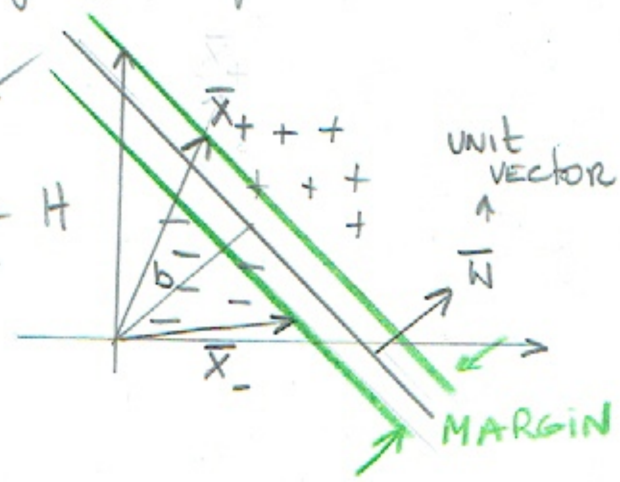
Support Vector Machines · Vladimir Vapnik 1994 ①

↳ classification algorithm (not so good for regression)
very popular
very powerful.



perceptron THEOREM

if there is a hyperplane that can separate the classes there are ∞ many!

↓

which one is the BEST ?!

↙

SVM: use hyperplane that maximizes the margin between the classes

$w^T x + b = 0$ ON H

| $\bar{w}$ is ⊥ H
| $\bar{w}$ is unit vector.

Assume 2 points x' and x'' on H.

$$\rightarrow \begin{array}{l} w^T x' + b = 0 \\ w^T x'' + b = 0 \quad - \\ \hline w^T \cdot (x' - x'') = 0 \end{array}$$

→ meaning w^T and $(x' - x'')$ are orthogonal.

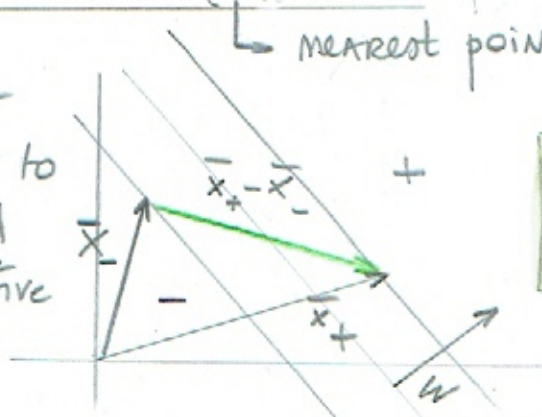
→ w^T is orthogonal to any vector on the plane

→ $w^T \perp H$.

distance of vector $\left\{ \begin{array}{l} \bar{x}_- \\ \bar{x}_+ \end{array} \right.$ to H ?

└ nearest point to the plane!

$\bar{x}_+$, $\bar{x}_-$ are vectors closest to the plane H in their respective classes.



$$\boxed{\text{Margin} = (\bar{x}_+ - \bar{x}_-) \cdot \frac{\bar{w}}{\|w\|}}$$

<u>set</u>: $\begin{cases} \bar{w}^T \cdot \bar{x}_+ + b \geq 1 \\ \bar{w}^T \cdot \bar{x}_- + b \leq -1 \end{cases} \rightarrow y_i(\bar{w} \cdot \bar{x} + b) \geq 1$ (set)

↳ $y_i = +1$ in $+$
$y_i = -1$ in $-$

(1) <u>set</u>: $\boxed{y_i(\bar{w}^T \cdot \bar{x}_i + b) = 0}$

↳ IN GUTTER

$(1) \rightarrow \bar{w}^T \cdot \bar{x}_+ = 1 - b \rightarrow \bar{x}_+ = \frac{1-b}{w}$

$\bar{w}^T \cdot \bar{x}_- \leq -1 - b \rightarrow \bar{x}_- = \frac{-(1+b)}{w}$

$\rightarrow \text{MARGIN} = \left[\left(\frac{1-b}{w}\right) + \left(\frac{1+b}{w}\right)\right] \cdot \frac{\bar{w}}{\|w\|}$

$= \frac{2}{\|w\|}$

<u>GOAL</u> $\rightarrow \text{MAX } \frac{2}{\|w\|} \Leftrightarrow \text{MAX } \frac{1}{\|w\|} \Leftrightarrow \min \|w\|$

$\Leftrightarrow \boxed{\min \frac{1}{2}\|w\|^2}$

subject to: $y_i(\bar{w}^T \cdot \bar{x} + b) \geq 1$

→ <u>constraint optimization Problem</u> → Lagrange multipliers

↳ inequality constraints

↳ KKT (KARUSH – KUHN – TUCKER)

<u>SVM</u>: minimize $\frac{1}{2}\|w\|^2$ WITH condition that all points are correctly classified!

