good "generative" story ?





Want a density like above to explain this data.

A NEW GENERATIVE MODEL

MIXTURE OF GAUSSIANS

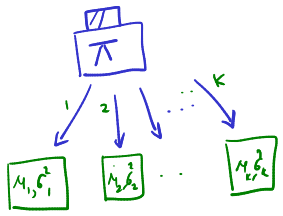
STEP 1: Pick which mixture a data point comes from.

STEP 2: Generate data point from that mixture.

STEP 1: Generate a mixture component among $\{1, \cdots, K\}$ $\quad z_i \in \{1, \cdots, K\}$

$$P(z_i = l) = \pi_l \qquad \left[\sum_{i=1}^{K} \pi_i = 1 \quad ; \quad 0 \le \pi_i \le 1 \;\; \forall i \right]$$

STEP 2: Generate $x_i \sim N(\mu_{z_i}, \sigma^2_{z_i})$



$\{x_1, \cdots, x_n\}$ → OBSERVED

$\{z_1, \cdots, z_n\}$ → UNOBSERVED/LATENT.

Latent variable Models

Parameters: $\pi = [\pi_1 \; \pi_2 \cdots \pi_K]$

$+ K \; (\mu_K, \sigma^2_K)$

Total: $2K + K - 1$

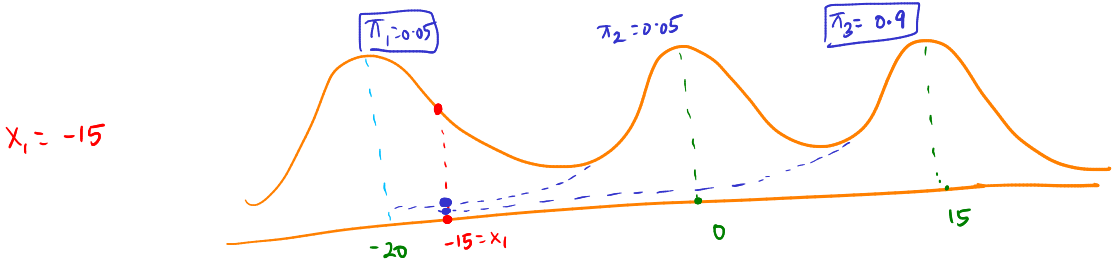
$3K - 1$

Max. Likelihood for GMM

$$L\left(\begin{matrix} \mu_1, \cdots, \mu_K \\ \sigma_1^2, \cdots, \sigma_K^2, \\ \pi_1, \cdots, \pi_K \end{matrix} \quad x_1, \cdots, x_n \right) = \prod_{i=1}^{n} f_{mix}\left(x_i \; ; \begin{matrix} \mu_1, \cdots, \mu_K \\ \sigma_1^2, \cdots, \sigma_K^2 \\ \pi_1, \cdots, \pi_K \end{matrix} \right)$$

$$= \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k \cdot f(x_i ; \mu_k, \sigma_k^2) \right]$$

NORMAL / Gaussian Density.



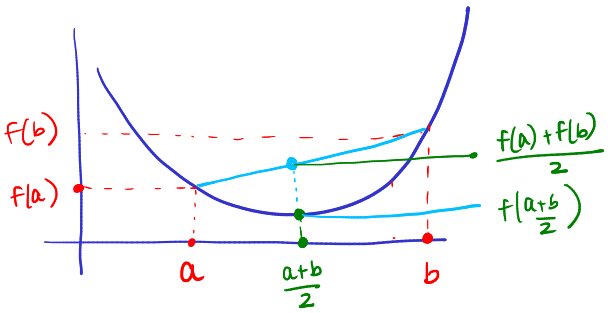
$$L(\theta) = \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k \frac{e^{-(x_i - \mu_k)^2 / 2\sigma_k^2}}{\sqrt{2\pi}\,\sigma_k} \right]$$

θ: all parameters

$$\log L(\theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k \frac{e^{-(x_i - \mu_k)^2 / 2\sigma_k^2}}{\sqrt{2\pi}\,\sigma_k} \right) \quad — (*)$$

- Not possible to solve this analytically.
- Need an alternate way to solve this efficiently!

Quick detour - Convex functions

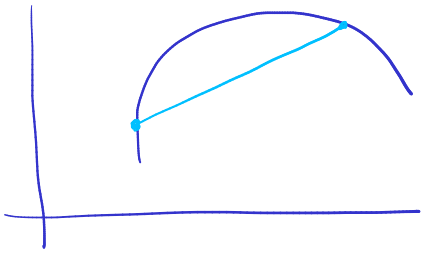


$\forall a, b$

$$f\left(\frac{a+b}{2}\right) \leq \frac{f(a)+f(b)}{2}$$

⇓

CONVEX FUNCTION

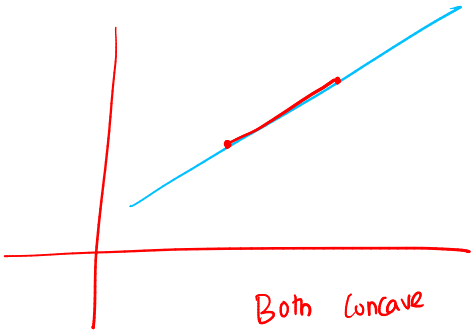


$\forall a, b$

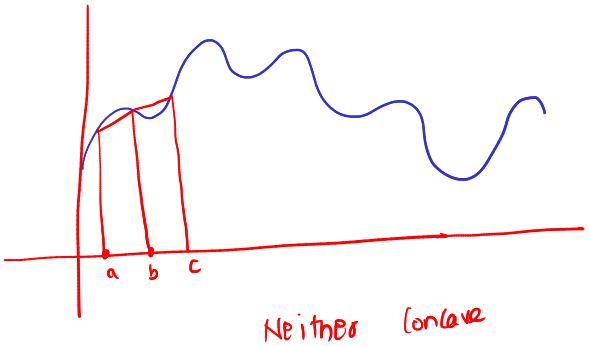
$$f\left(\frac{a+b}{2}\right) \geq \frac{f(a)+f(b)}{2}$$

⇓

CONCAVE FUNCTION

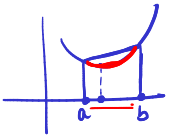


Both Concave and Convex.



Neither Concave nor Convex.

$$f\left(\frac{1}{2}\cdot a + \frac{1}{2} b\right) \leq \frac{1}{2} f(a) + \frac{1}{2} f(b)$$

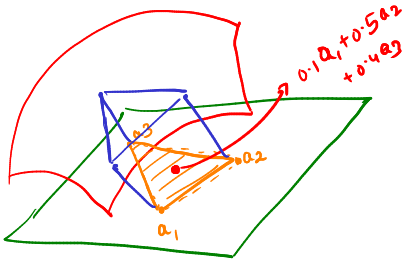


$$\Rightarrow \quad f\left(\lambda a + (1-\lambda) b\right) \leq \lambda f(a) + (1-\lambda) f(b) \qquad \lambda \in [0,1]$$

For Concave

$$f\left(\lambda_1 a_1 + \lambda_2 a_2 + \cdots + \lambda_k a_k\right) \geq \lambda_1 f(a_1) + \cdots + \lambda_k f(a_k)$$

$$\left[\sum_{i=1}^{k} \lambda_i = 1 \qquad 0 \leq \lambda_i \leq 1\right]$$



JENSEN'S INEQUALITY.

$$f\left(\sum_{k=1}^{K} \lambda_k a_k\right) \geq \sum_{k=1}^{K} \lambda_k f(a_k)$$

- Log is a concave function! [Why? Exercise]

- How can we exploit Jensen's for performing maximum likelihood.

Recall

$$\text{(*)} \quad \log L(\theta) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K}\left(\pi_k \, e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_k}\right)\right)$$

- INTRODUCE for every data point i, two parameters $\{\lambda_1^i, \ldots, \lambda_K^i\}$ s.t $\forall i \sum_{k=1}^{K} \lambda_k^i = 1$, $0 \le \lambda_k^i \le 1 \ \forall k, i$

$$\log L(\theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \lambda_k^i \left(\frac{\pi_k \, e^{\frac{-(x_i - \mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k}}{\lambda_k^i} \right) \right)$$

By Jensen's

$$\log L(\theta) \ge \text{modified\_log } L(\theta, \lambda)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k^i \log \left(\frac{\pi_k \, e^{\frac{-(x_i - \mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k}}{\lambda_k^i} \right)$$

- Note that the above modified log likelihood gives a lower bound for the true log likelihood at θ $\rightarrow \{\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2, \pi_1, \ldots, \pi_K\}$ for any choice of λ $\rightarrow \{\lambda_1^1, \ldots, \lambda_K^1, \lambda_1^2, \ldots, \lambda_K^2, \ldots, \lambda_1^n, \ldots \lambda_K^n\}$

- But what are we gaining?

Key insight:

- If we fix λ, it is easy to maximize w.r.t θ
- If we fix θ, it is easy to maximize w.r.t λ.

Fix λ and maximize over θ

$$\max_{\theta} \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k^i \left[\log \left(\pi_k \frac{e^{-(x_i - \mu_k)^2/2\sigma_k^2} \frac{1}{\sqrt{2\pi}\sigma_k}}{\lambda_k^i} \right) \right]$$

$$= \max_{\theta} \sum_{i=1}^{n} \sum_{k=1}^{K} \left[\lambda_k^i \log \pi_k - \lambda_k^i \frac{(x_i - \mu_k)^2}{2\sigma_k^2} - \lambda_k^i \log \sqrt{2\pi}\sigma_k \right]$$

Take derivative w.r.t μ, σ to get

$$\hat{\mu}_k^{MML} = \frac{\sum_{i=1}^{n} \lambda_k^i x_i}{\sum_{i=1}^{n} \lambda_k^i} \qquad \hat{\sigma}_k^{2\,MML} = \frac{\sum_{i=1}^{n} \lambda_k^i (x_i - \hat{\mu}_k^{MML})^2}{\sum_{i=1}^{n} \lambda_k^i}$$

$$\max_{\pi_1, \ldots, \pi_K} \sum_{i=1}^{n} \left(\sum_{k=1}^{K} \lambda_k^i \log \pi_k \right)$$

s.t $\sum_k \pi_k = 1 \; ; \; \pi_k \geq 0$

Can solve using method of Lagrange multipliers

$$\hat{\pi}_k^{MML} = \frac{\sum_{i=1}^{n} \lambda_k^i}{n}$$

Fixing λ, we get

$$\hat{\mu}_k^{MML} = \frac{\sum_{i=1}^{n} \lambda_k^i x_i}{\sum_{i=1}^{n} \lambda_k^i} \qquad \hat{\sigma}_k^{2\,MML} = \frac{\sum_{i=1}^{n} \lambda_k^i (x_i - \hat{\mu}_k^{MML})^2}{\sum_{i=1}^{n} \lambda_k^i}$$

$$\hat{\pi}_k^{MML} = \frac{\sum_{i=1}^{n} \lambda_k^i}{n}$$

- Fix θ and maximize λ

$$\sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k^i \log \left(\frac{\pi_k \, e^{\frac{-(x_i - \mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k}}{\lambda_k^i} \right)$$

$$= \sum_{i=1}^{n} \left[\sum_{k=1}^{K} \lambda_k^i \log(a_{ik}) - \lambda_k^i \log \lambda_k^i \right]$$

where $a_{ik} = \pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k}$

Fix any i,

$$\max_{\lambda_1^i, \dots \lambda_K^i} \sum_{k=1}^{K} \left[\lambda_k^i \log(a_k^i) - \lambda_k^i \log \lambda_k^i \right]$$

$$\text{s.t} \quad \sum_{k=1}^{K} \lambda_k^i = 1 \qquad 0 \le \lambda_k^i \le 1$$

Can be solved analytically

$$\hat{\lambda}_k^{i\,MML} = \frac{\left(\frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \right) \cdot \pi_k}{\sum_{l=1}^{K} \left(\frac{1}{\sqrt{2\pi}\sigma_l} e^{-\frac{(x_i - \mu_l)^2}{2\sigma_l^2}} \cdot \pi_l \right)} \leftarrow P(x_i)$$

$P(x_i | z_i = k)$, $P(z_i = k)$

$$P(z_i = k | x_i) = \frac{P(x_i | z_i = k) \cdot P(z_i = k)}{P(x_i)}$$

ALGORITHM - E.M ALGORITHM (1970s Dempster et al.)

→ Initialize θ^0 (0: iteration) $= \{ \mu_1^0, \dots, \mu_K^0, \sigma_1^{2\,0}, \dots \sigma_K^{2\,0}, \pi_1^0, \dots \pi_K^0 \}$

usually comes from Lloyd's

→ until convergence $(\| \theta^{t+1} - \theta^t \| \le \epsilon)$ (tolerance parameter)

$$\lambda^{t+1} = \arg\max_{\lambda} \text{modified\_log} L(\theta^t, \lambda)$$

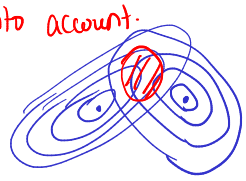
$$\theta^{t+1} = \arg\max_{\theta} \text{modified\_log} L(\theta, \lambda^{t+1})$$

→ end.

Maximization step

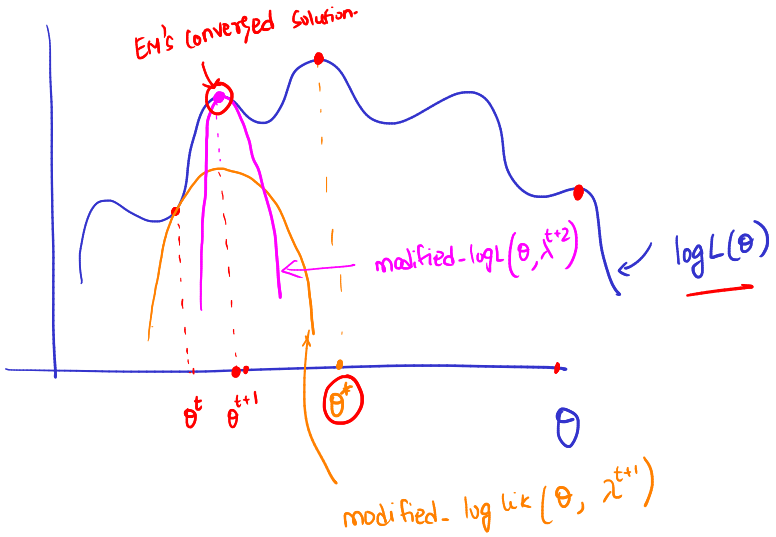
Expectation step

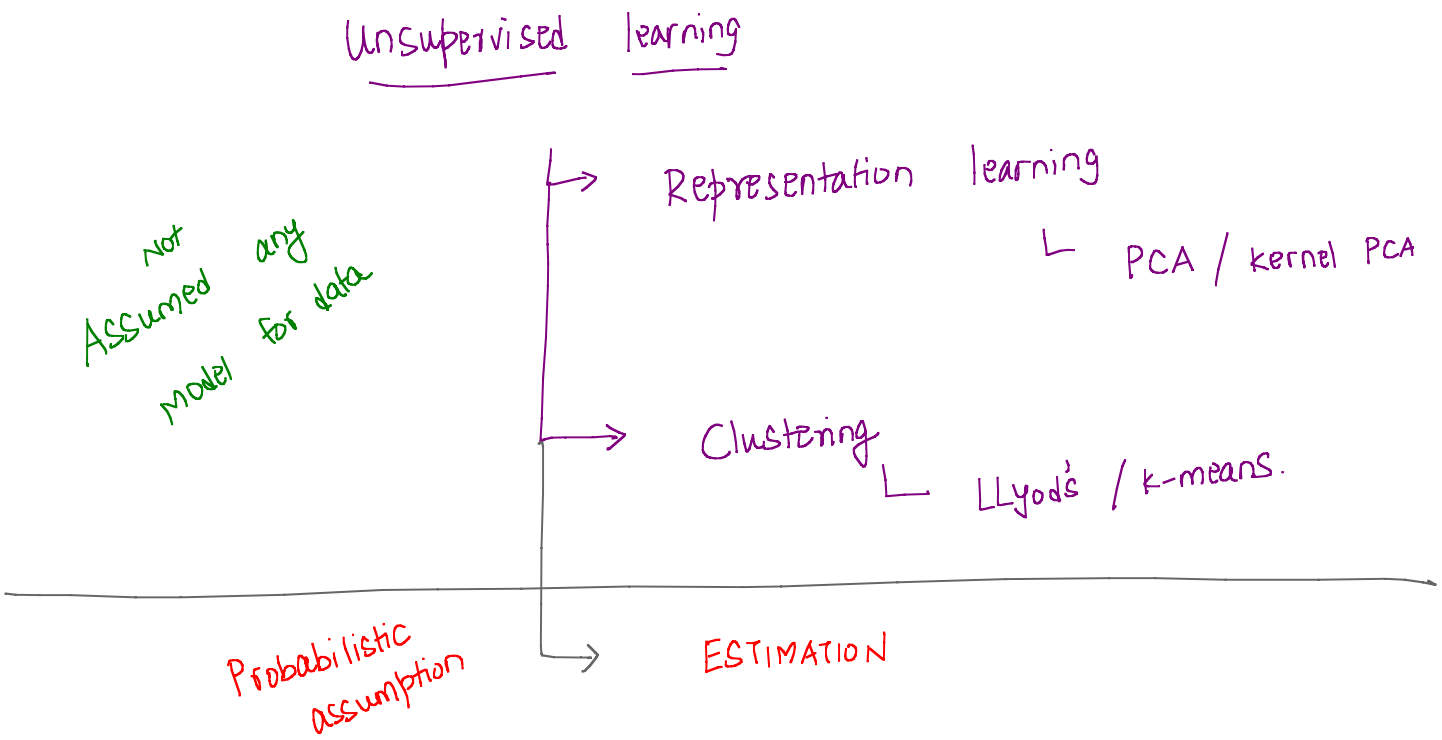
- EM produces "soft clustering"
- EM takes variances into account.



- EM clusters need not be Voronoi regions!

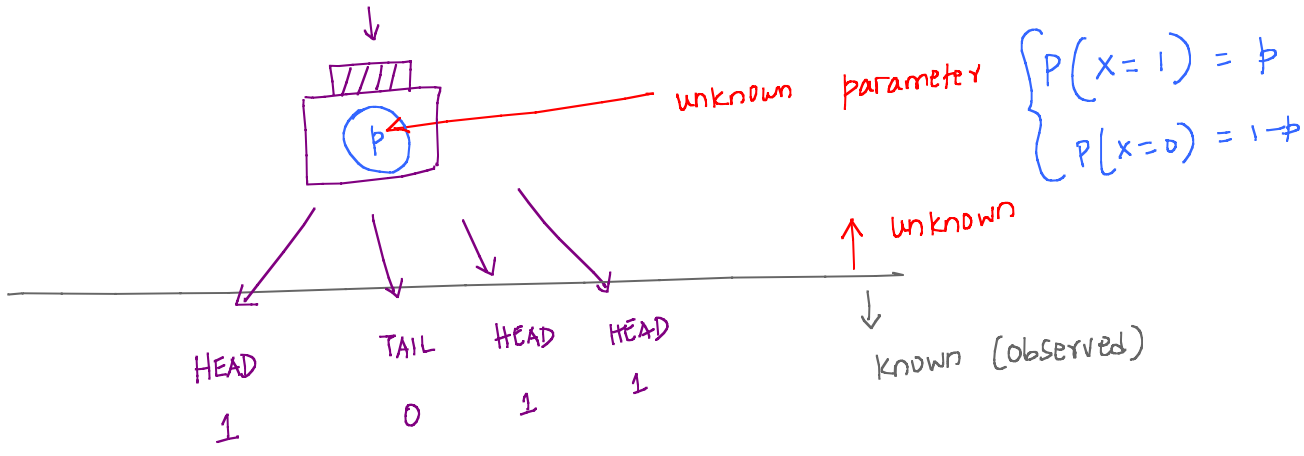
EM Converges to a local-maximum of log likelihood.




Unsupervised learning
Representation learning
PCA / kernel PCA
Clustering
LLyod's / k-means.
Not Assumed any model for data
Probabilistic assumption
ESTIMATION

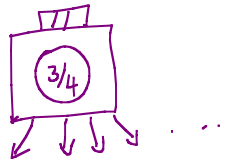
" There is some probabilistic mechanism that generates data about which we don't know "something". Given data, find/estimate what we don't know"

- OBSERVE data
- "ASSUME" a probabilistic model that generates data
- ESTIMATE unknown parameters using data.



OBSERVE $\{1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1\}$

ESTIMATE : $9/12 = \boxed{3/4}$



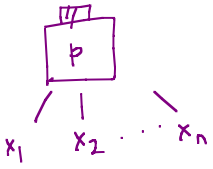
GUESS = $2/3$? Y

= 0.0001 ? Y

= 0 ? N

= 1 ? N

ASSUMPTIONS



OBSERVATIONS ARE

(i) INDEPENDENT

(ii) IDENTICALLY DISTRIBUTED

INDEPENDENCE

$$P(x_i / x_j) = P(x_i) \quad \forall i \neq j$$

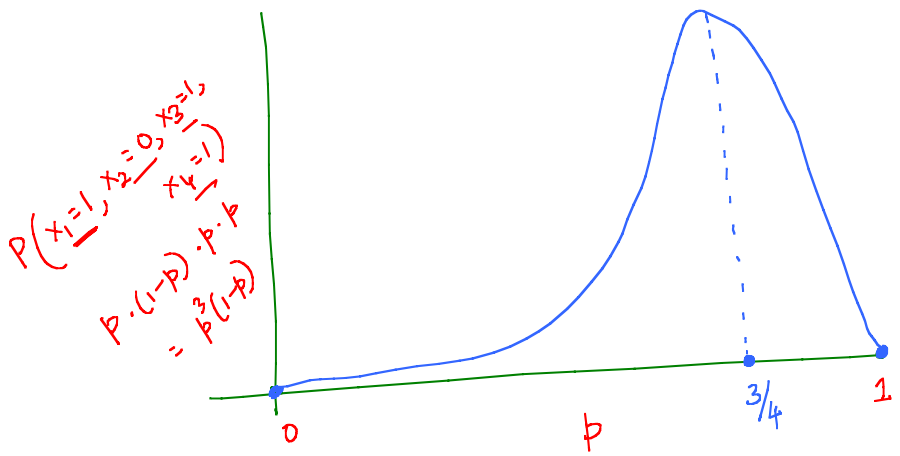
IDENTICAL DISTRIBUTION

$$P(x_i = 1) = P(x_j = 1) = p \quad \forall i, j$$

LIKELIHOOD FUNCTION

(Eg) : $\{1, 0, 1, 1\}$

$3/4$



FISHER'S PRINCIPLE OF MAXIMUM LIKELIHOOD

$$L\left(p ; \{x_1, x_2, \ldots, x_n\}\right) = P\left(x_1, x_2, \ldots, x_n ; p\right)$$ (p: underlying parameter.)

$$= P(x_1; p) \cdot P(x_2; p) \cdots P(x_n; p) \quad \text{[Independence]}$$

$$= \prod_{i=1}^{n} p^{x_i} (1-p)^{(1-x_i)}$$

If $x_i = 1$: $\Rightarrow p^1 (1-p)^0 = p$

If $x_i = 0$: $\Rightarrow p^0 (1-p)^{1-0} = 1-p$

$$\hat{p}_{ML} = \arg\max_p \prod_{i=1}^{n} p^{x_i} (1-p)^{1-x_i}$$

$$= \arg\max_p \log\left(\prod_{i=1}^{n} p^{x_i} (1-p)^{1-x_i} \right) \quad \text{[log is monotonic increasing]}$$

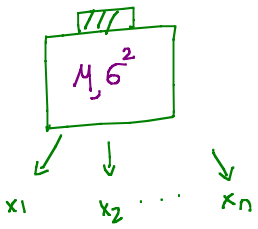
$$= \underset{p}{\arg\max} \sum_{i=1}^{n} \left[x_i \log p + (1-x_i) \log (1-p) \right]$$

Take derivative of $\log L(p)$, set it to 0 to get

$$\hat{p}_{ML} = \frac{1}{n} \sum_{i=1}^{n} x_i$$ ← Fraction of 1's

---

Data = $\{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}$ $\forall i$



$x_i \sim$ Gaussian (μ, σ^2) $\forall i$

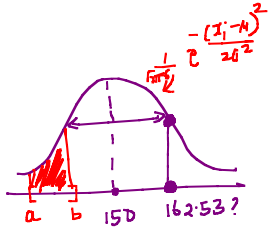
μ -> unknown ; σ^2 -> known.

$$L\left(\mu, \sigma^2, \{x_1, \cdots, x_n\}\right) = P\left(x_1, \cdots, x_n ; \mu, \sigma^2\right)$$

$$= \prod_{i=1}^{n} P\left(x_i ; \mu, \sigma^2\right) \quad \times$$

$$P(x_i ; \mu, \sigma^2) = 0$$

$x_i = 162.53$ cm.



$$L\left(\mu, \sigma^2, \{x_1, \cdots, x_n\}\right) = f_{x_1, \cdots x_n}\left(x_1, \cdots, x_n ; \mu, \sigma^2\right)$$

$$= \prod_{i=1}^{n} f_{x_i}\left(x_i ; \mu, \sigma^2\right)$$

$$= \prod_{i=1}^{n} \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \right]$$

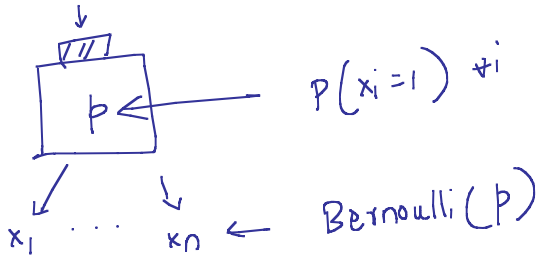
$$\log L\left(\mu, \sigma^2, \{x_1, \cdots, x_n\}\right) = \sum_{i=1}^{n} \left[\log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \frac{(x_i - \mu)^2}{2\sigma^2} \right]$$

$$\hat{\mu}_{ML} = \underset{\mu}{\operatorname{argmax}} \sum_{i=1}^{n} -(x_i - \mu)^2$$

$$\boxed{\hat{\mu}_{ML} = \frac{1}{n} \sum_{i=1}^{n} x_i}$$

Consider the Coin examples.

BAYESIAN MODELING



Let's say Someone says: "I believe the bias p is somewhere close to 1"

We may have "HUNCH" about parameters

Goal: Incorporate "hunch / belief" about parameters of interest into the estimation procedure.

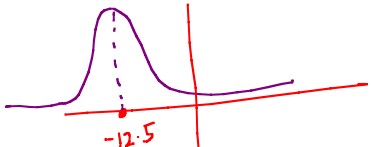
APPROACH : Think of the parameter to estimate as a "random" variable.

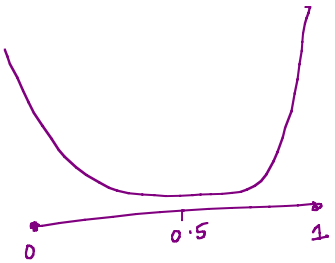
EARLIER

μ

p

NOW





HUNCH → Codified using a probability distribution over θ

$P(\theta)$

PRIOR

⇓ DATA.

⇓

$P(\theta / \text{DATA})$

UPDATED HUNCH

Codified using a prob. distribution

POSTERIOR

Bayes law

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$$

A → Parameters θ

B → DATA $\{x_1, \dots, x_n\}$

$$P(\theta \mid \{x_1, \dots, x_n\}) = \left(\frac{P(\{x_1, \dots, x_n\} \mid \theta)}{P(\{x_1, \dots, x_n\})} \right) \cdot P(\theta)$$

$P(\theta \mid \{x_1, \dots, x_n\})$: POSTERIOR

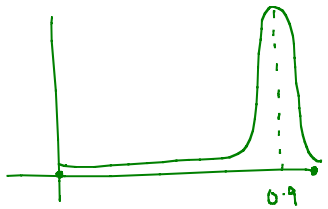
$P(\{x_1, \dots, x_n\} \mid \theta)$: LIKELIHOOD

$P(\{x_1, \dots, x_n\})$: EVIDENCE — Does not depend on θ.

$P(\theta)$: PRIOR

EXAMPLE

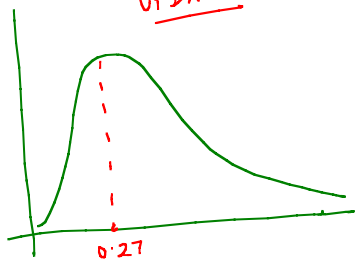
HUNCH

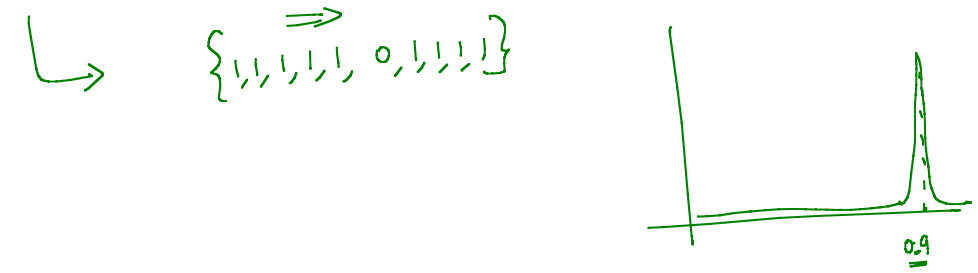


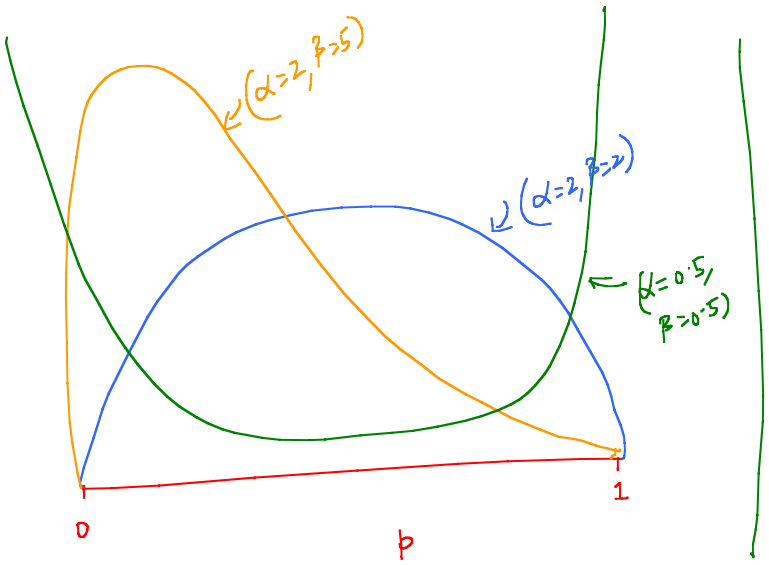
DATA

$\{0, 0, 0, 0, 0, 1, 1, 0, 0, 0\}$

UPDATED HUNCH







DATA - Bernoulli (p)

PRIOR ? $P(\theta)$

BETA PRIOR

$$f(p; \alpha, \beta) = \begin{cases} \dfrac{p^{\alpha-1}(1-p)^{\beta-1}}{Z} & \forall\, p \in [0,1] \end{cases}$$

$$P(\theta|_{DATA}) \propto P(DATA/\theta) \cdot P(\theta)$$

$$f_{p/DATA}(p) \propto \underbrace{\left[\prod_{i=1}^{n} p^{x_i}(1-p)^{(1-x_i)}\right]}_{\text{LIKELIHOOD}} \cdot \underbrace{\left[p^{\alpha-1}(1-p)^{\beta-1}\right]}_{\text{PRIOR}}$$

$$f_{p/DATA}(p) \propto p^{\sum x_i + \alpha - 1}(1-p)^{\sum(1-x_i)+\beta-1}$$

↳ same functional form as the PRIOR!!

BETA PRIOR (α, β) $\xrightarrow[\text{Bernoulli}]{\text{DATA}}$ BETA POSTERIOR $(\alpha + n_h, \beta + n_t)$

one possible guess $= \frac{\alpha + n_h}{\alpha + n_h + \beta + n_t} = \frac{\alpha + n_h}{(\alpha+\beta) + n}$

$\mathbb{E}[\text{Posterior}] = \mathbb{E}[\text{Beta}(\alpha + n_h, \beta + n_t)] =$

MAP Estimator - Maximum Aposteriori Estimator

$\hat{p}_{MAP}$

SUPERVISED LEARNING

Input: $\{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$ ← Features / Attributes

$\{y_1, \ldots, y_n\}$ ← LABELS (supervision)

- $\{+1, -1\}$, $\{0, 1\}$ → BINARY CLASSIFICATION (Spam vs non-spam)
- $\{0, 1, \ldots, K\}$ → MULTICLASS CLASSIFICATION [digit classification]
- $\in \mathbb{R}$ → REGRESSION [Rainfall prediction]

REGRESSION

INPUT / TRAINING DATA: $\{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$

$\{y_1, \ldots, y_n\}$ $y_i \in \mathbb{R}$.

Goal: Learn $h : \mathbb{R}^d \to \mathbb{R}$

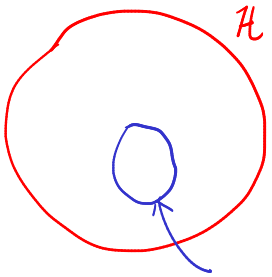
- How do we measure "goodness" of a function $h : \mathbb{R}^d \to \mathbb{R}$
- $\text{error}(h) = \sum_{i=1}^{n} (h(x_i) - y_i)^2$
- How small can this error be? 0
- Which h achieves zero error?

$h(x_i) = y_i \quad \forall i$

What is the problem?

- By "memorizing", we can get zero error on training data
- What we care is about test performance.
- Impose "STRUCTURE" to reduce search space.

SIMPLEST STRUCTURE – LINEAR STRUCTURE.



← set of all functions $h: \mathbb{R}^d \to \mathbb{R}$.

$$\mathcal{H}_{linear} = \left\{ h_w : \mathbb{R}^d \to \mathbb{R} \text{ s.t } h_w(x) = w^T x \text{ for all } w \in \mathbb{R}^d \right\}$$

GOAL:

$$\min_{h_w \in \mathcal{H}_{linear}} \sum_{i=1}^{n} \left(h_w(x_i) - y_i\right)^2$$

(or) equivalently.

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2$$ → LINEAR REGRESSION

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2 = \|X^T w - y\|_2^2$$

$$X^T = \begin{bmatrix} — x_1 — \\ — x_2 — \\ \vdots \\ — x_n — \end{bmatrix}_{n \times d} \quad \begin{bmatrix} | \\ w \\ | \end{bmatrix}_{d \times 1} \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}_{n \times 1}$$

$$\min_{w \in \mathbb{R}^d} \left(X^T w - y\right)^T \left(X^T w - y\right)$$ ← unconstrained Quadratic (in w) Optimisation problem

Solution: Take derivative (gradient) and set to zero.

$$f(w) = \left(X^T w - y\right)^T \left(X^T w - y\right)$$

$$\nabla f(w) = 2\left(XX^T\right)w - 2(Xy)$$

Solution satisfies $\boxed{\left(XX^T\right)w^* = Xy}$

OBSERVATION

Like PCA, w^* depends on a "covariance" like matrix. But it also involves y.

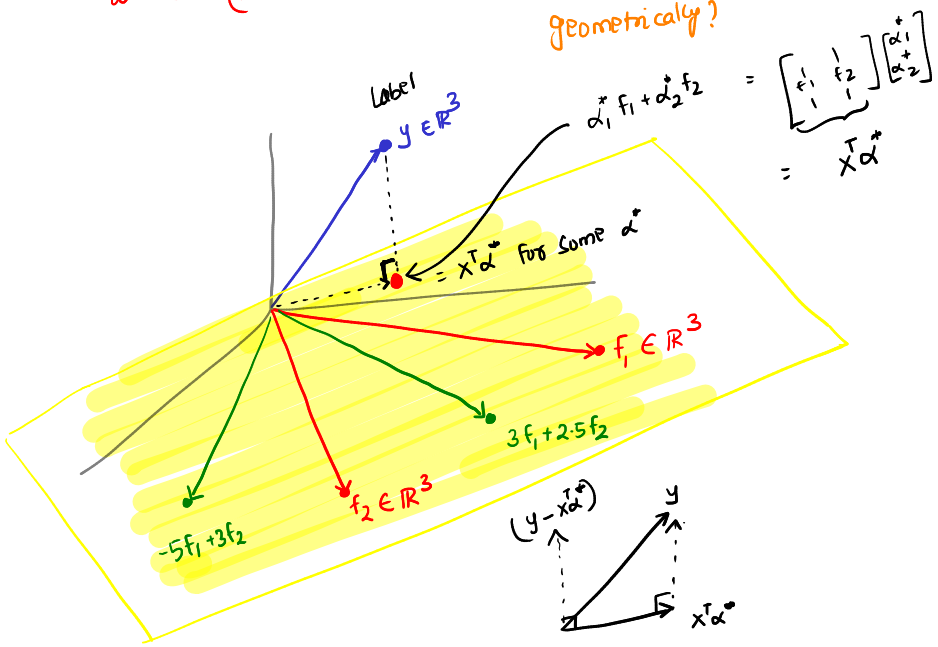
$$w^* = (XX^T)^\dagger Xy$$ ← Pseudo-inverse.

- → Lin reg has closed form solution
- → GEOMETRIC VIEW ?
- → COMPUTATIONAL CONSIDERATIONS ?
- → NON-LINEAR FEATURE → LABEL RELATIONSHIP ?
- → PROBALISTIC VIEW ?

$$w^* = (XX^T)^\dagger Xy$$ ← How can we interpret this geometrically?

f_1 height, f_2 weight

$$\begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix}\begin{bmatrix} [\,] & [\,] \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}\Bigg| \begin{matrix} d=2 \\ n=3 \end{matrix}$$



$$(y - X^T\alpha^*)^T (X^T\alpha^*) = 0$$

$$y^T X^T \alpha^* - \alpha^{*T}(XX^T)\alpha^* = 0 \quad \text{— (1)}$$

Recall, $w^* = (XX^T)^\dagger Xy$

Substituting $w^* = \alpha^*$ on L.H.S, we get

$$y^T X^T\left((XX^T)^\dagger Xy\right) - \left((XX^T)^\dagger Xy\right)^T (XX^T)(XX^T)^\dagger Xy$$

$$= 0$$

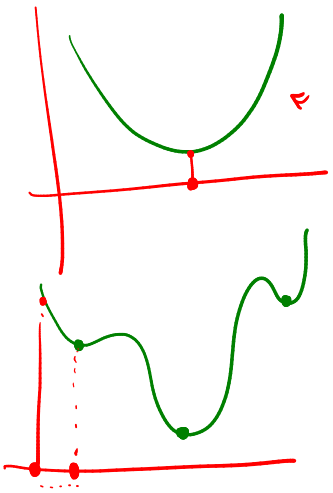
[use XX^T is Symmetric]

CONCLUSION: $X^T w^*$ is the PROJECTION OF the Labels onto the subspace spanned by the features.

COMPUTATIONAL CONSIDERATIONS

$$w^* = (XX^T)^{-1} X y$$

inverse computation is expensive if d is large $O(d^3)$



- We know w^* is the solution of an unconstrained optimization
- we can apply GRADIENT DESCENT.

$$w^{t+1} = w^t - \eta^t \nabla f(w^t)$$

η^t: scalar stepsize. $\nabla f(w^t)$: gradient at w^t

$$f(w) = \|Xw - y\|^2 = \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

$$\nabla f(w) = 2(XX^T)w - 2Xy \quad \text{[verify this]}$$

Gradient descent update for Linear regression

$$w^{t+1} = w^t - \eta^t \left[2(XX^T)w - 2(Xy) \right]$$

- what if n is large [hundreds of millions], might want to avoid XX^T
- How to adapt gradient descent ?

STOCHASTIC GRADIENT DESCENT

for $t = 1, \dots, T$

- At each step, sample a bunch of data points (B) uniformly at random from the set of all points

- PRETEND this sample is the entire dataset and take a gradient step w.r.t it

$$2\left(\tilde{X}\tilde{X}^T w^t - \tilde{X}\tilde{y}\right)$$

↳ Manageable because $\tilde{X} \in \mathbb{R}^{d \times B}$ ↳ # Sampled points.

end

After T rounds, use

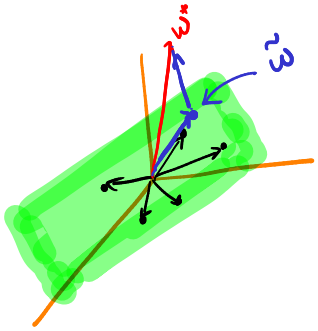
$$w_{SGD}^T = \frac{1}{T}\sum_{i=1}^{T} w^t$$

as opposed to w^T as in Standard Gradient descent

Guaranteed to converge to optima with high probability

NON-LINEAR REGRESSION

$$w^* = \left(XX^T\right)^{\dagger} Xy$$



- w^* must lie in the span of data points.

$$\sum_{i=1}^{n}\left(w^{*T}x_i - y_i\right)^2 = \sum_{i=1}^{n}\left(\tilde{w}^T x_i - y_i\right)$$

$$w^* = \tilde{w} + \tilde{w}_\perp$$

$$\forall i \quad w^{*T}x_i = \left(\tilde{w} + \tilde{w}_\perp\right)^T x_i = \tilde{w}^T x_i + \underbrace{\tilde{w}_\perp^T x_i}_{0}$$

$$w^* = \arg\min \sum_{i=1}^{n} (\phi(x_i) - y_i)^2$$

$w^* = X\alpha^*$ for some $\alpha^* \in \mathbb{R}^n$

$= (XX^T)^\dagger Xy$

$X\alpha^* = (XX^T)^\dagger Xy$

$(XX^T) X\alpha^* = (XX^T)(XX^T)^\dagger Xy$

$(XX^T) X\alpha^* = Xy$

$X^T(XX^T) X\alpha^* = X^T Xy$

$(X^TX)^2 \alpha^* = (X^TX)y$

$:= K$

$\Rightarrow \quad K^2\alpha^* = Ky$

$\alpha^* = K^{-1}y$

$\in \mathbb{R}^{n \times n}$

KERNEL - REGRESSION

PREDICTION

for some $x_{test} \in \mathbb{R}^d$

$w^{*T} \phi(x_{test})$

$= \left(\sum_{i=1}^{n} \alpha_i^* \phi(x_i)\right)^T \phi(x_{test})$

$= \sum_{i=1}^{n} \alpha_i^* K(x_i, x_{test})$

How important is i^{th} point towards w^*

How similar is x_{test} to x_i

PROBABILISTIC VIEW OF LINEAR REGRESSION

$x \in \mathbb{R}^d \quad y \in \mathbb{R}$

Dataset $\{(x_1, y_1) \ldots, (x_n, y_n)\}$

$y|x \sim w^T x + \epsilon$

w: unknown but fixed $\in \mathbb{R}^d$

ϵ: Noise $N(0, \sigma^2)$ → Gaussian.

- Can view this as an "ESTIMATION" problem
- Solution approach – Maximum Likelihood.

Likelihood

$$\mathcal{L}\left(w;\ \begin{matrix} x_1,\dots,x_n \\ y_1,\dots,y_n \end{matrix}\right) = \prod_{i=1}^{n} e^{-\frac{(w^T x_i - y_i)^2}{2\sigma^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma}$$

$$\log \mathcal{L}\left(w;\ \begin{matrix} x_1,\dots,x_n \\ y_1,\dots,y_n \end{matrix}\right) = \sum_{i=1}^{n} -\frac{(w^T x_i - y_i)^2}{2\sigma^2} \cdot \frac{1}{\sqrt{2\pi}\sigma}$$

equivalently

$$\max_w \sum_{i=1}^{n} -(w^T x_i - y_i)^2$$

$$\equiv \min_w \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

$$\hat{w}_{ML} = w^* = (XX^T)^{\dagger} Xy$$

CONCLUSION: Maximum Likelihood estimator assuming ZERO MEAN GAUSSIAN NOISE is same as LINEAR REGRESSION with SQUARED ERROR!

What else have we gained?

- Can study properties of estimators $\hat{w}_{ML}$!

$w^* = \hat{w}_{ML}$ ← max likelihood $= \boxed{(XX^T)^{\dagger} X y}$ ← random

$y/x = w^T x + \epsilon \rightsquigarrow N(0, \sigma^2)$

$N(w^T x, \sigma^2)$

$\{(x_1, y_1), \ldots (x_n, y_n)\}$

$w^T x_i + \epsilon_i$

$w \in \mathbb{R}^d \; ; \hat{w}_{ML} \in \mathbb{R}^d$

Want a way to understand how good $\hat{w}_{ML}$ is in estimating w

Mean Squared error → $\mathbb{E}\left[\|\hat{w}_{ML} - w\|^2\right] = \sigma^2 \cdot \text{trace}\left((XX^T)^{-1}\right)$

$\mathbb{E}$: over randomness in y

$A = \begin{bmatrix} a_1 & & \\ & a_2 & \\ & & \ddots \\ & & & a_d \end{bmatrix}$

$\text{tr}(A) = \sum_{i=1}^{d} a_i = \sum_{i=1}^{d} \lambda_i$

(λ_i: i^{th} Eigenvalue of A)

$\boxed{\text{trace}\left((XX^T)^{-1}\right)}$

Let Eigenvalues of (XX^T) be $\{\lambda_1, \ldots, \lambda_d\}$

Eigenvalues of $(XX^T)^{-1}$ are $\left\{\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_d}\right\}$

Mean sq. error $(\hat{w}_{ML})$

$\mathbb{E}\left(\|\hat{w}_{ML} - w\|^2\right) = \sigma^2 \left(\sum_{i=1}^{d} \frac{1}{\lambda_i}\right)$

Consider the following estimator:

$\hat{w}_{new} = \left(XX^T + \lambda I\right)^{-1} X y$

$\lambda \in \mathbb{R}_+$; $I \in \mathbb{R}^{d \times d}$

$\hat{w}_{ML} = (XX^T)^{-1} X y$

For some matrix A, let Eigenvalues be $\{\lambda_1, \ldots, \lambda_d\}$

. What are Eigenvalues of $A + \lambda I$? $\{\lambda_1 + \lambda, \ldots, \lambda_d + \lambda\}$

$Av_i = \lambda_i v_i$

$(A + \lambda I) v_i = Av_i + \lambda v_i$

$= \lambda_i v_i + \lambda v_i$

$= (\lambda_i + \lambda) v_i$

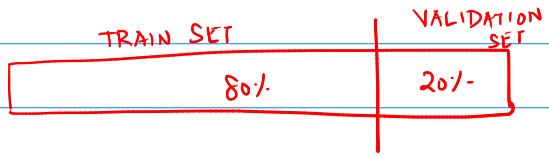
$\text{trace}\left((XX^T + \lambda I)^{-1}\right) = \left(\sum_{i=1}^{n} \frac{1}{\lambda_i + \lambda}\right)$

EXISTENCE Thm : (Informal)

$\exists \lambda \in \mathbb{R}$ s.t

$\vec{w}_{new} = (XX^T + \lambda I)^{-1} Xy$ has lesser mean sq. error than $\vec{w}_{ML}$

In practice, find λ by CROSS VALIDATION



- Train on the training set and check for error on validation set.
- Pick λ that gives least error.