LAGRANGE

$\min \frac{1}{2}\|w\|_2^2$ WITH CONSTRAINT $y_i(w^T x_i + b) \geq 1$

$- \sum_i \alpha_i \left[y_i(w^T x_i + b) - 1 \right]$

$$* \quad \boxed{L(w, b, \alpha) = \frac{1}{2}\|w\|_2^2 - \sum \alpha_i \left(y_i (w^T x_i + b) - 1 \right)}$$

MINIMIZE w, b

MAXIMIZE α.

WITH $\alpha_i \geq 0$ → because of INEQUALITY CONSTRAINT)

instead of MINIMIZING OVER w, b subject to constraints involving α, we can MAXIMIZE over α subject to relations obtained previously for w and b.

$$\frac{\partial L}{\partial w} = \bar{w} - \sum \alpha_i y_i x_i \overset{\text{SET}}{=} 0 \text{ (VECTOR)}$$

$$\rightarrow \boxed{\bar{w} = \sum \alpha_i y_i \bar{x}_i}$$

$$\frac{\partial L}{\partial b} = \boxed{\sum \alpha_i y_i \overset{\text{set}}{=} 0} \quad (**)$$

→ PUT IN (*)
→ free of w, b

$$(*) \quad L = \frac{1}{2}\left(\sum \alpha_i y_i \bar{x}_i\right)\left(\sum \alpha_j y_j x_j\right)$$
$$- \left(\sum \alpha_i y_i \bar{x}_i\right) \cdot \left(\sum \alpha_j y_j \bar{x}_j\right)$$
$$- \sum \alpha_i y_i b \rightarrow 0 \text{ AS PER } (**)$$
$$+ \sum \alpha_i$$

we can TRAIN A classifier in high D WITHOUT COMPUTING $\bar{w}, b$ that define H (***)

MAX $\alpha_i \alpha_j$ → $\boxed{L(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \, \bar{x}_i \cdot \bar{x}_j}$ — w, b free!

subject to: $\alpha_i \geq 0$ (KKT) $\quad \sum \alpha_i y_i = 0$

↳ MAX only depends on ON dot product of PAIRS of samples

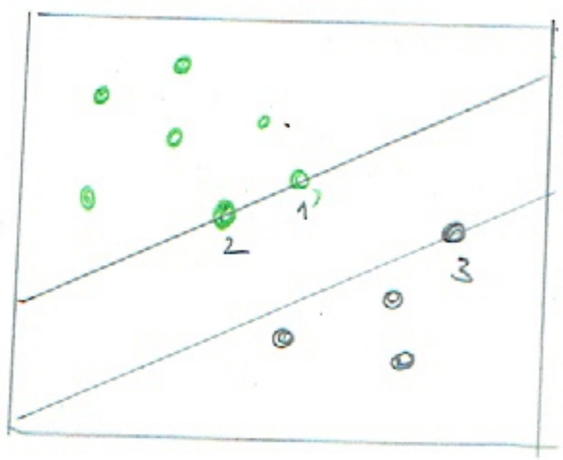
Support vectors

IN GUTTER $\rightarrow y_i(\bar{w}^T x_i + b) - 1 = 0$

$\underset{L(\alpha)}{\Rightarrow} \alpha > 0 \Rightarrow \boxed{\bar{w} = \sum_{x_i = SV} \alpha_i y_i x_i}$

→ All points outside gutter do not contribute to MAX $L(\alpha)$ ↳ $\alpha = 0$

→ Robust with respect to outliers!