K-FOLD CROSS VALIDATION

- Train on folds $\{F_1, \dots F_{i-1}, F_{i+1}, \dots F_k\}$
- Validate on F_i

- Pick λ that gives least average error.

LEAVE ONE OUT CROSS VALIDATION

$\vec{w}_{new} = (XX^T + \lambda I)^{-1} Xy$

IS there an alternate way to understand $\vec{w}_{ML}$?

BAYESIAN MODELING

- NEED A PRIOR on w i.e., $P(\vec{w})$ ($\vec{w} \in \mathbb{R}^d$)

LIKELIHOOD $y|x \sim N(w^T x, 1)$ — For simplicity. Can use σ^2 as well.

A CHOICE FOR PRIOR

$w \sim N(0, \tau^2 I)$ — $0 \in \mathbb{R}^d$; $\tau^2 I$ is the covariance matrix $\mathbb{R}^{d \times d}$.

$$\begin{bmatrix} \tau^2 & & 0 \\ & \tau^2 & \\ 0 & & \tau^2 \end{bmatrix}$$

As usual,

$$P\left(w \mid \{(x_1,y_1) \cdots (x_n,y_n)\}\right) \propto P\left(\{(x_1,y_1),\cdots,(x_n,y_n)\} \mid w\right) \cdot P(w)$$

$$\propto \left(\prod_{i=1}^{n} e^{-\frac{(y_i - w^T x_i)^2}{2}}\right) \cdot \left(\prod_{i=1}^{d} e^{-\frac{(w_i - 0)^2}{2\tau^2}}\right)$$

$$\left(\prod_{i=1}^{d} e^{-\frac{w_i^2}{2\tau^2}}\right) = e^{-\sum_{i=1}^{d} \frac{w_i^2}{2\tau^2}}$$

$$\propto \left(\prod_{i=1}^{n} e^{-\frac{(y_i - w^T x_i)^2}{2}}\right) \cdot e^{-\frac{\|w\|^2}{2\tau^2}}$$

How will the MAP estimate look like?

$$\hat{w}_{MAP} = \arg\max_w \sum_{i=1}^{n} -\frac{(y_i - w^T x_i)^2}{2} - \frac{\|w\|^2}{2\tau^2}$$

$$\hat{w}_{MAP} = \arg\min_w \frac{1}{2} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \frac{1}{2\tau^2}\|w\|^2 \rightarrow f(w)$$

Take gradient, set it to 0 to solve for $\hat{w}_{MAP}$.

$$\nabla F(w) = (XX^T)w - Xy + \frac{w}{\nu^2}$$ [verify]

$$\hat{w}_{MAP} = \left(XX^T + \frac{1}{\nu^2} I\right)^{-1} Xy$$

CROSS VALIDATE in PRACTICE.

CONCLUSION: MAP ESTIMATION for linear regression with a Gaussian prior $N(0, \nu^2 I)$ for w is equivalent to "NEW" estimator we used earlier.

LINEAR REGRESSION

$$\hat{w}_{ML} = \arg\min_w \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

RIDGE REGRESSION

$$\hat{w}_R = \arg\min_w \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|^2$$

$\hat{w}_R$: RIDGE

$\sum_{i=1}^{n} (w^T x_i - y_i)^2$: LOSS

$\lambda \|w\|^2$: REGULARIZER ($\lambda = \frac{1}{\nu^2}$)

---

f_1 height	f_2 weight	f_3 2 height + 3 weight	label: 3 height + 4 weight
1	1	1	
0	c_1	c_2	
3	4	0	

Linear regression / Ridge regression

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|^2$$

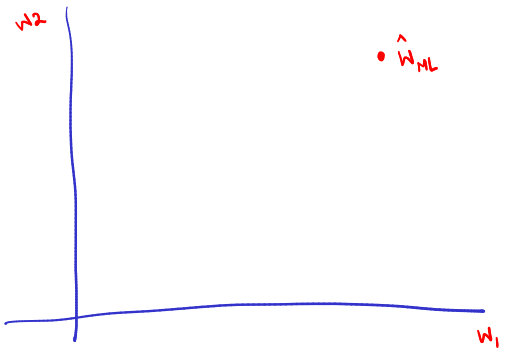
$\lambda \|w\|^2 \rightarrow \sum_{i=1}^{d} w_i^2 = \|w\|^2$

$\downarrow$

REGULARIZATION

PARAMETER SPACE

$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$



Where is $\hat{w}_R$ -> solution of the ridge regression problem?

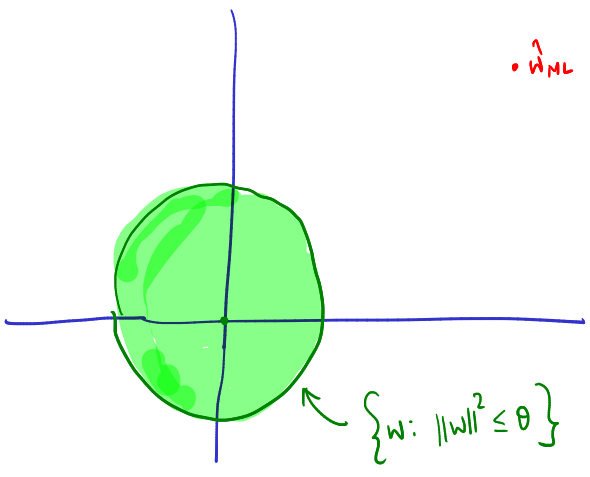
$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|^2 \quad \text{(A)}$$

$\equiv$ is equivalent to

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (w^T x_i - y_i)^2 \quad \text{(B)}$$

s.t $\|w\|^2 \leq \theta$ (depends on λ)

For every choice of $\lambda > 0$, $\exists \theta$ s.t the optimal solutions of problems (A) and (B) coincide.



$\|w\|^2 \leq \theta$

$\Rightarrow w_1^2 + w_2^2 \leq \theta$

- What is the loss / error / objective function value of linear regression at $\hat{w}_{ML}$

$$\sum_{i=1}^{n} \left(\hat{w}_{ML}^T x_i - y_i \right)^2 = f\left(\hat{w}_{ML}\right)$$

Consider the set of all w s.t

$$f(w) = f(\hat{w}_{ML}) + c \qquad c > 0$$

$$S_c = \left\{ w : \; f(w) = f(\hat{w}_{ML}) + c \right\}$$

i.e., every $w \in S_c$ satisfies

$$\underbrace{\| X^T w - y \|^2}_{f(w)} = \underbrace{\| X^T \hat{w}_{ML} - y \|^2}_{f(\hat{w}_{ML})} + c$$

On simplification [Please do this], one gets

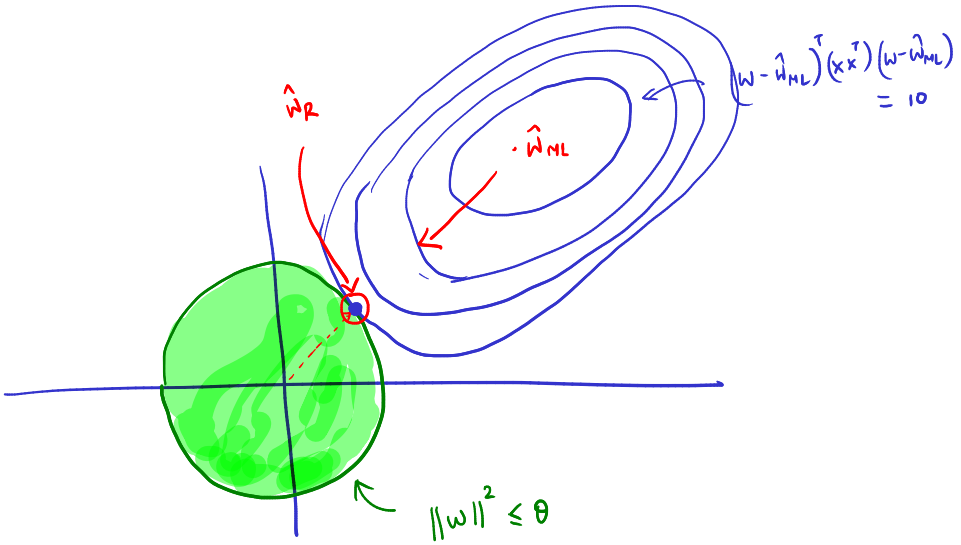
$$\underbrace{(w - \hat{w}_{ML})^T (XX^T)(w - \hat{w}_{ML})}_{} = c'$$

Some constant that depends on c, (XX^T), $\hat{w}_{ML}$ and not on w

If $XX^T = I$,

$$(w - \hat{w}_{ML})^T I (w - \hat{w}_{ML}) = c'$$

$$\| w - \hat{w}_{ML} \|^2 = c'$$



Conclusions:

- Ridge regression pushes feature values towards 0. But does not necessarily make it 0.

- An alternate way to regularize would mean be using $\|\cdot\|_1$ norm instead of $\|\cdot\|_2^2$ norm

$$\|w\|_1 = \sum_{i=1}^{d} |w_i|$$

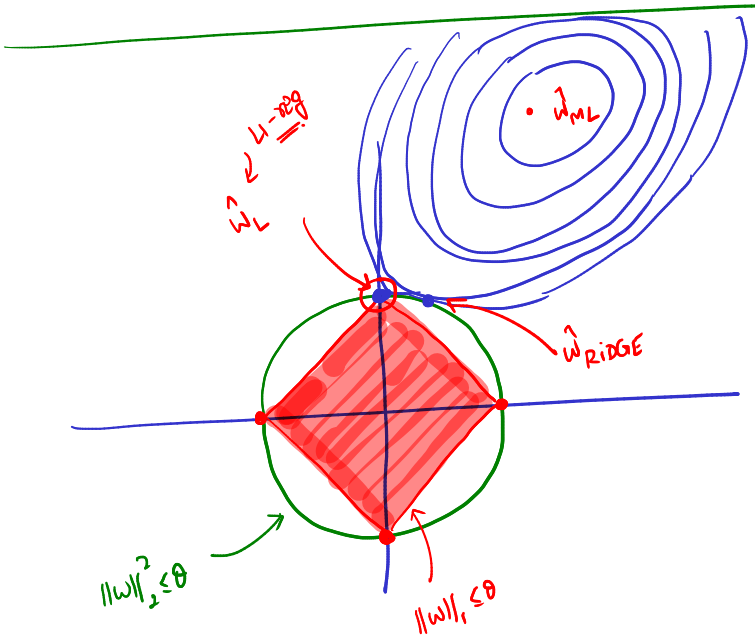
L1 Regularization

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|_1$$

$$=$$

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

$$\text{s.t} \quad \|w\|_1 \leq \theta$$

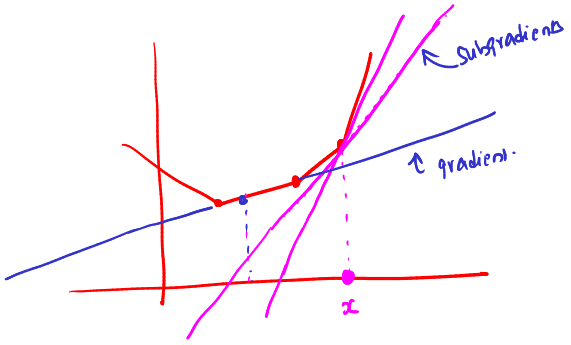


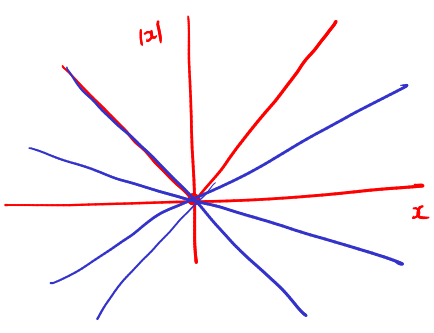
L1 Reg: LASSO - LEAST ABSOLUTE SHRINKAGE and SELECTION OPERATOR.

Points

- LASSO does not have a closed form solution
- Sub-gradient methods are usualy used to solve LASSO.

Sub-grad at 0 = $[-1, 1]$

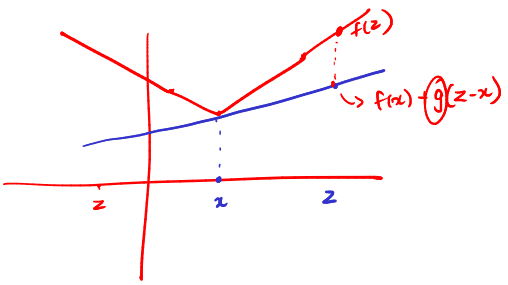




Subgradient

A vector $g \in \mathbb{R}^d$ is a sub-gradient of $f: \mathbb{R}^d \to \mathbb{R}$ at a point $x \in \mathbb{R}^d$ if

$$\forall z \quad f(z) \geq f(x) + g^T(z-x)$$



Why Subgradients?

- If function f to minimize is a Convex function, then sub-gradient descent converges!
- There are other special purpose methods for LASSO → (eg) IRLS [Iterative Reweighted Least Squares].

SUPERVISED LEARNING

BINARY CLASSIFICATION

$\{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$ (i)

$y_1, \ldots, y_n$ $y_i \in \{0,1\}$ / $\{-1,+1\}$

Goal: $h : \mathbb{R}^d \to \{0,1\}$

LOSS / ERROR

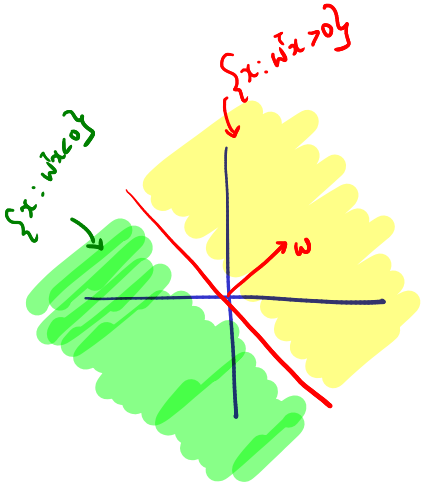
$$\text{Loss}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(h(x_i) \neq y_i\right)$$

$$\mathbb{1}(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{o/w} \end{cases}$$

0-1-loss

$$\min_{h \in \mathcal{H}_{\text{linear}}} \sum_{i=1}^{n} \mathbb{1}\left(h(x_i) \neq y_i\right)$$

NP-hard problem in general

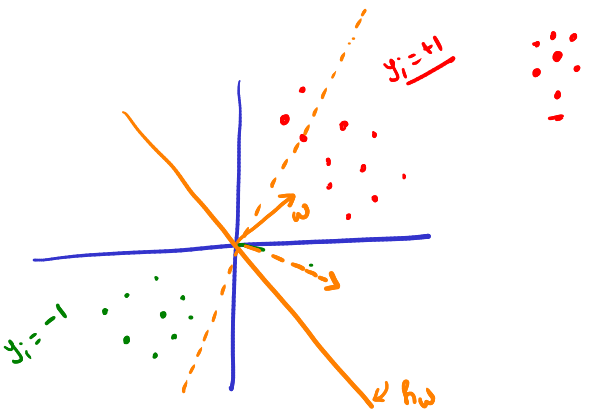


$$\mathcal{H}_{\text{linear}} = \left\{ h_w : h_w(x) = \text{Sign}(w^T x) \right\}$$

$$\text{Sign}(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{o/w} \end{cases}$$

- Can we use linear regression to solve classification problem?

$\{(x_1, y_1) \ldots (x_n, y_n)\}$ → [Lin Reg] → $w \in \mathbb{R}^d$ → $h_w : \mathbb{R}^d \to \{0,1\}$



Conclusion

Regression is SENSITIVE to location of the data points and not just the "side" on which the data lies w.r.t separator.

SIMPLE ALGORITHMS FOR CLASSIFICATION

- Given a test point $x_{test} \in \mathbb{R}^d$,
  find the closest point x^* to x_{test} in the training set

- Predict $y_{test} = y^*$

ISSUE: can get affected by outliers

Fix: Ask more neighbours

(K-NN) (K- Nearest Neighbours)

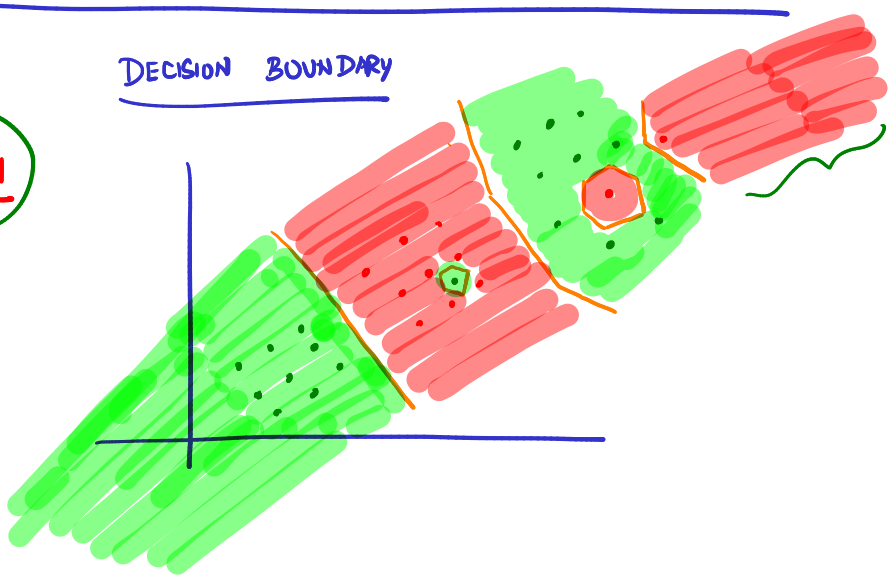
- Given x_{test}, find the K - closest points in the training set - $(x_1^*, x_2^*, \ldots, x_k^*)$

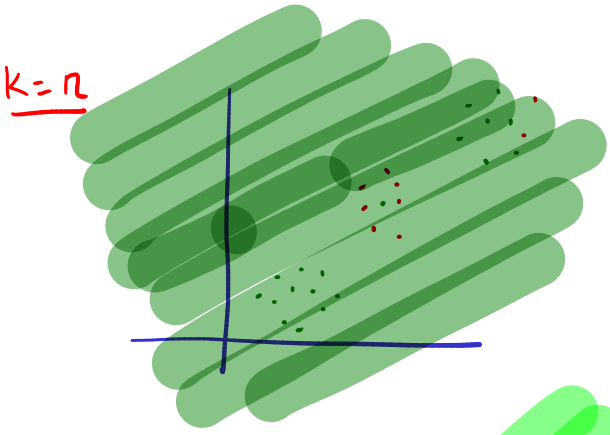
- PREDICT y_{test} = majority $(y_1^*, y_2^*, \ldots, y_k^*)$

DECISION BOUNDARY

K=1

SENSITIVE TO OUTLIERS!

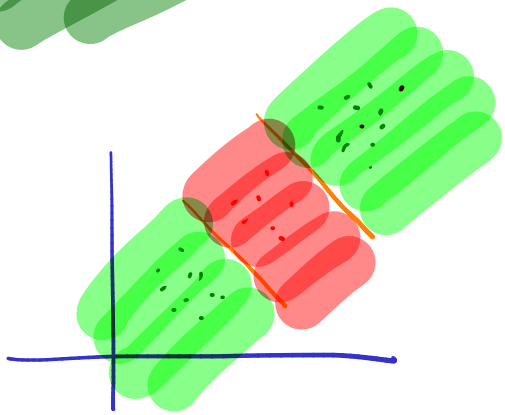




$k = n$

Every point is predicted $y = -1$ (green).

Too Smooth!



k^*.

A good choice.

Choosing k

→ Can treat as a HYPER PARAMETER

→ Smaller the k, complicated the decision boundary

→ Soln: CROSS-VALIDATE for k

ISSUES with k-NN

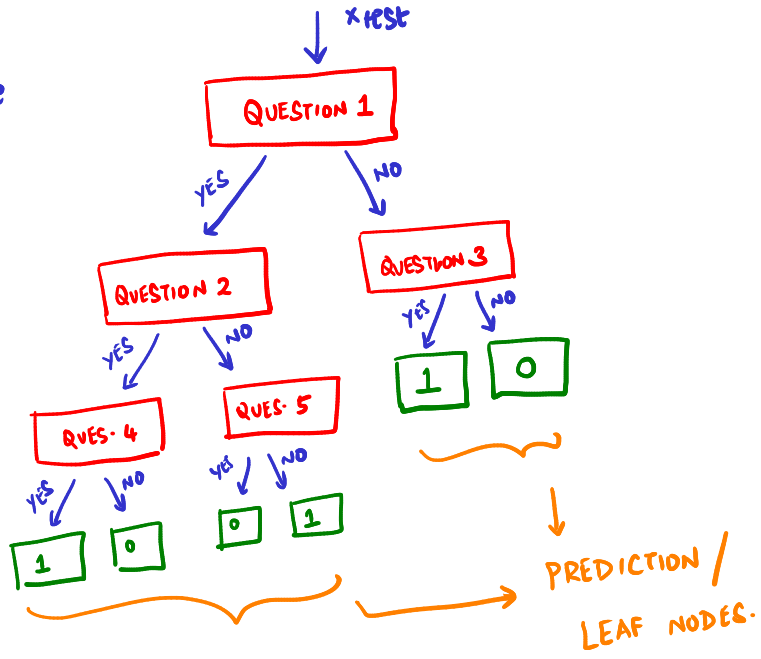
- Choosing a distance function
- PREDICTION is COMPUTATIONALLY EXPENSIVE.
- No MODEL is LEARNT.
  - Cannot throw away data after "LEARNING"

DECISION TREES

INPUT: Dataset $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ $\forall i$ $x_i \in \mathbb{R}^d$, $y_i \in \{+1, 0\}$

OUTPUT: DECISION TREE

DECISION tree



PREDICTION: Given x_{test}, traverse through the tree to reach a leaf node.

y_{test} = value in leaf node.

QUESTION: A question is a (feature, value) pair.

Eg: height $\leq$ 180cm ?

(f_3) θ

How to measure "goodness" of a Question?

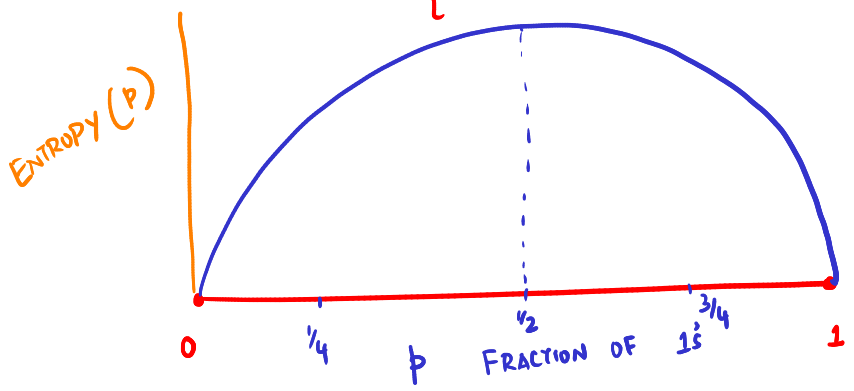
DATASET $D = \{(x_1, y_1) \ldots (x_n, y_n)\}$

$f_k \leq \theta$?

YES / NO

$D_{yes} = \{(x_1, y_1), (x_{10}, y_{10}), \ldots\}$ $\{(x_2, y_2), (x_3, y_3) \ldots\}$ D_{no}

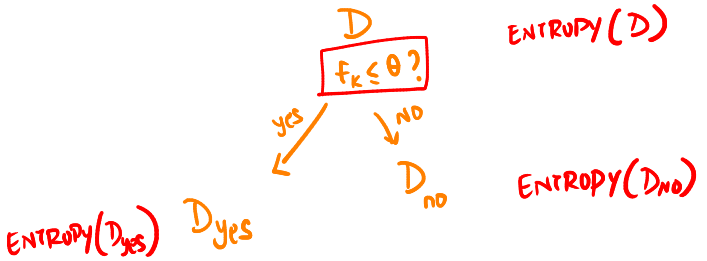
- Need is a measure of "Impurity" for a set of labels $\{y_1, \cdots, y_n\}$



$$\text{ENTROPY}(\{y_1, \cdots, y_n\}) = \text{ENTROPY}(p)$$

$$= -\left(p \log p + (1-p) \log (1-p)\right)$$

[convention $\log(0) = 0$]



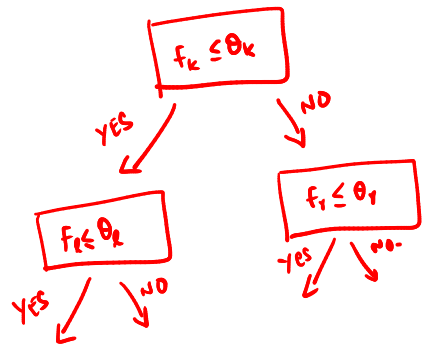
$$\text{INFORMATION GAIN}(\text{feature}, \text{value}) =$$

$$\text{ENTROPY}(D) - \left[\gamma \ \text{ENTROPY}(D_{yes}) + (1-\gamma)\ \text{ENTROPY}(D_{no})\right]$$

$$\gamma = \frac{|D_{yes}|}{|D|}$$

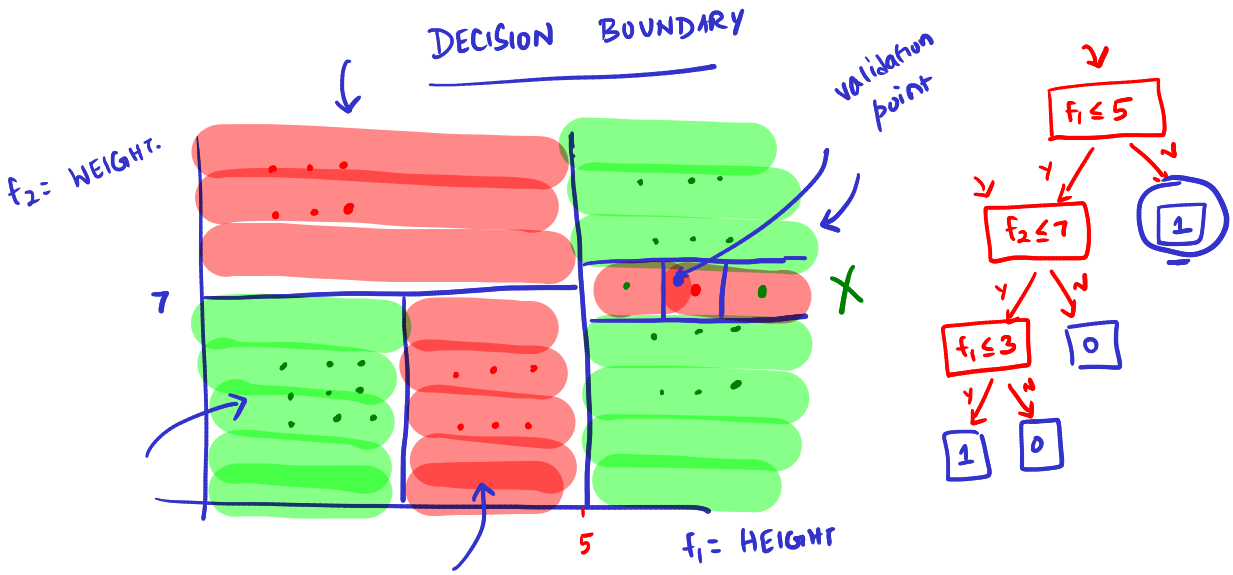
ALGORITHM - DECISION TREE

- DISCRETIZE each feature in [min, max] range
- Pick the Question that has the largest Information gain.
- Repeat the procedure for Dyes, Dno



POINTS

- Can stop growing a tree if a node becomes "SUFFICIENTLY" pure.
- DEPTH of the tree is a hyperparameter
- There are alternate measures for "goodness" of a Question
  → GINI INDEX


DECISION BOUNDARY
validation point
f_2 = WEIGHT.
7
5
f_1 = HEIGHT
$f_1 \leq 5$
Y
N
$f_2 \leq 7$
1
$f_1 \leq 3$
0
1
0

TYPES OF MODELING

DISTRIBUTION D OVER $X \times Y$ ← $\{+1, -1\}$ (FEATURE SPACE)

$\{x_1, \ldots, x_n\}$
$y_1, \ldots, y_n$

UNKNOWN BUT FIXED

$(x_i, y_i) \sim D$ ← LINK BETWEEN TRAIN and TEST.

CLASSIFICATION

- GENERATIVE MODEL
- DISCRIMINATIVE MODEL

GENERATIVE MODEL

MODEL

- $P(x, y)$

↳ NEXT.

DISCRIMINATIVE MODEL

MODEL

- $P(y/x)$

Eg: K-NN
DECISION-TREES

↳ $P(y=1/x) = 1$ if decision tree for x says 1.

$P(y=1/x) = 1$ if majority of neighbours say 1
$= 0$ otherwise

TYPES OF MODELING

DISTRIBUTION D OVER $X \times Y$ ← $\{+1, -1\}$ (FEATURE SPACE)

$\{x_1, \ldots, x_n\}$
$y_1, \ldots, y_n$

UNKNOWN BUT FIXED

$(x_i, y_i) \sim D$ ← LINK BETWEEN TRAIN and TEST.

CLASSIFICATION

- GENERATIVE MODEL
- DISCRIMINATIVE MODEL

GENERATIVE MODEL

MODEL

- $P(x, y)$

↳ NEXT.

DISCRIMINATIVE MODEL

MODEL

- $P(y/x)$

Eg: K-NN
DECISION-TREES

↳ $P(y=1/x) = 1$ if decision tree for x says 1.

$P(y=1/x) = 1$ if majority of neighbours say 1
$= 0$ otherwise

DATA = $\{(x_1, y_1) \ldots (x_n, y_n)\}$

$x_i \in \{0,1\}^d$

$y_i \in \{0,1\}$

EXAMPLE: SPAM CLASSIFICATION

$x_i \in \{0,1\}^d$ ← # words in dictionary.

Eg: "Hello, how are you?"

$P(x, y)$ $\quad$ ("Hello, how are you?", spam) $= 0.01$

$P(x, y) = P(x) \cdot P(y/x) = P(y) \cdot P(x/y)$

$P(x)$: $P(\text{email})$ $\quad$ $P(y/x)$: $P(\text{spam}/\text{email})$

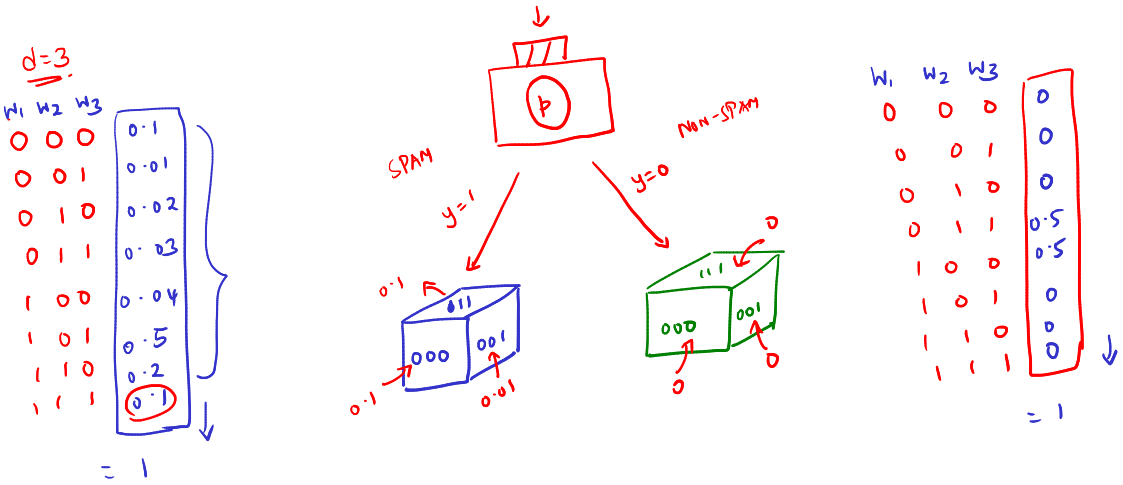
$P(y)$: $P(\text{spam})$ $\quad$ $P(x/y)$: $P(\text{email}/\text{spam})$

GENERATIVE STORY

STEP-1: DECIDE the Label by tossing a coin

$P(y_i = 1) = p$

STEP 2: DECIDE features using the label in step 1 by $P(x_i / y_i)$



parameters in the model $= 1 + (2^d - 1) + (2^d - 1)$

1: to decide label

$(2^d - 1)$: $P(x|y=1)$

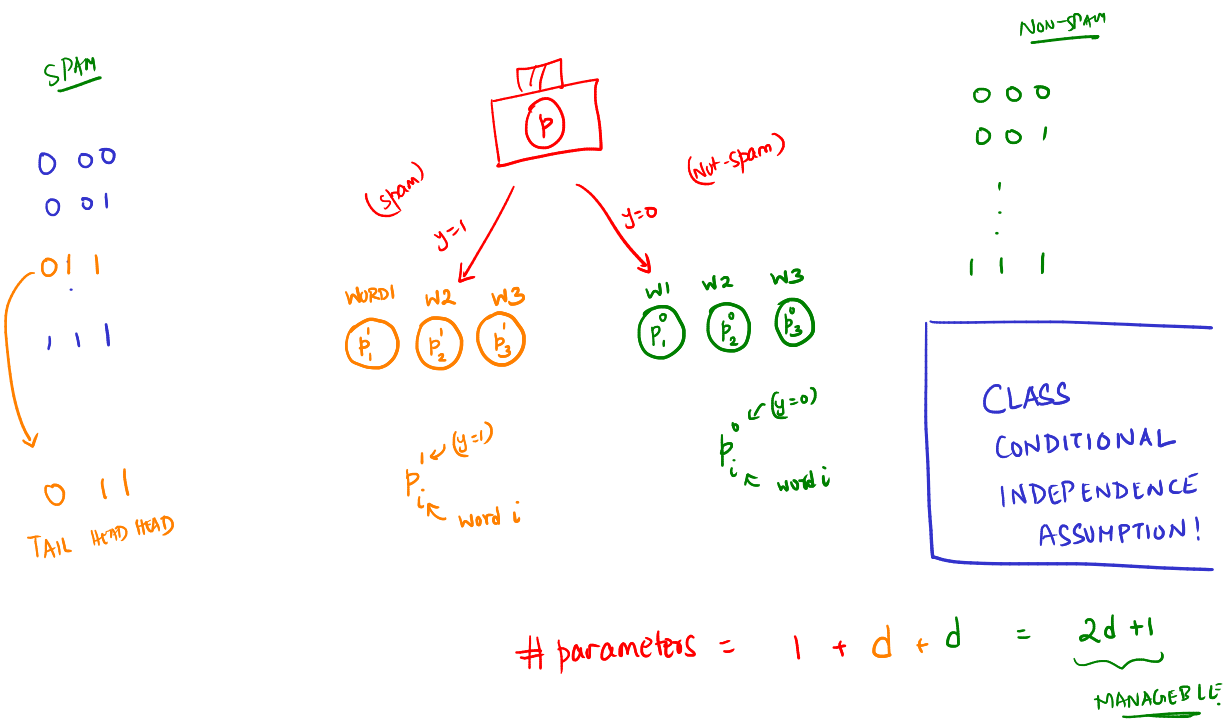
$(2^d - 1)$: $P(x|y=0)$

$= 1 + 2(2^d - 1)$

$= 2^{d+1} - 1$

ISSUE

- Too many parameters !
- Not a reasonable story.



parameters $= 1 + d + d = 2d + 1$

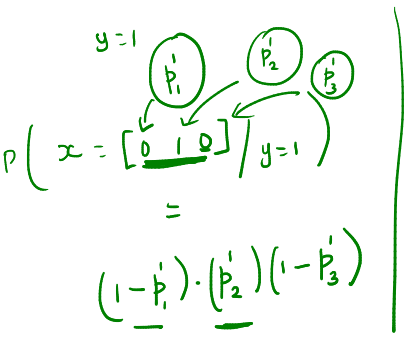
MANAGEBLE

STEP 1: $P(y=1) = p$

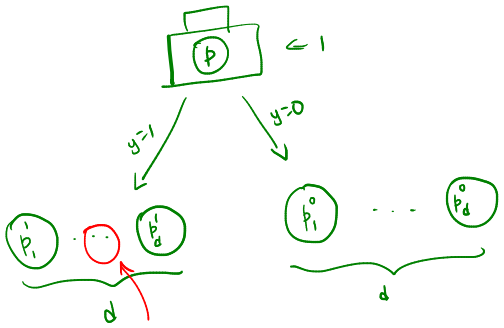
STEP 2: $P\left(x = [f_1 \; f_2 \; \cdots \; f_d] \,/\, y\right)$

$$= \prod_{i=1}^{d} \left(p_i^{y}\right)^{f_i} \left(1 - p_i^{y}\right)^{(1-f_i)}$$

FEATURES ARE CONDITIONALLY INDEPENDENT GIVEN LABEL



- How to estimate the parameters
- SOLUTION: MAXIMUM LIKELIHOOD!



CLASS CONDITIONAL INDEPENDENCE

$2d+1$

PARAMETER ESTIMATION

$$p, \{p_1^1, \dots p_d^1\}, \{p_1^0, \dots, p_d^0\}$$

MAX. LIKELIHOOD ESTIMATES

$1 \rightarrow \hat{p} = \frac{1}{n} \sum_{i=1}^{n} y_i \rightarrow$ {Fraction of Spam emails in the dataset}

$\forall j \in \{1, \dots, d\}$ $\quad \hat{p}_j^y = \dfrac{\sum_{i=1}^{n} \mathbb{1}(f_j^i = 1, y_i = y)}{\sum_{i=1}^{n} \mathbb{1}(y_i = y)}$

$\forall y \in \{0, 1\}$

$\sum_{i=1}^{n} \mathbb{1}(y_i = y) \leftrightarrow$ Number of emails with label y.

$\hookrightarrow$ Fraction of y-labelled emails that contain the j^{th} word.

PREDICTION

Given $x^{test} \in \{0,1\}^d$, what is $\hat{y}^{test}$?

$$P(y^{test} = 1 \mid x^{test}) > P(y^{test} = 0 \mid x^{test})$$

$\Rightarrow \hat{y}^{test} = 1$

$= 0$ otherwise.

How to obtain $P(y/x)$ from $P(y)$ and $P(x/y)$?

BAYES RULE !

$$P(y^{test}=1 \mid x^{test}) = \frac{P(x^{test} \mid y^{test}=1) \cdot P(y^{test}=1)}{P(x^{test})}$$

$$P(y^{test}=0 \mid x^{test}) = \frac{P(x^{test} \mid y^{test}=0) \cdot P(y^{test}=0)}{P(x^{test})}$$

$$P(x^{test} \mid y^{test}=1) \cdot P(y^{test}=1)$$

$$= P(x^{test} = [f_1\ f_2 \cdots f_d] \mid y^{test}=1) \cdot P(y^{test}=1)$$

$$= \left(\prod_{j=1}^{d} (\hat{p}_j^1)^{f_j} (1-\hat{p}_j^1)^{(1-f_j)} \right) \cdot \hat{p}$$

$X_{test} = [0\ 0\ 1\ 0]$ ($f_1\ f_2\ f_3\ f_4$)

IF

$$\left(\prod_{j=1}^{d} (\hat{p}_j^1)^{f_j} (1-\hat{p}_j^1)^{(1-f_j)} \right) \cdot \hat{p} > \left(\prod_{j=1}^{d} (\hat{p}_j^0)^{f_j} (1-\hat{p}_j^0)^{(1-f_j)} \right) \cdot (1-\hat{p})$$

$\Rightarrow$ PREDICT $\hat{y}^{test} = 1$

ELSE $\hat{y}^{test} = 0$.

MODEL USES 2 main Things

NAIVE BAYES ALGORITHM.

- CLASS CONDITIONAL INDEPENDENCE
  - may not hold in practice
  - NAIVE assumption
  - Still works well in practice.
- BAYES THEOREM

PITFALLS IN NAIVE BAYES TO WATCH OUT FOR.

- IF a word does not appear in the train set but appears in a test datapoint,

$\hat{p}_j^1 = 0 \qquad \hat{p}_j^0 = 0$

$$P\left(y^{test}=1 \,\middle|\, x^{test}=[f_1 f_2 \cdots f_d]\right) \propto \left(\prod_{i=1}^{d} (\hat{p}_i^1)^{f_i} (1-\hat{p}_i^1)^{(1-f_i)}\right) \hat{p}$$

(f_i: zebra $=1$; $\hat{p}_i^1$: 0; $(1-\hat{p}_i^1)$: 1; product: 0)

$$P\left(y^{test}=0 \,\middle|\, x^{test}=[f_1 \cdots f_d]\right) \propto \left(\prod_{i=1}^{d} (\hat{p}_i^0)^{f_i} (1-\hat{p}_i^0)^{(1-f_i)}\right) (1-\hat{p})$$

(zebra $=1$; $\hat{p}_i^0$: 0; $(1-\hat{p}_i^0)$: 1; product: 0)

Possible Fix

- Can add two "pseudo" emails with all words present - one email has label 0 and another has label 1

LAPLACE SMOOTHING

$x_1^{pseudo} = [1\ 1\ 1\ 1\ 1 \cdots 1]$ $\quad y_1^{pseudo} = 1$

$x_0^{pseudo} = [1\ 1\ 1 \cdots 1]$ $\quad y_0^{pseudo} = 0$

DECISION FUNCTION OF NAIVE BAYES.

Given x_{test}; $y_{test} = 1$ if $\dfrac{P(y_{test}=1 \mid x_{test})}{P(y_{test}=0 \mid x_{test})} \geq 1$

$$\log\left(\frac{P(y_{test}=1 \mid x_{test})}{P(y_{test}=0 \mid x_{test})}\right) \geq 0$$

$$\log\left(\frac{P(x_{test} \mid y_{test}=1)\cdot P(y_{test}=1) \,/\, P(x_{test})}{P(x_{test} \mid y_{test}=0)\cdot P(y_{test}=0) \,/\, P(x_{test})}\right) \geq 0$$

$x_{test} = [f_1\ f_2 \cdots f_d]$

$$\log\left(\prod_{i=1}^{d} \frac{(\hat{p}_i^1)^{f_i}(1-\hat{p}_i^1)^{(1-f_i)}}{(\hat{p}_i^0)^{f_i}(1-\hat{p}_i^0)^{(1-f_i)}} \cdot \frac{\hat{p}}{(1-\hat{p})}\right) \geq 0$$

$$= \log\left(\prod_{i=1}^{d} \left(\frac{\hat{p}_i^1}{\hat{p}_i^0}\right)^{f_i} \left(\frac{1-\hat{p}_i^1}{1-\hat{p}_i^0}\right)^{(1-f_i)} \cdot \frac{\hat{p}}{1-\hat{p}}\right) \geq 0$$

$$= \sum_{i=1}^{d}\left(f_i \log\left(\frac{\hat{p}_i^1}{\hat{p}_i^0}\right) + (1-f_i)\log\left(\frac{1-\hat{p}_i^1}{1-\hat{p}_i^0}\right) + \log\left(\frac{\hat{p}}{1-\hat{p}}\right)\right) \geq 0$$

$$= \sum_{i=1}^{d} f_i\left(\log\left(\frac{\hat{p}_i^1(1-\hat{p}_i^0)}{\hat{p}_i^0(1-\hat{p}_i^1)}\right)\right) + \log\left(\frac{(1-\hat{p}_i^1)}{(1-\hat{p}_i^0)}\right) + \log\left(\frac{\hat{p}}{1-\hat{p}}\right) \geq 0$$

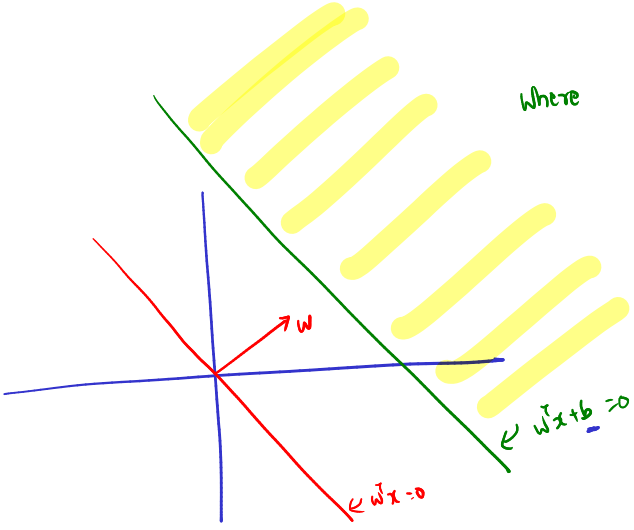
$x_{test} = [f_1 \cdots f_d]$

DECISION FUNCTION is of the form

Predict $y_{test} = 1$ iF $w^T x_{test} + b \geq 0$

$w \in \mathbb{R}^d$

where $w_i = \log\left(\frac{\hat{p}_i^1(1-\hat{p}_i^0)}{\hat{p}_i^0(1-\hat{p}_i^1)}\right)$, $b =$



CONCLUSION:

→ DECISION FUNCTION OF NAIVE BAYES IS LINEAR!

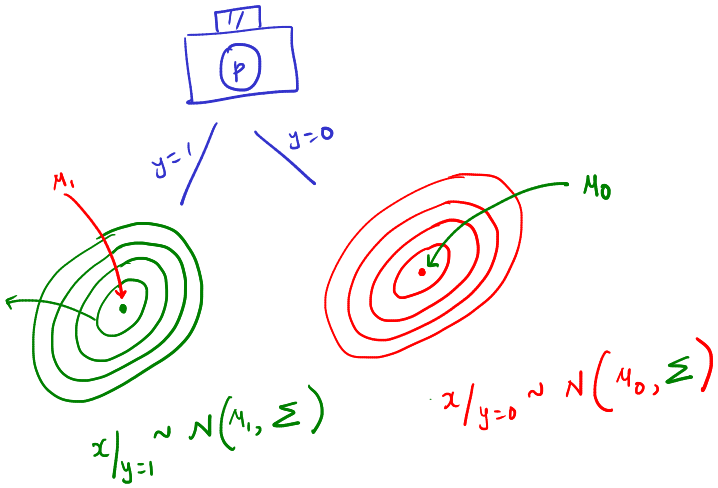
DATA: $\{(x_1, y_1), \ldots (x_n, y_n)\}$

$x_i \in \mathbb{R}^d \qquad y_i \in \{0, 1\}$

A GENERATIVE STORY

PARAMETERS

- p
- μ_0, μ_1
- Σ



NOTE: In this model, covariances are assumed to be same

MAXIMUM LIKELIHOOD ESTIMATES

$$\hat{p} = \frac{\sum_{i=1}^{n} y_i}{n}$$ ← FRACTION of points labelled 1.