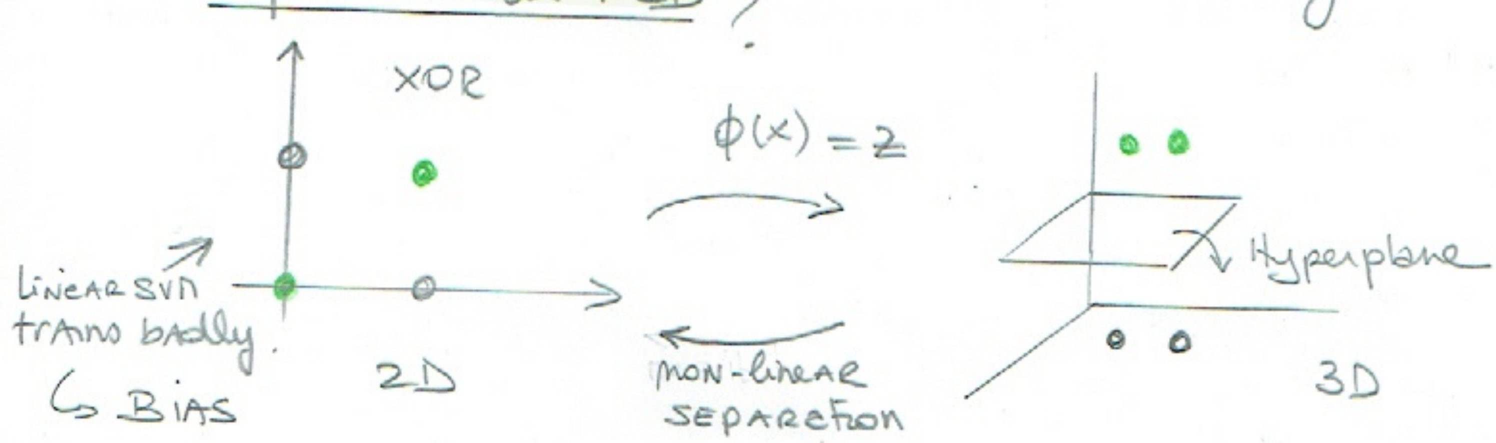
3 support vectors!
↓
Achieve the MARGIN!

$\bar{w} = \sum_{SV} \alpha_i y_i x_1$

W has d dimensions
but: only 3 αs contribute in above example!

Up to now we worked with linearly separable DATA.

→ what happens if the DATA is not linearly separable in 2D?



2D → 3D ⇒ d↑ ⇒ computationally more expensive

(***) p3 → $\boxed{L(\alpha) = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j y_i y_j \alpha_i \alpha_j z_i^T z_j}$

What if we go from 2D → 10^6D

* αs are related to number of training samples AND NOT dependent on dimension of $\bar{z}$

* $z_i^T z_j$ is not really a concern even if dimension z is 10^6

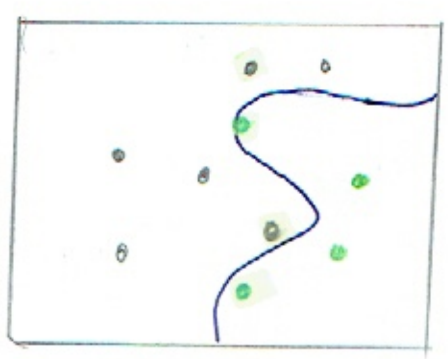
NOW: Assume we found hyperplane in Z-space

→ what happens in 2D-space

↳ support vectors live in Z-space!

we know which are the vectors that are support vectors! $\alpha \neq 0$

look for them in 2D



KERNEL trick

$\boxed{K(x,y) = \langle \phi(x), \phi(y) \rangle}$ → dot product = SCALAR

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

ϕ is function $d \to m$ with $m >>> d$

to calculate $\langle \phi(x), \phi(y) \rangle$ we would calculate $\phi(x), \phi(y)$ first and then do dot product

WITH Kernel → no need to go to m-dim space!

example $x = \begin{pmatrix} x_1 & 1 \\ x_2 & 2 \\ x_3 & 3 \end{pmatrix}$ $y = \begin{pmatrix} y_1 & 4 \\ y_2 & 5 \\ y_3 & 6 \end{pmatrix}$ $\in \mathbb{R}^{3=d}$

$$\phi(x) = (x_1x_1, x_1x_2, x_1x_3, x_2x_1, x_2x_2, x_2x_3, x_3x_1, x_3x_2, x_3x_3)$$

↳ 9 dot products

$$\boxed{K(x,y) = (\langle x,y \rangle)^2}$$

$$\phi(x) = (1, 2, 3, 2, 4, 6, 3, 6, 9)$$
$$\phi(y) = (16, 20, 24, 20, 25, 30, 24, 30, 36)$$

$$k(x,y) = \langle \phi(x), \phi(y) \rangle = 16 + 40 + 72 + 40 + 100 + 180 + 72 + 180 + 324 = 1024$$

$3D \longrightarrow 9D$

$$K(x,y) = (4 + 10 + 18)^2 = 1024$$

Same Result but much Faster!

Kernel functions

$K(x,y) = x^T y$ linear

$K(x,y) = (1 + x^T y)^P$ polynomial

$K(x,y) = e^{\frac{-(x-y)^2}{\sigma^2}}$ RBF (Radial Base Function)

very popular!
works out of the box
every problem becomes
linearly separable