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{n} \mathbb{1}(y_i = 1) \cdot x_i}{\sum_{i=1}^{n} \mathbb{1}(y_i = 1)}$$ ← Sample mean of data points labelled 1.

$$\hat{\mu}_0 = \frac{\sum_{i=1}^{n} \mathbb{1}(y_i = 0) \cdot x_i}{\sum_{i=1}^{n} \mathbb{1}(y_i = 0)}$$ ← Sample mean of data points labelled 1.

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu}_{y_i})(x_i - \hat{\mu}_{y_i})^T$$

PREDICTION ? Bayes rule.

$$P(y_{test} \mid x_{test}) \propto \underbrace{P(x_{test} \mid y_{test})}_{f(x_{test}; \hat{\mu}_{y_{test}}, \hat{\Sigma})} \cdot \underbrace{P(y_{test})}_{\hat{p}}$$

Predict $y_{test} = 1$ if

$$f(x_{test}; \hat{\mu}_1, \hat{\Sigma}) \cdot \hat{p} \geq f(x_{test}; \hat{\mu}_0, \hat{\Sigma}) \cdot (1 - \hat{p})$$

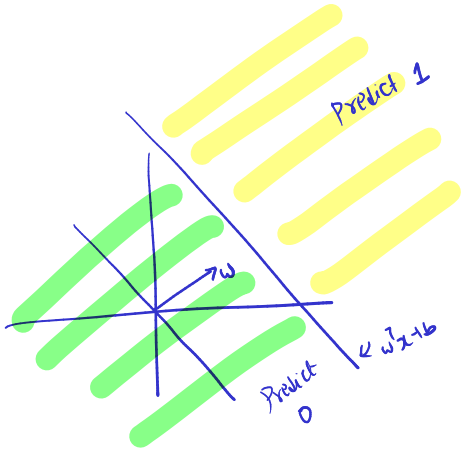
$$e^{-(x_{test} - \hat{\mu}_1)^T \hat{\Sigma}^{-1} (x_{test} - \hat{\mu}_1)} \cdot \hat{p} \geq e^{-(x_{test} - \hat{\mu}_0)^T \hat{\Sigma}^{-1} (x_{test} - \hat{\mu}_0)} (1 - \hat{p})$$

on simplification [take log]

Predict 1 if

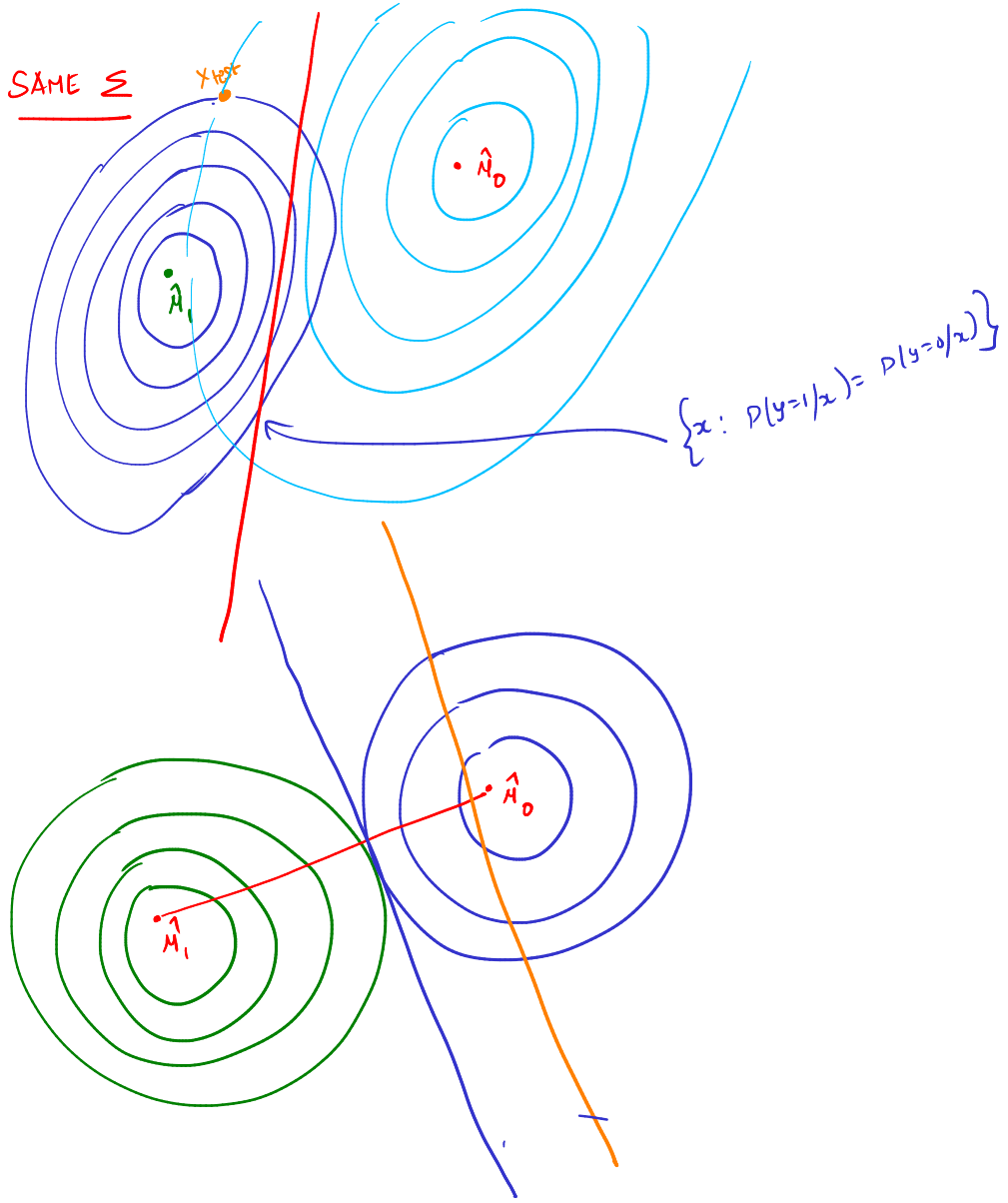
$$\underbrace{(\hat{\mu}_1 - \hat{\mu}_0)^T \hat{\Sigma}^{-1}}_{w^T} x_{test} + \underbrace{\hat{\mu}_0^T \hat{\Sigma}^{-1} \hat{\mu}_0 - \hat{\mu}_1^T \hat{\Sigma}^{-1} \hat{\mu}_1}_{b} + \log\left(\frac{1-\hat{p}}{\hat{p}}\right) \geq 0$$

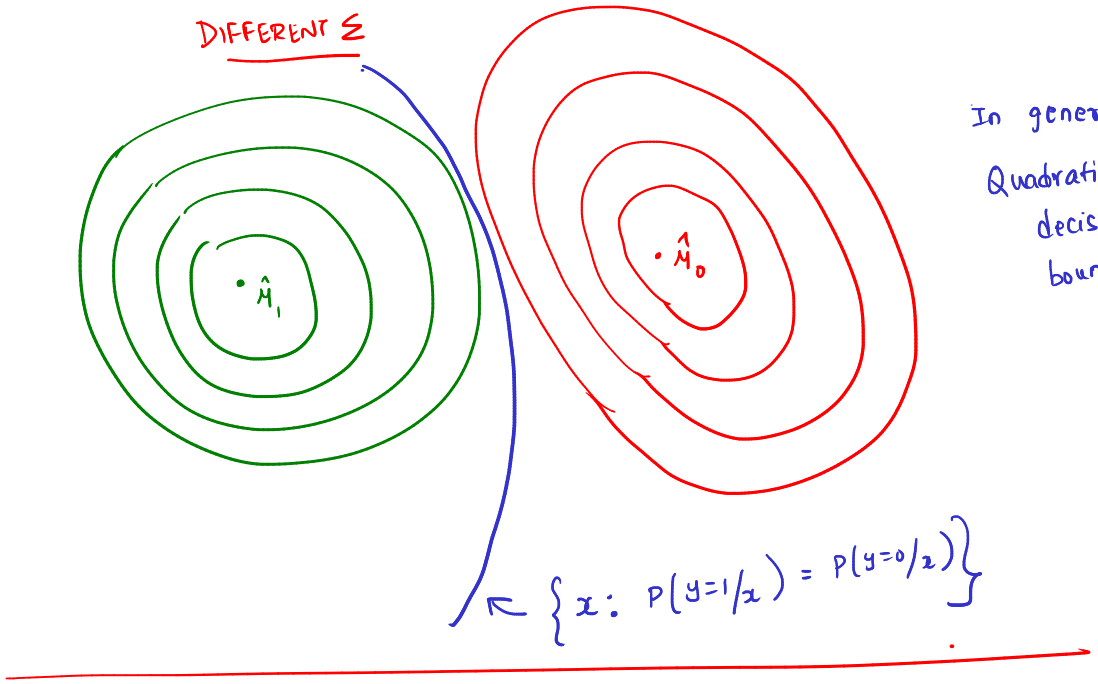
$$w^T x_{test} + b \geq 0$$



DECISION FUNCTION is LINEAR!

↳ Σ is same for both classes.





In general, Quadratic decision boundary.

GAUSSIAN NAIVE BAYES

Supervised

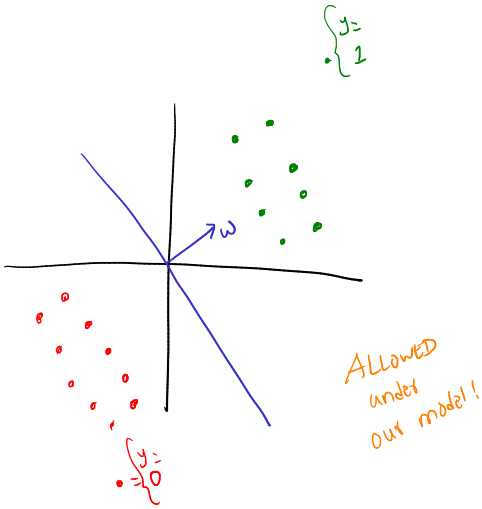
↓

Binary classification

Generative	Discriminative
• Naive Bayes	• K-NN
• Gaussian discriminant Analysis	• Decision trees

Discriminative models for classification

How to model $P(y=1/x)$?

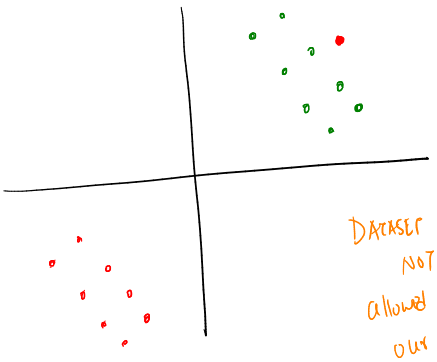


ALLOWED under our model!

SIMPLEST ASSUMPTION

$$P(y=1/x) = \begin{cases} 1 & \text{if } w^T x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

LINEAR SEPARABILITY ASSUMPTION



DATASET is NOT allowed under our model.

Goal.

$$\min_{h \in \mathcal{H}} \sum_{i=1}^{n} \mathbb{1}(h(x_i) \neq y_i)$$

NP-HARD for a general dataset even if $\mathcal{H}$ is just linear hypotheses.

- How about with extra "Linear separability" assumption.

LINEAR SEPARABILITY ASSUMPTION

$\exists w \in \mathbb{R}^d$ s.t $\text{sign}(w^T x_i) = y_i \quad \forall i \in [n]$

PERCEPTRON [Rosenblatt, 1950s]

Input: $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ $\quad x_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$

ITERATION

$w^0 = 0 \in \mathbb{R}^d$ $\quad [0 \; 0 \; \cdots \; 0]$

Until convergence

- Pick (x_i, y_i) pair from the dataset
- IF $\text{sign}(w^{t^T} x_i) = y_i$

  do nothing

  else $\quad w^{t+1} = w^t + x_i y_i$ ← UPDATE RULE.

  ($x_i \in \mathbb{R}^d$, $y_i \in \{\pm 1\}$)

  end.

end.

UPDATE RULE

if mistake $\quad w^{t+1} = w^t + x_i y_i$

MISTAKE TYPE 1

Predicted = 1 ← $(w^{t^T} x_i) \geq 0$

Actual = -1 $\quad y_i = -1$

$$w^{t+1} = w^t + x_i y_i$$

$$(w^{t+1})^T x_i = (w^t + x_i y_i)^T x_i$$

$$= \underbrace{w^{t^T} x_i}_{\geq 0} + \underbrace{y_i \|x_i\|^2}_{-1 \; (>0)}$$

Negative.

MISTAKE TYPE 2

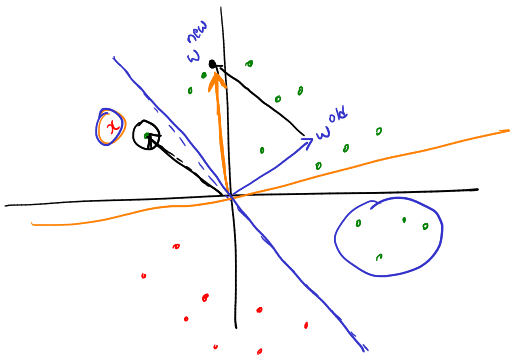
Pred = -1 ← $(w^{t^T} x_i) < 0$

Act = +1 ← $y_i = +1$

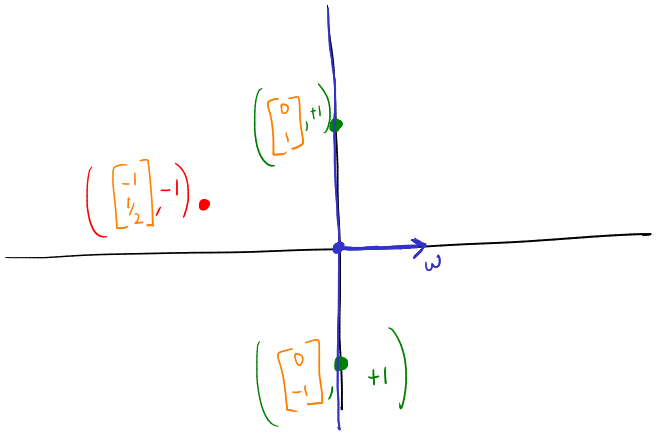
$$(w^{t+1})^T x_i = \underbrace{w^{t^T} x_i}_{<0} + \underbrace{y_i \|x_i\|^2}_{+1 \; (>0)}$$

⇒ update rule pushes w in the "right" direction for x_i

$w^{new} = w^{old} + x \cdot (\underbrace{1}_{y})$



- Fixing w for one x might affect decision for other data points.
- So need more careful argument for convergence.



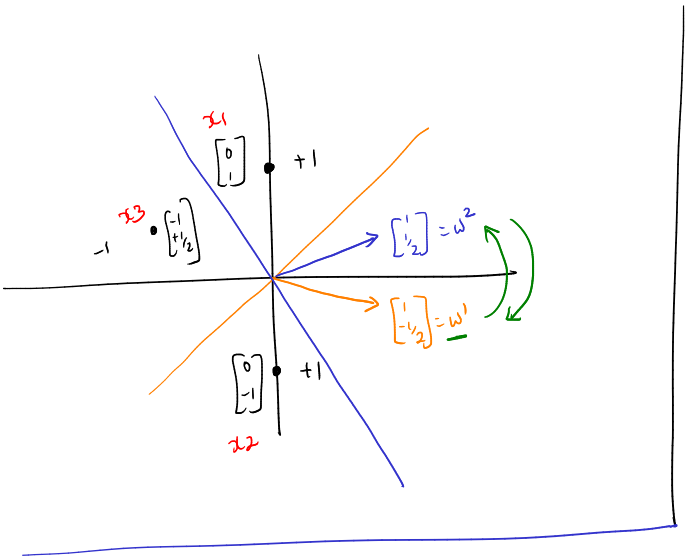
Is this a Linearly separable dataset?

$$\left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, +1 \quad \begin{bmatrix} 0 \\ -1 \end{bmatrix}, +1 \quad \begin{bmatrix} -1 \\ 1/2 \end{bmatrix}, -1 \right\}$$

Is there a $w \in \mathbb{R}^2$ s.t $w^T x_i \geq 0 \Rightarrow y_i = +1$

$w^T x_i < 0 \Rightarrow y_i = -1$

PERCEPTRON



$w^0 = [0 \ 0]$

$w^{0T} x_1 = 0$; $w^{0T} x_2 = 0$; $w^{0T} x_3 = 0$

$\hat{y}_1 = +1$ $\hat{y}_2 = +1$ $\hat{y}_3 = +1$

$w^1 = w^0 + x_3 y_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ 1/2 \end{bmatrix} \times -1$

$= \begin{bmatrix} 1 \\ -1/2 \end{bmatrix}$

$w^2 = w^1 + x_1 y_1 = \begin{bmatrix} 1 \\ -1/2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot 1$

$w^2 = \begin{bmatrix} 1 \\ 1/2 \end{bmatrix}$

$w^3 = w^2 + x_2 y_2 = \begin{bmatrix} 1 \\ 1/2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \times 1$

$w^3 = \begin{bmatrix} 1 \\ -1/2 \end{bmatrix} = w^1$ ←

ASSUMPTION

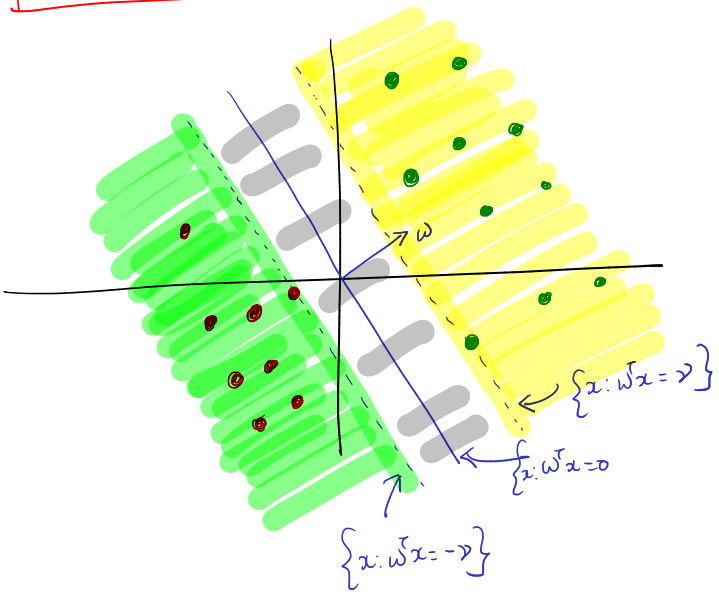
(1) LINEAR SEPARABILITY with γ-MARGIN

A Dataset $\{(x_1, y_1), \ldots (x_n, y_n)\}$

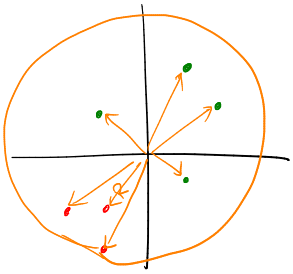
is Linearly Separable with γ-margin if $\exists w^* \in \mathbb{R}^d$ s.t

$(w^{*T} x_i) y_i \geq \gamma \quad \forall i$

for some $\gamma > 0$

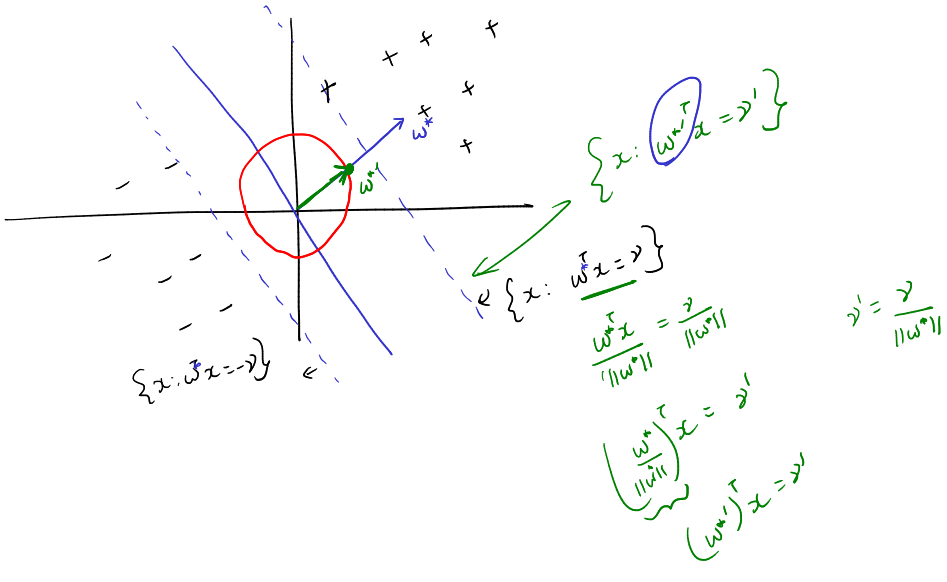


(2) RADIUS ASSUMPTION



$\forall i \in D \quad \|x_i\|_2 \leq R$ for some $R > 0$.

(3) Without loss of generality, assume $\|w^*\| = 1$



ANALYSIS OF "MISTAKES" OF PERCEPTRON

- Observe that an update happens only when a mistake occurs.
- Say w^t is the current guess and a mistake happens w.r.t (x, y)

$$w^{t+1} = w^t + x \cdot y$$

$$\|w^{t+1}\|^2 = \|w^t + x \cdot y\|^2$$

$$= (w^t + x \cdot y)^T (w^t + x \cdot y)$$

$$= \|w^t\|^2 + \underbrace{2 \cdot (w^{t^T} x) y}_{\leq 0 \text{ because mistake}} + \underbrace{\|x\|^2 \cdot \overbrace{y^2}^{=+1}}_{\leq R^2}$$

$$\|w^{t+1}\|^2 \leq \|w^t\|^2 + R^2$$

$$\|w^{t+1}\|^2 \leq \left(\|w^{t-1}\|^2 + R^2\right) + R^2$$

$$\|w^{t+1}\|^2 \leq \|w^0\|^2 + t R^2$$

$\|w^0\|^2 = 0$

$$\Rightarrow \boxed{\|w^{t+1}\|^2 \leq t R^2} \quad \text{(mistakes = } t\text{)} \quad \text{---------- (1)}$$

$$(w^{t+1})^T w^* = (w^t + x \cdot y)^T w^*$$

$$= w^{t^T} w^* + \underbrace{(w^{*^T} x) y}_{\geq \gamma}$$

$$(w^{t+1})^T w^* \geq w^{t^T} w^* + \gamma$$

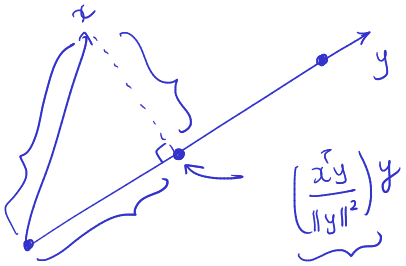
$$\geq \left(w^{k-1}{}^T w^* + \gamma\right) + \gamma$$

$$\left(w^{k+1}\right)^T w^* \geq w^{0T} w^* + k\gamma$$

$w^0 = 0$

$$\Rightarrow \boxed{\left(w^{k+1}\right)^T w^* \geq k\gamma} \quad \text{—(2)}$$

For any x, y



$$\left\| \left(\frac{x^T y}{\|y\|^2}\right) y \right\|^2 \leq \|x\|^2$$

[Pythagoras]

$$(x^T y)^2 \frac{\|y\|^2}{\|y\|^4} \leq \|x\|^2$$

$$\Rightarrow \boxed{(x^T y)^2 \leq \|x\|^2 \|y\|^2}$$

Cauchy-Schwartz

From (2)

$$k\gamma \leq \left(w^{k+1}\right)^T w^*$$

$$\Rightarrow k^2\gamma^2 \leq \left(\left(w^{k+1}\right)^T w^*\right)^2 \leq \|w^{k+1}\|^2 \underbrace{\|w^*\|^2}_{1}$$

(x, y)

[From C.S]

$$\Rightarrow \boxed{\|w^{k+1}\|^2 \geq k^2\gamma^2} \quad \text{—(3)}$$

$$k^2\gamma^2 \leq \|w^{k+1}\|^2 \leq kR^2$$

↑ From (3) ↑ From (1)

$$\Rightarrow k^2\gamma^2 \leq kR^2$$

$$\Rightarrow \boxed{k \leq R^2/\gamma^2}$$

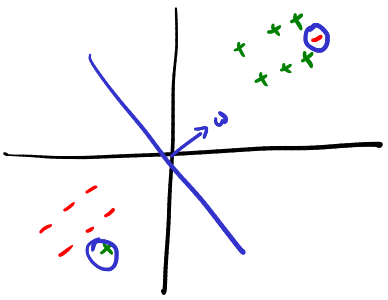
$k \leftarrow$ # mistakes

RADIUS MARGIN BOUND

$\Rightarrow$ # mistakes is bounded [because $\gamma > 0$].

$\Rightarrow$ PERCEPTRON CONVERGES !

Perceptron mistakes $\leq \frac{R^2}{\gamma^2}$ ← Radius, ← margin

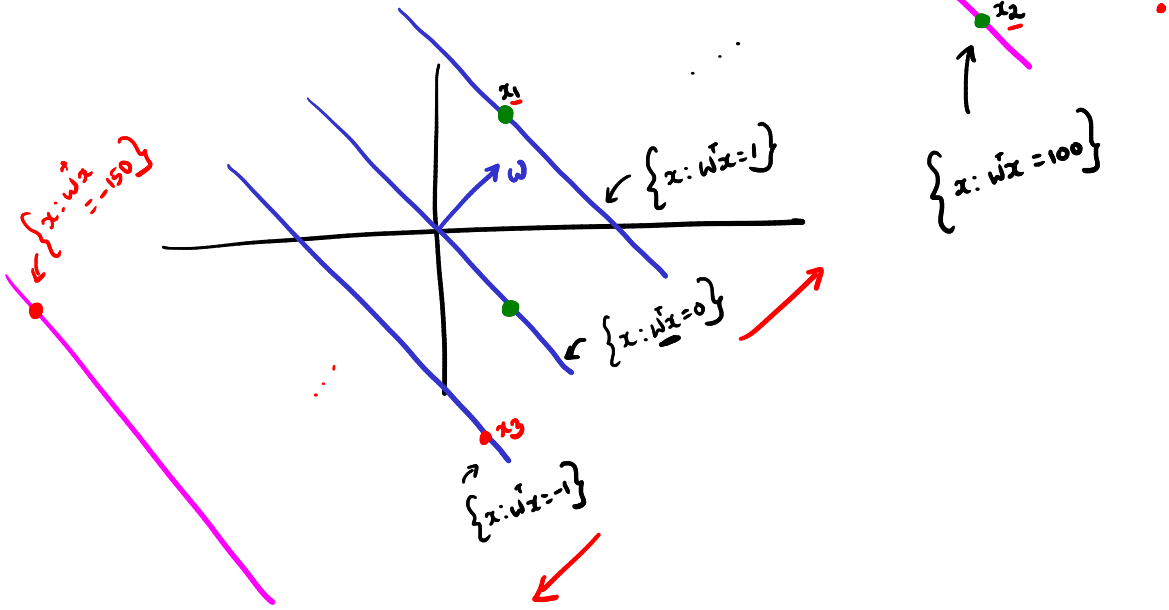


$$P(y=1|x) = \begin{cases} 1 & \text{if } w^T x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Can we model probabilities differently?

- Start with a simple LINEAR model

$z \in [-\infty, \infty]$ ← $z = w^T x$ ← feature, $x \in \mathbb{R}^d$



- Larger the score ($z = w^T x$), more the probability of being +1

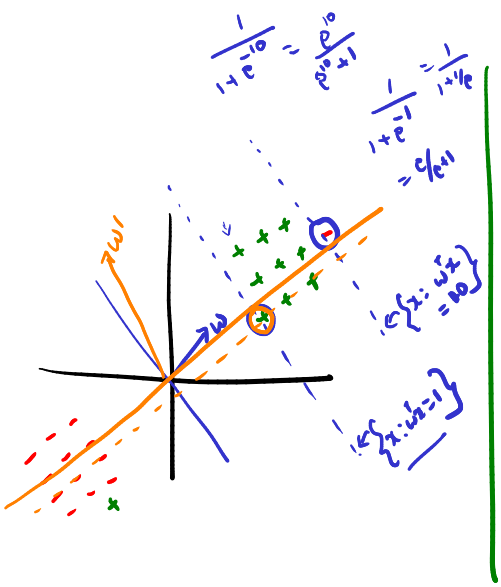
$$g(z) = 0.5 \quad \text{if } z = 0$$

$g(z) \to 1$ as $z \to \infty$

$g(z) \to 0$ as $z \to -\infty$

score

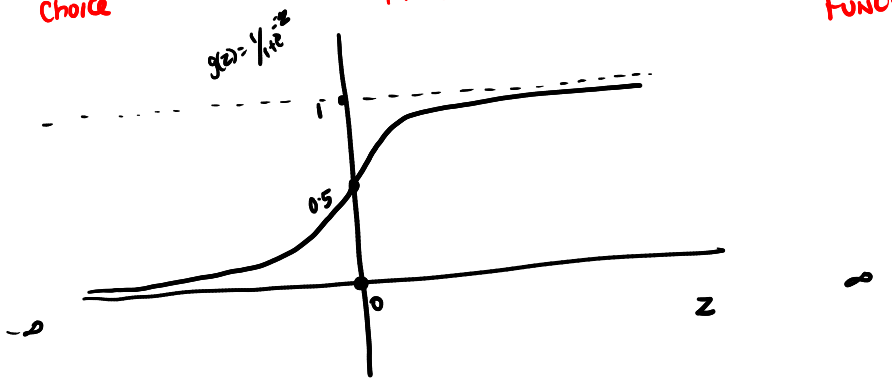
LINK FUNCTION



One popular Choice

$$g(z) = \frac{1}{1+e^{-z}}$$

SIGMOID / LOGISTIC FUNCTION



MODEL: LOGISTIC REGRESSION

$$P(y=1/x) = \frac{1}{1+e^{-w^T x}} = g(w^T x)$$

Dataset: $\{(x_1,y_1), \ldots, (x_n,y_n)\}$ $\quad y_i \in \{0,1\}$.

How to find w: Maximum Likelihood.

$$L(w; \text{Data}) = \prod_{i=1}^{n} \left(g(w^T x_i)\right)^{y_i} \left(1 - g(w^T x_i)\right)^{(1-y_i)}$$

$$\log L(w; \text{Data}) = \sum_{i=1}^{n} y_i \log\left(g(w^T x_i)\right) + (1-y_i)\log\left(1 - g(w^T x_i)\right)$$

$$= \sum_{i=1}^{n} y_i \log\left(\frac{1}{1+e^{-w^T x_i}}\right) + (1-y_i)\log\left(\frac{e^{-w^T x_i}}{1+e^{-w^T x_i}}\right)$$

$$= \sum_{i=1}^{n} \left[(1-y_i)(-w^T x_i) - \log\left(1+e^{-w^T x_i}\right)\right]$$

Goal:

$$\max_w \underbrace{\sum_{i=1}^{n} \left[(1-y_i)(w^T x_i) - \log\left(1+e^{-w^T x_i}\right)\right]}_{\log L(w)}$$

- No closed form expression
- Can perform Gradient descent. [ascent]

$$\nabla \log L(w) = \sum_{i=1}^{n} \left[(1-y_i)(-x_i) - \left(\frac{e^{-w^T x_i}}{1+e^{-w^T x_i}} \right)(-x_i) \right]$$

$$= \sum_{i=1}^{n} \left[-x_i + y_i x_i + x_i \left(\frac{e^{-w^T x_i}}{1+e^{-w^T x_i}} \right) \right]$$

$$= \sum_{i=1}^{n} y_i x_i - x_i \left(\frac{1}{1+e^{-w^T x_i}} \right)$$

$$= \sum_{i=1}^{n} x_i \left(y_i - \frac{1}{1+e^{-w^T x_i}} \right)$$

Gradient update rule (step-size)

$$w_{t+1} = w_t + \eta_t \nabla \log L(w_t)$$

$$= w_t + \eta_t \left(\sum_{i=1}^{n} x_i \underbrace{\left(y_i - \frac{1}{1+e^{-w_t^T x_i}} \right)}_{\theta_i} \right)$$

$x_i \in \mathbb{R}^d$; $y_i \in \{0,1\}$; $\frac{1}{1+e^{-w_t^T x_i}} = g(w_t^T x_i)$

$x_{test} \in \mathbb{R}^d$

$$\hat{y}_{test} = \text{Sign}\left(\hat{w}^T x_{test} \right)$$

KERNEL VERSION

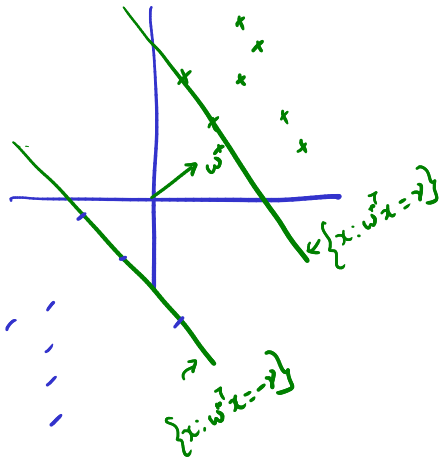
- Can argue $w^* = \sum_{i=1}^{n} \alpha_i x_i$ [Formal theorem is called the Representer theorem]

REGULARIZED VERSION

$$\min_w \sum_{i=1}^{n} \left[\log\left(1+e^{-w^T x_i}\right) + w^T x_i (1-y_i) \right] + \frac{\lambda}{2} \|w\|^2$$

(cross validate hyper parameter λ)

$\frac{\lambda}{2}\|w\|^2$: Regularizer



PERCEPTRON

$$\#\text{ mistakes} \leq \frac{R^2}{\gamma^2}$$

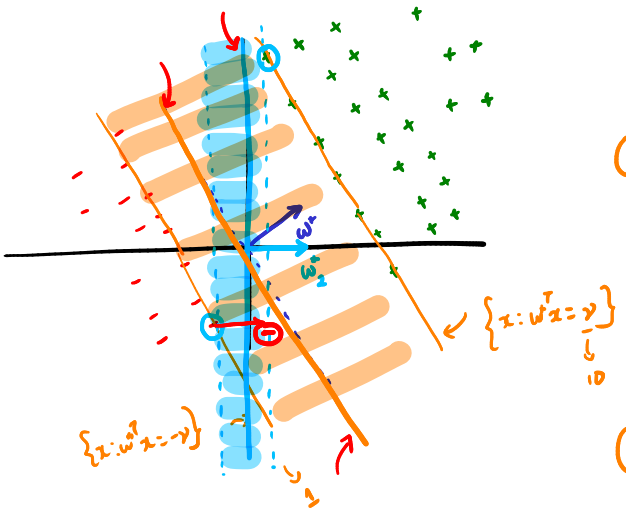
$\|x_i\|^2 \leq R^2$

Dataset - L.S with margin γ

$(w^{*T}x_i)y_i \geq \gamma \quad \forall i$

$\gamma > 0$

"QUALITY" OF FINAL SOLUTION



Question

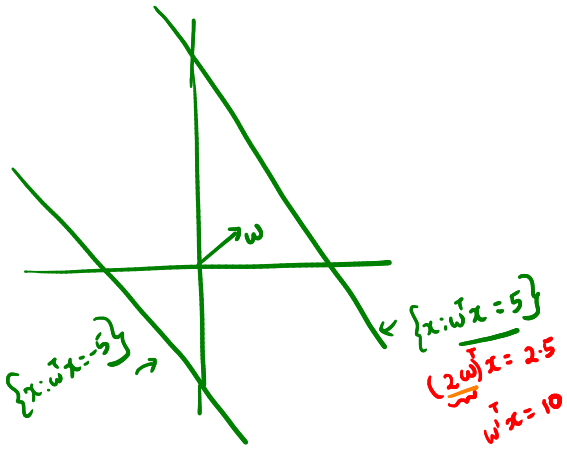
Given that we prefer classifiers with large margin, can we directly find them?

Observation

① # mistakes depends on the best possible w^*'s margin.

② w_{Perc} need not necessarily be w^*. It could be w_2^* also (blue line)

Goal: To come up with a formulation that maximizes "margin"



$$\max_{w,\gamma} \gamma$$

such that

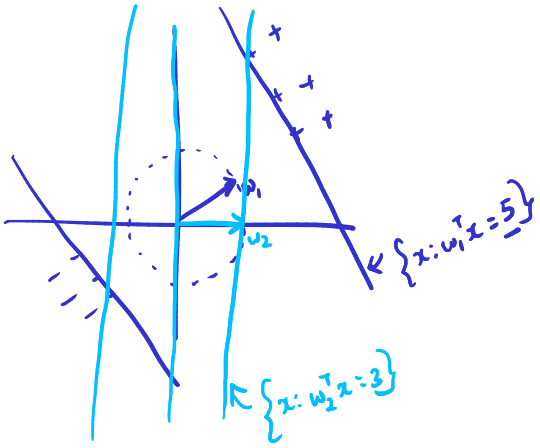
$$(w^Tx_i)y_i \geq \gamma \quad \forall i$$

Issue: Can scale w arbitrarily.

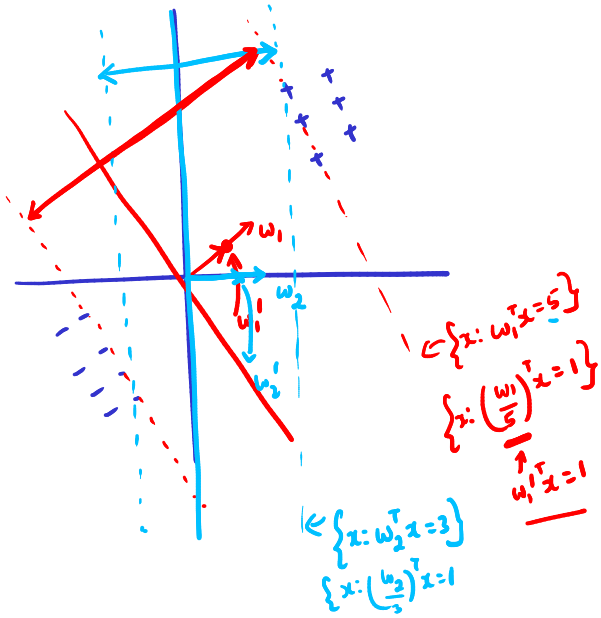
$$\max_{w,\gamma} \gamma$$

$$(w^Tx_i)y_i \geq \gamma \quad \forall i$$

$$\|w\|^2 = 1$$



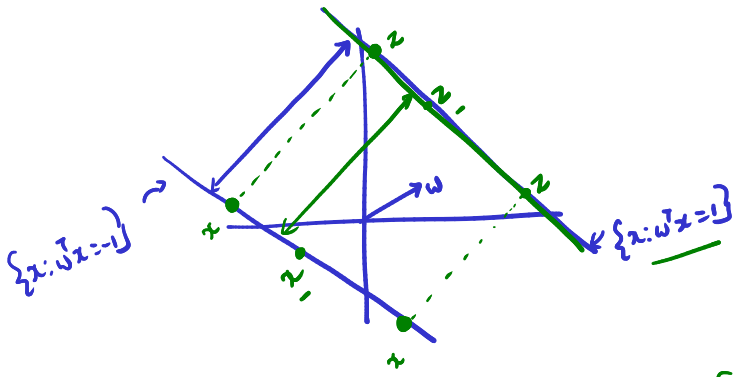
$\Rightarrow$



$$\max_{w} \quad \text{width}(w)$$

$$\text{s.t} \quad (w^T x_i) y_i \geq 1 \quad \forall i$$

What is width(w) ?



$$w^T x = -1$$

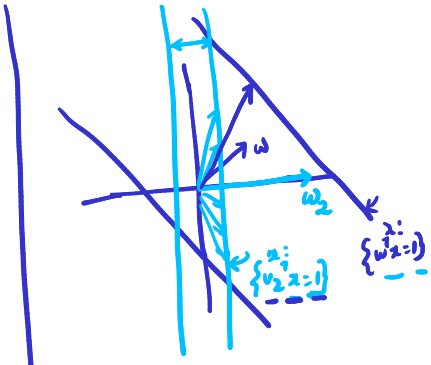
$$\min_{z} \quad \frac{1}{2}\|x - z\|^2$$

$$\text{s.t} \quad w^T z = +1$$

Solution: $\text{width}(w) = \dfrac{2}{\|w\|^2}$

$$\max_{w} \quad \frac{2}{\|w\|^2}$$

$$\text{s.t} \quad (w^T x_i) y_i \geq 1$$



Equivalently

$$\min_{\omega} \quad \frac{1}{2}\|\omega\|^2$$

$$\text{s.t} \quad (\omega^T x_i) y_i \geq 1 \quad \forall i$$

$$\min_{w \in \mathbb{R}^d} \ \frac{1}{2}\|w\|^2$$
$$\text{s.t} \ \ \underline{\forall i} \ \ (w^T x_i) y_i \geq 1$$

— (A)

DETOUR

$$\min_{w} \ f(w)$$
$$\text{s.t} \ \ g(w) \leq 0$$

←

$$\mathcal{L}(w, \alpha) = f(w) + \alpha \, g(w)$$

Fix any w.

Consider $\max_{\alpha \geq 0} \mathcal{L}(w, \alpha) = \max_{\alpha \geq 0} f(w) + \underline{\alpha} \, g(w)$ ←

$f(w)$	$g(w)$
w : $[1\ 2\ 3\ 4]$: -100	5

$\max_{\alpha \geq 0} -100 + \alpha 5$ ↑

$\alpha = 1 \ \ -95$
$\alpha = 10 \ \ -50$
$\alpha = 100 \ \ 400$
⋮

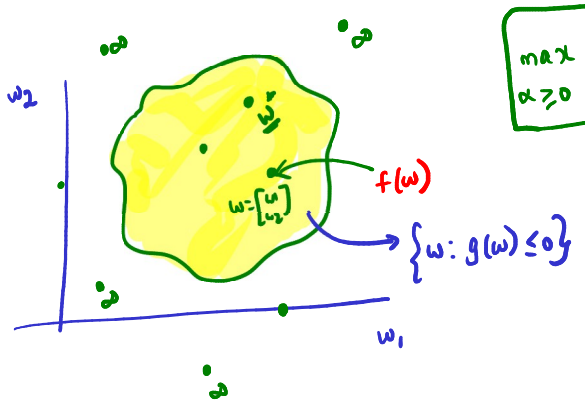
w	$f(w)$	$g(w)$
$[3\ 4\ 5\ 1]$	100	-5

$100 - 5\alpha$

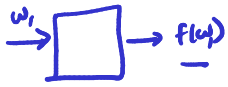
$\alpha = 1 \Rightarrow 95$
$\alpha = 10 \Rightarrow 50$ ↓
$\alpha = 0 \Rightarrow 100$

$$= \begin{cases} \infty & g(w) > 0 \\ f(w) & g(w) \leq 0 \end{cases}$$

$$\max_{\alpha \geq 0} \ f(w) + \alpha \, g(w)$$



$g(w_1) \leq 0$.



$g(w_2) > 0$

↓

$$\min_{w} \ f(w) \equiv \min_{w} \left[\max_{\alpha \geq 0} \ f(w) + \alpha \, g(w) \right]$$

- Can we swap min and max in (B)?
- In general, No! But if f and g are "nice" functions [convex functions], then yes!
  - ↳ [Quadratic / Linear]

$$\min_{w}\left[\max_{\alpha\geq 0} f(w)+\alpha g(w)\right] \equiv \max_{\alpha\geq 0}\left[\min_{w} f(w)+\alpha g(w)\right]$$

for convex f and g.

For multiple constraints

$$\min_{w} f(w) \quad \text{s.t} \quad g_i(w)\leq 0 \;\; \forall i=1\dots k$$

$$\equiv \min_{w}\left[\max_{\alpha_1,\dots,\alpha_k \geq 0} f(w)+\alpha_1 g_1(w)+\alpha_2 g_2(w)+\cdots+\alpha_k g_k(w)\right]$$

$$\Downarrow$$

$$\max_{\alpha_1\geq 0,\dots\alpha_k\geq 0}\left[\min_{w}\left[f(w)+\alpha_1 g_1(w)+\cdots+\alpha_k g_k(w)\right]\right]$$

$$\min_{w}\ \frac{1}{2}\|w\|^2 \leftarrow \text{Quadratic in } w$$

$$\text{s.t} \quad (w^T x_i)y_i \geq 1 \quad \forall i=1,\dots,n \leftarrow \text{Linear in } w$$

$$\equiv \underbrace{1-(w^T x_i)y_i}_{g_i(w)} \leq 0 \quad \forall i=1,\dots n$$

$$\alpha=\begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_n\end{bmatrix}$$

$$\mathcal{L}(w,\alpha) = \frac{1}{2}\|w\|^2+\sum_{i=1}^{n}\alpha_i\left(1-(w^T x_i)y_i\right)$$

$$\min_{w}\max_{\alpha\geq 0}\left[\frac{1}{2}\|w\|^2+\sum_{i=1}^{n}\alpha_i\left(1-(w^T x_i)y_i\right)\right] \equiv \max_{\alpha\geq 0}\left[\min_{w}\left[\frac{1}{2}\|w\|^2+\sum_{i=1}^{n}\alpha_i\left(1-(w^T x_i)y_i\right)\right]\right]$$

Fix some $\alpha \geq 0$. $\alpha = \begin{bmatrix} 1 \\ 5 \\ 7 \end{bmatrix}$

$$\min_w \underbrace{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i \left(1 - (w^T x_i) y_i\right)}$$

$$w_\alpha^* + \sum_{i=1}^{n} \alpha_i (-x_i y_i) = 0$$

$$w_\alpha^* = \sum_{i=1}^{n} \alpha_i x_i y_i \quad \in \mathbb{R}^d$$

$\alpha_i \geq 0$, $\{\pm 1\}$

→ Substitute back value of w_α^* in the objective

$$w_\alpha^* = X y \alpha$$

$$X = \underbrace{\begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{bmatrix}}_{d \times n} \underbrace{\begin{bmatrix} y_1 & & 0 \\ & \ddots & \\ 0 & & y_n \end{bmatrix}}_{n \times n} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}_{n \times 1}$$

$$\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i \left(1 - (w^T x_i) y_i\right)$$

Substitute $w_\alpha^* = Xy\alpha$ into

on simplification

$$= \alpha^T \underbrace{\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n \times 1}} - \frac{1}{2}(Xy\alpha)^T(Xy\alpha)$$

DUAL PROBLEM

$$\max_{\alpha \geq 0} \quad \alpha^T 1 - \frac{1}{2} \alpha^T y^T \boxed{X^T X} y \alpha \qquad \mathbb{R}^{n \times n}$$

kernel K

PRIMAL

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|w\|^2$$

s.t. $(w^T x_i) y_i \geq 1$

What have we gained?

- Dual variable dimension in $\mathbb{R}^n_+$ while primal problem dimension is $\mathbb{R}^d$
- Dual constraints are "easier"
- More importantly dual depends on $x^T x$ and so can be "KERNELIZED"!

$$w^*_{\alpha^*} = \sum_{i=1}^{n} \alpha_i^* x_i y_i$$

→ This says optimal w^* is a linear combination of the data points where importance of a datapoint is given by α_i^* (for i^{th} data point)

→ Question: Where are the "IMPORTANT" points? (i.e., points for which $\alpha_i^* > 0$)

REVISITING THE LAGRANGIAN

$$\min_w \left[\max_{\alpha \geq 0} f(w) + \alpha g(w) \right] \equiv \max_{\alpha \geq 0} \left[\min_w f(w) + \alpha g(w) \right]$$

(Primal) (Dual)

w^* is the primal solution

α^* is the dual solution

$$\max f(w^*) + \alpha g(w^*) = \min_w f(w) + \alpha^* g(w)$$

$$f(w^*) = \min_w f(w) + \alpha^* g(w)$$
$$\leq f(w^*) + \alpha^* g(w^*)$$

$$\Rightarrow f(w^*) \leq f(w^*) + \alpha^* g(w^*)$$

$$\Rightarrow \boxed{\alpha^* g(w^*) \geq 0} \quad -(1)$$

But we already know $\alpha^* \geq 0$ & $g(w^*) \leq 0$

$$\Rightarrow \boxed{\alpha^* g(w^*) \leq 0} \quad (2)$$

$$(1) \& (2) \Rightarrow \boxed{\alpha^* g(w^*) = 0} \rightarrow \text{COMPLEMENTARY SLACKNESS}$$

For multiple constraints,

$$\boxed{\alpha_i^* g_i(w^*) = 0 \quad \forall i}$$

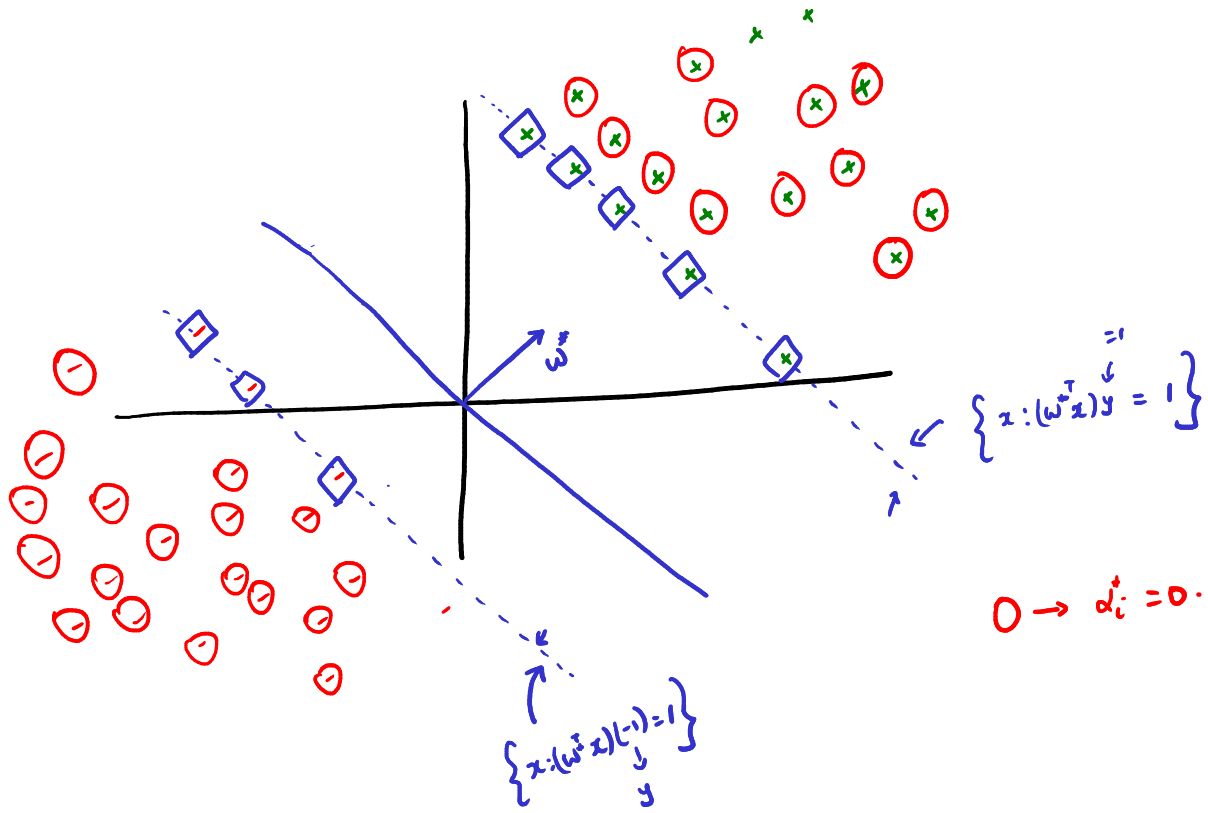
In our problem

$$\alpha_i^* \left(1 - (w^{*T} x_i) y_i\right) = 0 \quad \forall i \quad \text{[by complementary slackness]}$$

$$\Rightarrow \text{If } \alpha_i^* > 0 \overset{\text{C.S}}{\Rightarrow} 1 - (w^{*T} x_i) y_i = 0$$

$$\Downarrow$$

$$\boxed{(w^{*T} x_i) y_i = 1}$$



- only the points that are on the "SUPPORTING" hyperplane can contribute to w^*
- These special points are called "SUPPORT VECTORS"
- ALGORITHM → SUPPORT VECTOR MACHINE (SVM) [Vapnik et. al]
- w^* is a sparse linear combination of the data points.

Given x_{test} =

$$w^{*T} x_{test} = \left(\sum_{i=1}^{n} \alpha_i^* x_i y_i \right)^T x_{test}$$

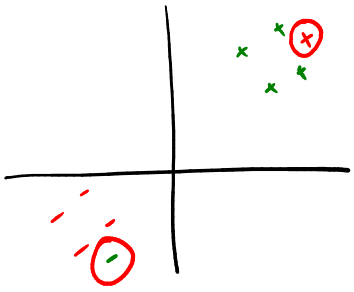
$$= \sum_{i=1}^{n} \alpha_i^* y_i (x_i^T x_{test})$$

$$w^{*T} \phi(x_{test}) = \sum_{i=1}^{n} \alpha_i^* y_i \, k(x_i, x_{test})$$

x_{test}

QUESTIONS

- How to adapt the SVM algorithm when data has outliers.



- KERNELS can help but is not the right way to solve this!

$$\min_{w} \quad \frac{1}{2}\|w\|^2$$

$$\text{s.t} \quad (w^T x_i) y_i \geq 1 \quad \forall i$$

Insight: Make every w feasible.

- Fix any w. w classifies some points correctly and misclassifies some points
- The incorrectly classified points "pay bribe" to go to the "correct" side!

MODIFIED FORMULATION

≥ 0 → HYPER PARAMETER.

SOFT MARGIN PRIMAL FORMULATION ←

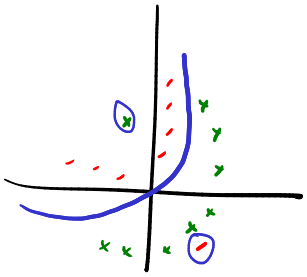
$$\min_{w,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{s.t} \quad (w^T x_i) y_i + \xi_i \geq 1 \quad \forall i$$

$$\xi_i \geq 0 \quad \forall i$$

$C = 0$
⇒ Bribes don't cost
⇒ $w = 0 \in \mathbb{R}^d$ is the solution

$C = \infty$
⇒ Linear separable case.



$$\min_{w,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t} \quad (w^T x_i)y_i + \xi_i \geq 1 \quad \forall i \qquad \leftarrow \; 1 - w^T x_i y_i - \xi_i \leq 0$$

$$\xi_i \geq 0 \quad \forall i \qquad -\xi_i \leq 0$$

$$L(w,\xi,\alpha,\beta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(1 - w^T x_i y_i - \xi_i) + \sum_{i=1}^{n}\beta_i(-\xi_i)$$

DUAL PROBLEM

$$\min_{w,\xi}\left[\max_{\substack{\alpha\geq 0\\ \beta\geq 0}} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(1 - w^T x_i y_i - \xi_i) + \sum_{i=1}^{n}\beta_i(-\xi_i)\right]$$

III Duality

$$\max_{\substack{\alpha\geq 0\\ \beta\geq 0}}\left[\min_{w,\xi}\left[\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(1 - w^T x_i y_i - \xi_i) + \sum_{i=1}^{n}\beta_i(-\xi_i)\right]\right]$$

$$\left[\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(1 - w^T x_i y_i - \xi_i) + \sum_{i=1}^{n}\beta_i(-\xi_i)\right] = L$$

Fix α, β

$$\frac{\partial L}{\partial w} = 0 \Rightarrow \boxed{w^*_{\alpha,\beta} = \sum_{i=1}^{n}\alpha_i x_i y_i} \quad -(1)$$

$w^*_{\alpha,\beta} = X Y \alpha$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow C + \alpha_i(-1) + \beta_i(-1) = 0$$

$$\Rightarrow \boxed{\alpha_i + \beta_i = C} \quad -(2)$$

Back Substituting $w^*_{\alpha,\beta}$ into the Lagrangian

Dual problem

$$\max_{\alpha \geq 0,\ \beta \geq 0,\ \alpha+\beta = C1} \quad \alpha^T 1 - \frac{1}{2}\alpha^T y^T x^T x\, y\, \alpha$$

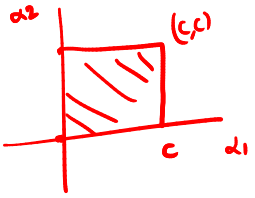
↑ no β term

$\alpha_i + \beta_i = C \ \forall i$

$\Downarrow\Downarrow$

$$\max_{0 \leq \alpha \leq C} \quad \alpha^T 1 - \frac{1}{2}\alpha^T y^T x^T x\, y\, \alpha$$

Kernelizable



Box constraints.

If $C = 0 \Rightarrow \alpha^* = 0 \in \mathbb{R}^n \Rightarrow w^* = \sum \alpha_i^* x_i y_i$

$\Rightarrow \boxed{w^* = 0}$

$C = \infty \Rightarrow$ Hard-margin

What do the COMPLEMENTARY SLACKNESS (C.S) say about the Soft-margin SVM?

Let (w^*, ξ^*) be the primal optimal solutions.

Let (α^*, β^*) be the dual optimal solutions.

C.S

(1) $\forall i \quad \alpha_i^* (1 - w^{*T} x_i y_i - \xi_i^*) = 0$ ←

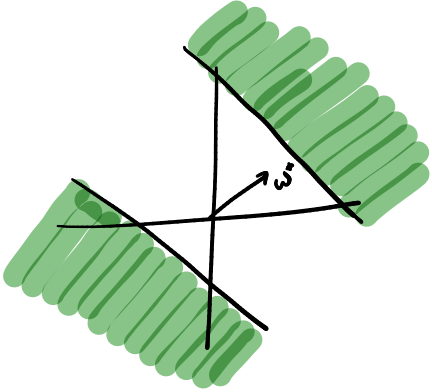
(2) $\forall i \quad \beta_i^* (\xi_i^*) = 0$

Various cases possible

(1) $\alpha_i^* = 0 \Rightarrow \beta_i^* = C \quad [\alpha_i^* + \beta_i^* = C]$

$\Downarrow$ C.S (2)

$\xi_i^* = 0$



we know $w^{*T} x_i y_i + \xi_i^* \geq 1$

$\Rightarrow$ $w^{*T} x_i y_i \geq 1$ $\Rightarrow$ w^* classifies (x_i, y_i) correctly !

② $\alpha_i^* \in (0, C)$ $\quad 0 < \alpha_i^* < C$

$\Rightarrow$ $\beta_i^* \in (0, C)$ $\quad 0 < \beta_i^* < C$

$\Rightarrow$ $\xi_i^* = 0$ [c.s ②]

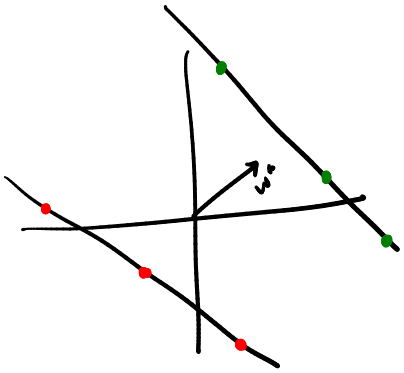
$\Downarrow$ [c.s ①]

$1 - w^{*T} x_i y_i - \xi_i^* = 0$

$\Downarrow$

$1 - w^{*T} x_i y_i = 0$

$\Rightarrow$ $\boxed{w^{*T} x_i y_i = 1}$



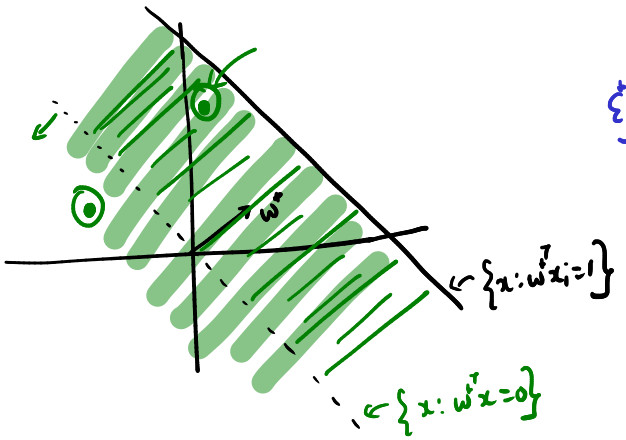
③ $\alpha_i^* = C$ $\xRightarrow{\text{c.s ①}}$ $1 - w^{*T} x_i y_i - \xi_i^* = 0$

$\Downarrow$

$\beta_i^* = 0$ $\qquad \xi_i^* = 1 - w^{*T} x_i y_i \geq 0$

$\Downarrow$

$\xi_i^* \geq 0$ $\Rightarrow$ $\boxed{w^{*T} x_i y_i \leq 1}$



Points where either x_i is incorrectly classified by w^* or correctly classified but with margin ≤ 1

Let's see this from the primal point of view.

Case ①

$w^{*T} x_i y_i < 1$

$w^{*T} x_i y_i + \xi_i^* \geq 1$ $\Rightarrow$ $\xi_i^* \geq 1 - w^{*T} x_i y_i$

$\Rightarrow$ $\xi_i^* > 0$ $\xRightarrow{\text{c.s ②}}$ $\beta_i^* = 0$ $\Rightarrow$ $\boxed{\alpha_i^* = C}$

Case ②

$$w^{*T} x_i y_i = 1$$

$$\xi_i^* \geq 1 - w^{*T} x_i y_i$$

$$\xi_i^* \geq 0 \Rightarrow \alpha_i^* \in [0, C]$$

Case ③

$$w^{*T} x_i y_i > 1$$

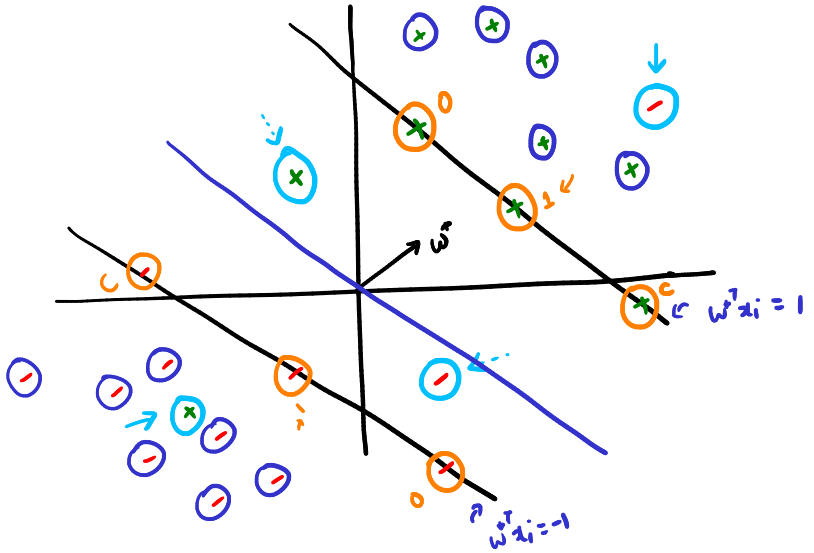
$$\Rightarrow \quad 1 - w^{*T} x_i y_i - \xi_i^* < 0$$

cs ① $\Rightarrow$ $\alpha_i^* = 0$

SUMMARY

$\alpha_i^* = 0$	$\Rightarrow$	$w^{*T} x_i y_i \geq 1$
$0 < \alpha_i^* < C$	$\Rightarrow$	$w^{*T} x_i y_i = 1$
$\alpha_i^* = C$	$\Rightarrow$	$w^{*T} x_i y_i \leq 1$

$w^{*T} x_i y_i < 1$	$\Rightarrow$	$\alpha_i^* = C$
$w^{*T} x_i y_i = 1$	$\Rightarrow$	$0 \leq \alpha_i^* \leq C$
$w^{*T} x_i y_i > 1$	$\Rightarrow$	$\alpha_i^* = 0$



$0 \rightarrow \alpha_i^* = 0$

$0 \rightarrow \alpha_i^* = C$

$0 \rightarrow \alpha_i^* \in [0, C]$

So-far

Generative

- Naive-Bayes
- G.D.A / G.N.B

Discriminative

- K-NN
- Decision trees
- Logistic regression
- Perceptron
- Support-vector machines.

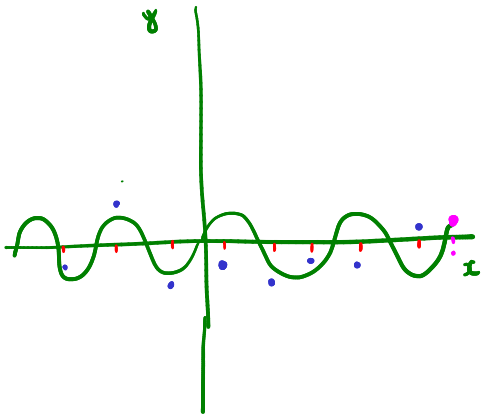
Goal:- Meta Classifiers (or) Ensemble Classifiers.

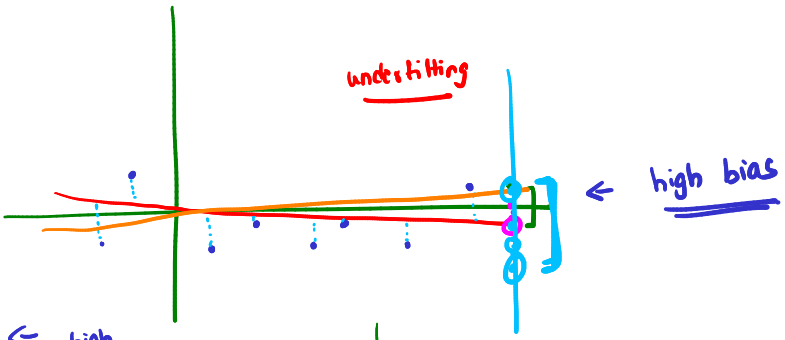
WEAK LEARNERS ⟶ STRONG LEARNERS.

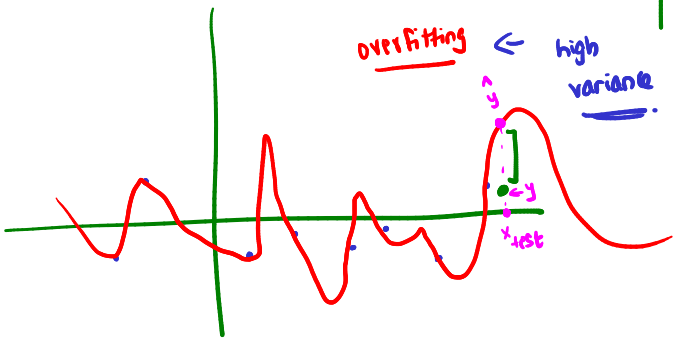
(better than random)

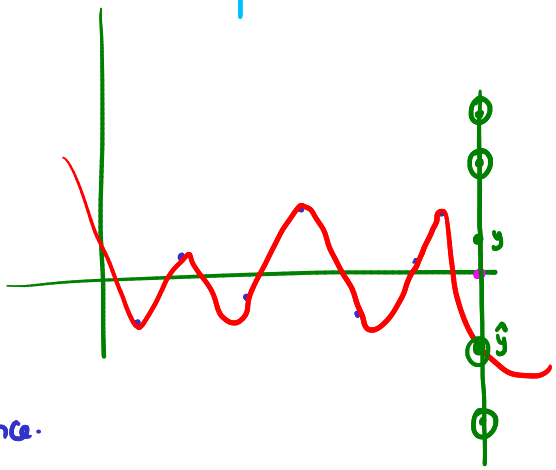
Overfitting – fit noise

Underfitting – missing out on structure thinking it is noise.









Error = bias + variance.

BAGGING - BOOTSTRAP AGGREGATION.

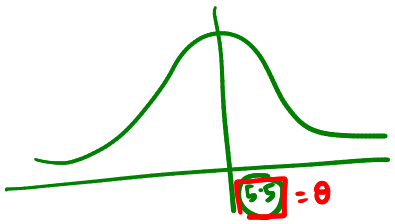
$\{x_1, x_2, \cdots, x_n\} \sim N(\mu, 1)$

$\hat{\mu}_1 = x_1$
$\hat{\mu}_2 = x_2$
$\vdots$
$\hat{\mu}_n = x_n$

$\hat{\mu}_{ML} = \frac{1}{n}\sum_{i=1}^{n} x_i$

$E[\hat{\mu}_{ML}] = \mu$

$E[\hat{\theta}] = \theta$ ← unbiased estimators



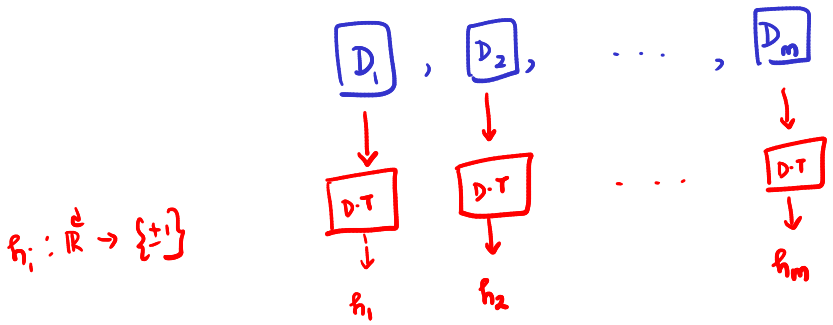
$\{5, 7.3, 6.7, 8.2, \cdots, 9\}$

$\hat{\mu}_1 = 5$
$\hat{\mu}_2 = 7.3$
$\vdots$
$\hat{\mu}_n = 9$

$E[\hat{\mu}_1] = E[x_1] = \mu$

"Averaging reduces variance"

Bagging



$h_i : \mathbb{R}^d \to \{\pm 1\}$

Each D_i has n datapoints.

$D_i = \{x_1', x_2', \cdots, x_n'\}$
$y_1', y_2', \cdots, y_n'$

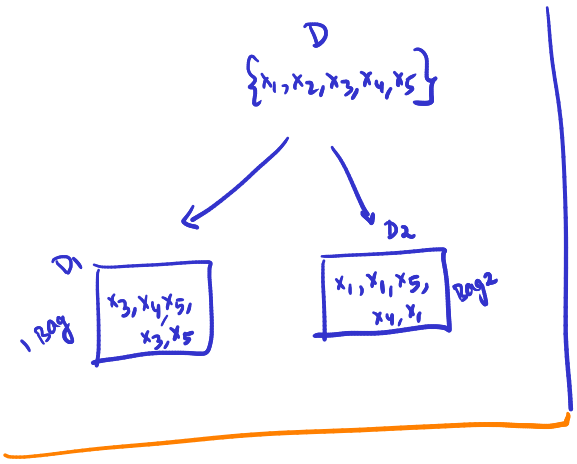
$x_i' \in \mathbb{R}^d$, $y_i' \in \{\pm 1\}$

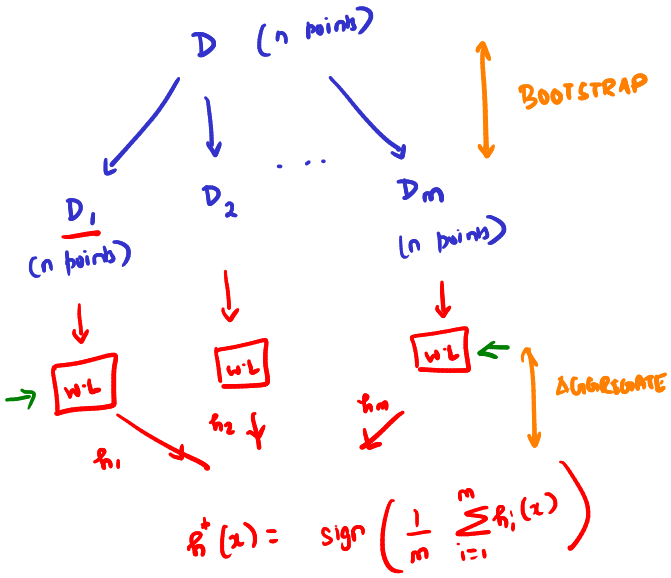
$h^*(x) = \text{Sign}\left(\frac{1}{m}\sum_{i=1}^{m} h_i(x)\right)$

$\text{Sign}(z) = +1$ if $z \geq 0$
$= -1$ o/w.

Input $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ $x_i \in \mathbb{R}^d$, $y_i \in \{\pm 1\}$

Bootstrapping → Sampling with replacement



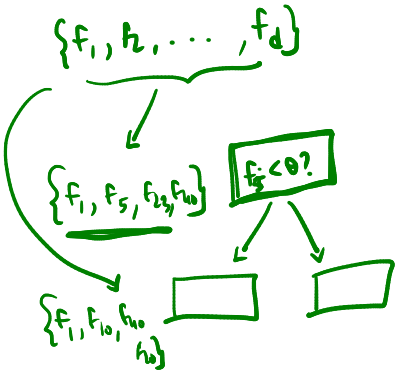


$$P_B\left(x_j \in D_i\right) = 1 - \underbrace{\left(1 - \frac{1}{n}\right)^n}_{} \approx 1 - \frac{1}{e}$$

$\approx$ 67% for large n

Bagging reduces variance!

Random Forest



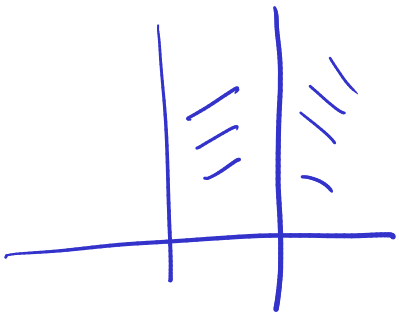
- Bags Decision trees (typically overfit trees)
- Feature bagging ($\sqrt{d}$)

- Bootstrapping – Sampling with replacement (uniform)
- Bagg – Averaging

BOOSTING

BOOSTING [ADABOOST]

↳ FREUND & SCHAPHIRE



Weak Learner ⟶ Strong Learner

Decision Stumps

1-Level or 2-Level

ADA-BOOST ALGORITHM

↳ Adaptive.

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ $\quad x_i \in \mathbb{R}^d$, $y_i \in \{\pm 1\}$

Initialize $D_0(i) = \frac{1}{n}$ (iteration)

for $t = 1, \ldots, T$

- h_t = Input (S, D_t) to a weak learner to get h_t

  $h_t : \mathbb{R}^d \to \{\pm 1\}$

- $$\tilde{D}_{t+1}(i) = \begin{cases} D_t(i) \cdot e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \\ D_t(i) \cdot e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \end{cases}$$

- $$D_{t+1}(i) = \frac{\tilde{D}_{t+1}(i)}{\sum_j \tilde{D}_{t+1}(j)}$$

end

	x_1	x_2	x_3	
D_0	0.33	0.33	0.33	←
D_1	0.3	0.5	0.2	→ h_1
	✗	✓		
D_2	$\frac{0.3 \times e^{\alpha_1}}{Z}$	$\frac{0.5\, e^{-\alpha_1}}{Z}$	$\frac{0.2\, e^{-\alpha_1}}{Z}$	

$Z = 0.3e^{\alpha_1} + 0.5e^{-\alpha_1} + 0.2e^{-\alpha_1}$

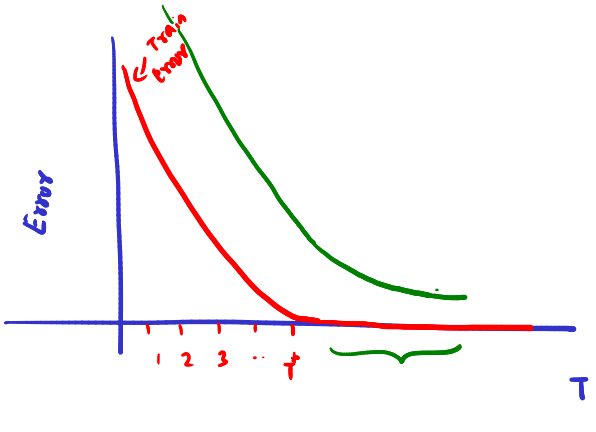
$h_1, h_2, \ldots, h_T$ $\quad$ $h_T^*(x) = \text{Sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$ $\quad$ $\alpha_t = \ln\sqrt{\frac{1-\text{err}(h_t)}{\text{err}(h_t)}}$

One can prove

If $T \geq \frac{1}{2\gamma^2} \ln(2n)$, then

Training error = 0

How good is my weak learner?



- Can't run in parallel

Dataset = $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ $\quad x_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$

Goal: $h : \mathbb{R}^d \to \{\pm 1\}$

Performance measure

$$\sum_{i=1}^{n} \mathbb{1}(h(x_i) \neq y_i)$$

$\mathbb{1}(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{o/w} \end{cases}$

$$\min_{h \in \mathcal{H}_{\text{linear}}} \sum_{i=1}^{n} \mathbb{1}(h(x_i) \neq y_i)$$

$$= \min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \mathbb{1}(\text{sign}(w^T x_i) \neq y_i)$$

↳ NP-HARD

$x \in \mathbb{R}^d$

$x' = [x \; 1] \in \mathbb{R}^{d+1}$

$w' = [w_1 \cdots w_d \; (w_{d+1})]$ (w: $w_1 \cdots w_d$, b: w_{d+1})

$w'^T x' = w^T x + b$

LOSS-FUNCTION VIEW

$(x, y) \in \mathbb{R}^d \times \{\pm 1\}$, $h : \mathbb{R}^d \to \{\pm 1\}$

$h(x) = \text{sign}(w^T x)$

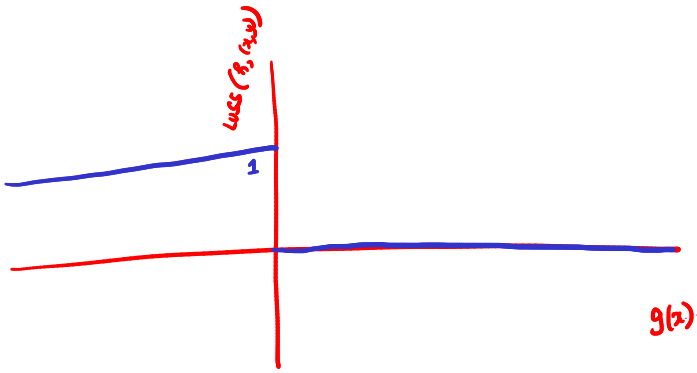
$$\mathbb{1}(h(x) \neq y) = \mathbb{1}((w^T x) \cdot y < 0)$$

($w^T x \geq 0 \Rightarrow +1$, $w^T x < 0 \Rightarrow -1$)

$< 0 \; -1$

$> 0 \; -1$

$< 0 \; +1$



$h(x) = \text{sign}(g(x))$

$$\sum_{i=1}^{n} \mathbb{1}(g(x_i) y_i < 0)$$

1

ALG 1: Using Regression for classification $\quad h(x) = \text{sign}(g(x))$

$$\text{Loss}(g, (x, y)) = \underbrace{(g(x) - y)^2}_{y \in \{\pm 1\}} = (g(x))^2 + y^2 - 2g(x) \cdot y$$

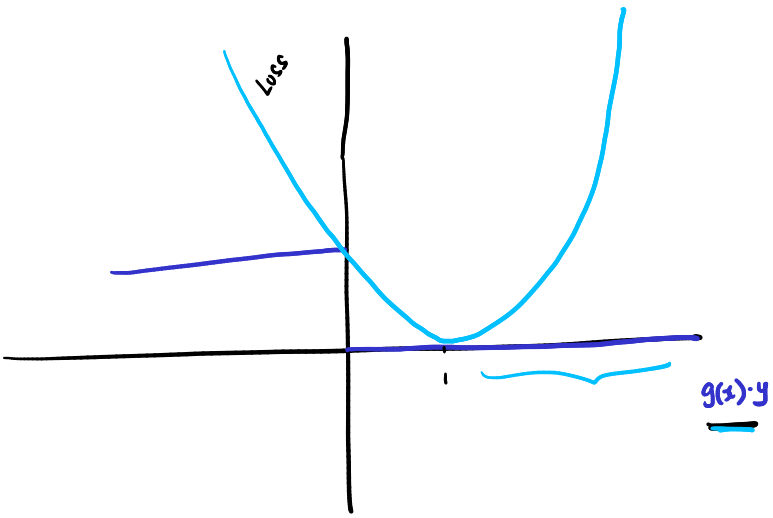
$$= (g(x))^2 + 1 - 2g(x) \cdot y \quad -(1)$$

$$= \boxed{(g(x) y - 1)^2} \quad \text{(since } y^2 = 1\text{)}$$

$$= (g(x) \cdot y)^2 + 1 - 2g(x) \cdot y$$

$= (g(x))^2 + 1 - 2g(x)\cdot y$ — ②

① = ②.



- → 0-1 loss
- → $(g(x)\cdot y - 1)^2$ → SQUARED LOSS

SUPPORT VECTOR MACHINES

$$\min_{w,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t} \quad (w^T x_i)y_i + \xi_i \geq 1$$

$$\xi_i \geq 0$$

$$\min_{w,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t} \quad \begin{cases} \xi_i \geq 1 - (w^T x_i)y_i \\ \xi_i \geq 0 \end{cases}$$

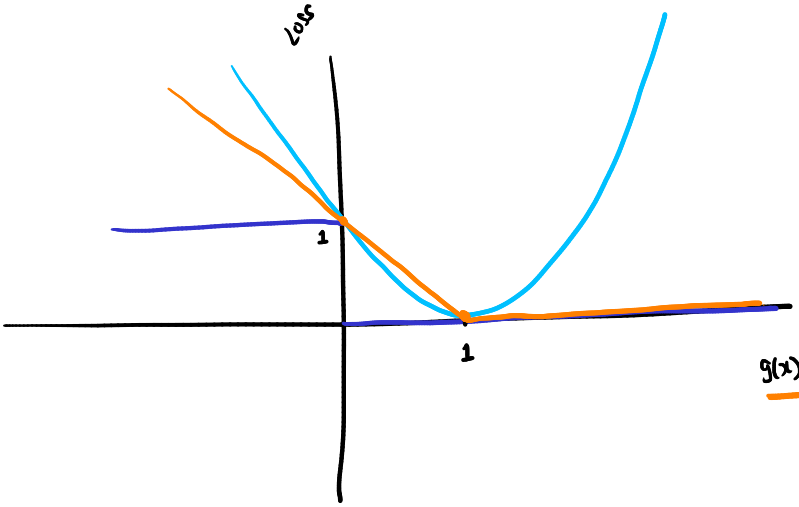
$$\xi_i \geq \max(0, 1-(w^T x_i)y_i)$$

$$= \min_{w \in \mathbb{R}^d} \underbrace{\frac{1}{2}\|w\|^2}_{\text{Regularizer}} + \underbrace{C\sum_{i=1}^{n}\max(0, 1-(w^T x_i)y_i)}_{\text{Loss}}$$

$(w^T x_i)y_i \rightarrow g(x_i)$

Model dependent — REGULARIZATION +

Data dependent — HINGE LOSS



- 0-1 loss
- Squared loss $(g(x)-1)^2$
- $\max(0, 1-g(x)\cdot y)$ HINGE LOSS

LOGISTIC REGRESSION

$$\sigma(a) = \frac{1}{1+e^{-a}}$$

$z_i \in \{0, 1\}$

$$\max_{w} \prod_{i=1}^{n} \left(\sigma(w^T x_i)\right)^{z_i} \left(1 - \sigma(w^T x_i)\right)^{(1-z_i)}$$

$z_i = 1$ if $y_i = +1$

$z_i = 0$ if $y_i = -1$

$$\max_{w} \sum_{i=1}^{n} z_i \log\left(\sigma(w^T x_i)\right) + (1-z_i) \log\left(1 - \sigma(w^T x_i)\right)$$

$$= \min_{w} \sum_{i=1}^{n} \left[-z_i \log\left(\sigma(w^T x_i)\right) + (z_i - 1) \log\left(1 - \sigma(w^T x_i)\right) \right]$$

Loss for a single point $z_i = 1$ $(y_i = +1)$

$$= -\log\left(\sigma(w^T x_i)\right) = -\log\left(\frac{1}{1+e^{-w^T x_i}}\right)$$

$$= \log\left(1 + e^{-w^T x_i}\right) = \boxed{\log\left(1 + e^{-(w^T x_i) y_i}\right)}$$

Loss for a single point $z_i = 0$ $(y_i = -1)$

$$-\log\left(1 - \sigma(w^T x_i)\right)$$

$$= -\log\left(1 - \frac{1}{1+e^{-w^T x_i}}\right)$$

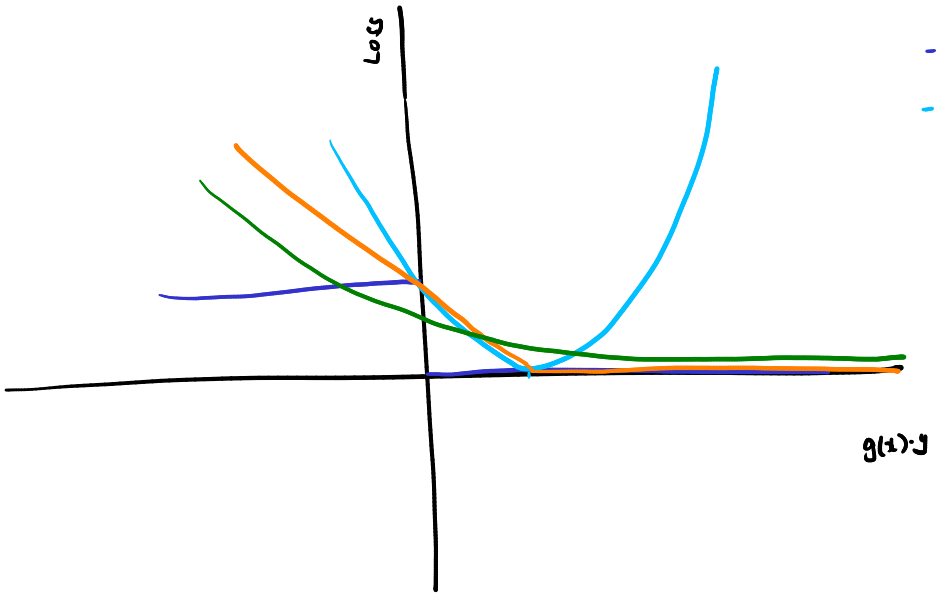
$$= -\log\left(\frac{e^{-w^T x_i}/e^{-w^T x_i}}{(1+e^{-w^T x_i})/e^{-w^T x_i}}\right)$$

$$= -\log\left(\frac{1}{e^{w^T x_i} + 1}\right)$$

$$= \log\left(1 + e^{w^T x_i}\right)$$

$$= \boxed{\log\left(1 + e^{-w^T x_i y_i}\right)}$$

$$= \min_{w} \sum_{i=1}^{n} \log\left(1 + e^{-\underbrace{w^T x_i y_i}_{g(x_i) y_i}}\right)$$



- 0-1 loss $\mathbb{1}(g(x)\,y < 0)$
- $(g(x)y - 1)^2$ SQ. loss
- $\max(1 - g(x)\cdot y, 0)$
- $\log(1 + e^{-g(x)\cdot y})$ LOGISTIC LOSS.

CONCLUSIONS

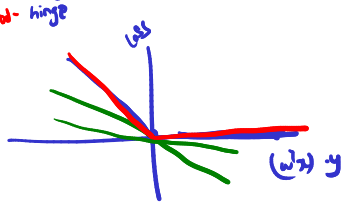
- 0-1 loss is NP-hard to minimize
- Different algorithms use different "surrogate" loss
- Surrogates are convex and hence easy to minimize.

PERCEPTRON

$$w_{t+1} = w_t + x_t y_t$$

HINGE LOSS

$$\underset{\text{aka hinge}}{\ell}(w, (x,y)) = \max(0, -(w^T x)y)$$



$$\nabla_w \ell_{hinge} = \begin{cases} -xy & (w^T x)y < 0 \quad \text{(mistake)} \\ 0 & (w^T x)y > 0 \\ & (w^T x)y = 0 \end{cases}$$

$[-1, 0]\, xy$

↳ chooses $-xy$ when mistake, 0 o/w.

$$w_{t+1} = w_t - \eta_t \nabla_w \text{loss}(w_t)$$

$$= w_t - (-x_t y_t)$$

- Perceptron can be interpreted as S.G.D with modified hinge loss with step size = 1

BOOSTING

- $\text{Loss}(h, (x,y)) = e^{-y h(x)}$ ↳ Exponential loss.

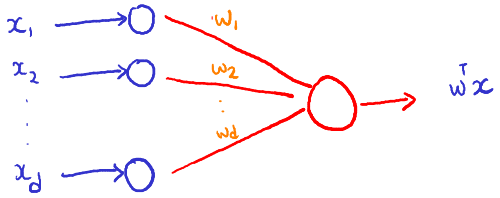
$$\min_{w} \sum_{i=1}^{n} \underbrace{L(w^T x_i, y_i)}_{\text{Loss}} + R(w)$$

↳ Regularizer

NEURAL NETWORKS

$x \in \mathbb{R}^d$ $\quad$ $\text{sign}(w^T x)$

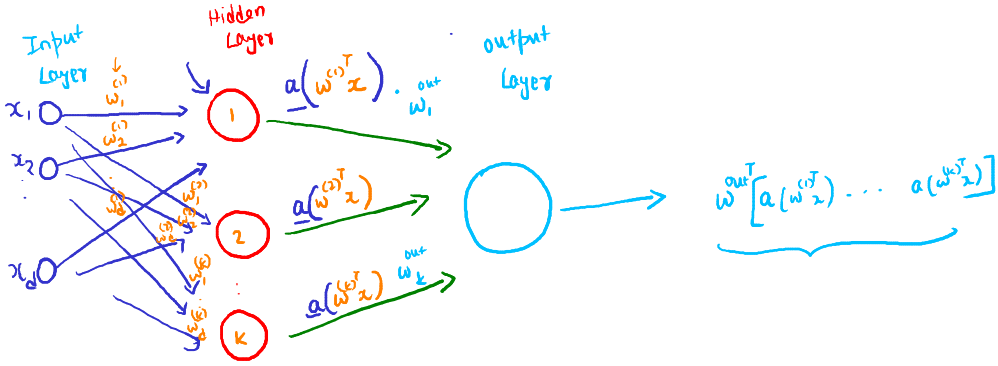
$[x_1\ x_2 \cdots x_d]$



PARAMETERS

$\{w^{(1)}, \cdots, w^{(k)}\}$ $\quad w^{(i)} \in \mathbb{R}^d$

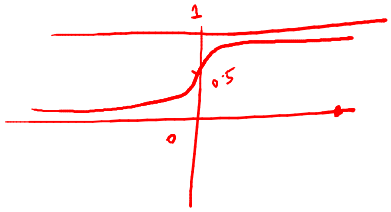
$w^{out} \in \mathbb{R}^k$



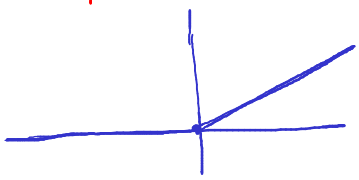
$\hat{y} = \sum_{i=1}^{k} w_i^{out} \, a(w^{(i)T}x)$

(# hidden layer nodes; activation function)

Examples of activation functions / non-linearities



- $a(z) = \dfrac{1}{1 + e^{-z}}$ [SIGMOID]



- $a(z) = \max(0, z)$ [Rectified Linear unit]

$\{w^{(1)}, \ldots, w^{(L)}, w^{out}\}$

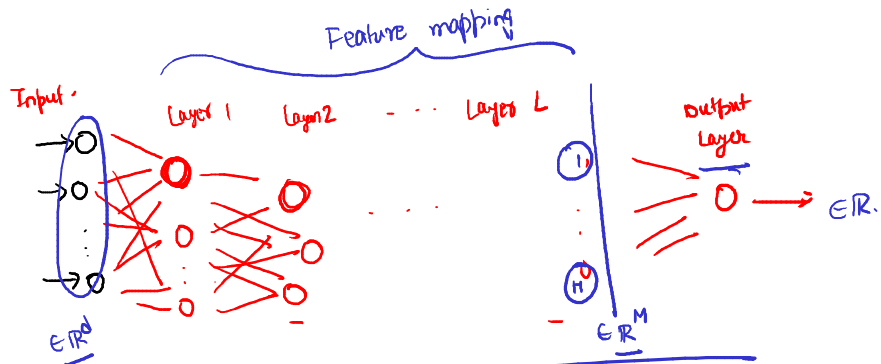
Regression

$$L\left(NN(x_i, \theta), y_i\right)$$

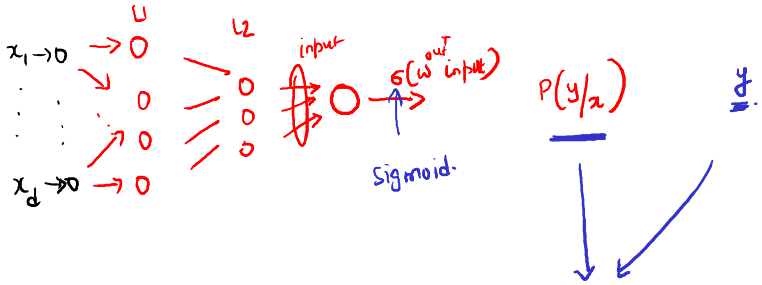
$$= \sum_{i=1}^{n} \left(NN(x_i;\theta) - y_i\right)^2$$

$$\equiv w^T x_i$$

- Learn θ^* using Gradient descent



- Gradient computed taking advantage of Chain rule → BACK-PROPAGATION
- Converges to local minima!



CROSS-ENTROPY LOSS

Conclusion

CNN, RNN, LSTM, Attentions, Transformers,

- NN Learns local minima of non-concave functions
- Typically works very well in practice.
  - ↳ especially for unstructured data.