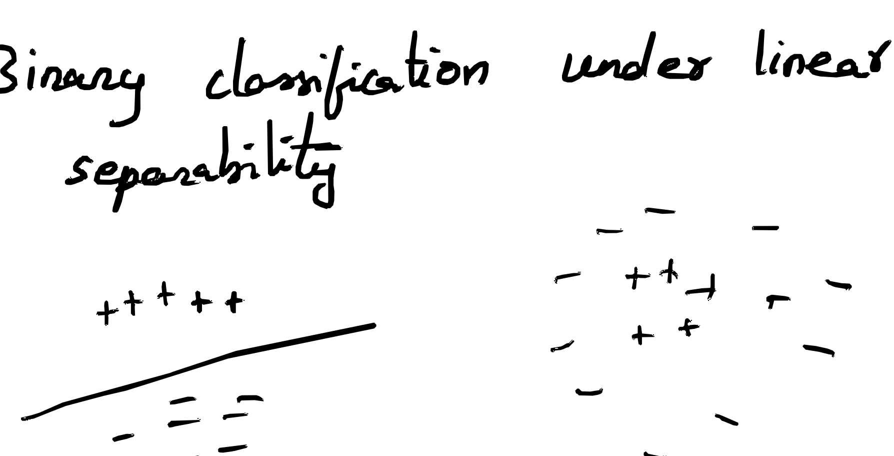
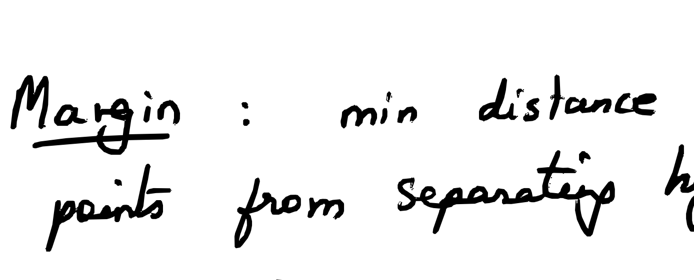
Perceptron

Binary classification under linear separability

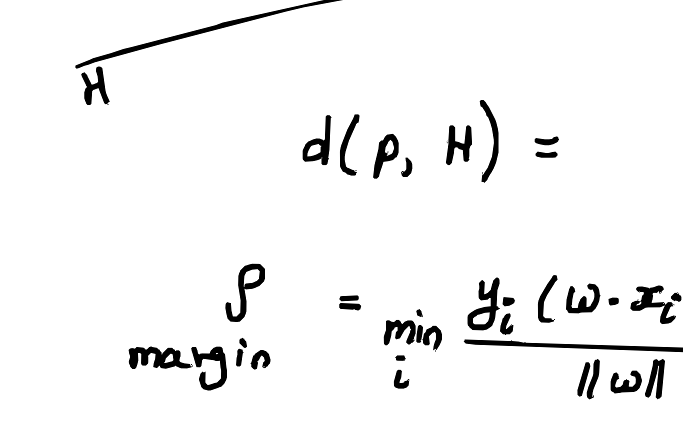


<u>Margin</u> : min distance of training points from separating hyperplane



$$d(p, H) = \frac{|w \cdot p + b|}{\|w\|}$$

$$\rho_{\text{margin}} = \min_i \frac{y_i (w \cdot x_i + b)}{\|w\|}$$



$[n] = \{1, \dots, n\}$ $\quad x_i \in \mathbb{R}^d \quad y_i = \pm 1$, $i = 1, \dots, n.$

Assumption on data for theorem on Perceptron convergence:

- linearly separability : $\exists\ \rho > 0$ s.t. $\rho := \min_{i \in [n]} \frac{y_i\, w \cdot x_i}{\|w\|}$ for some $w \in \mathbb{R}^d$

- bounded data: $R > 0$ s.t. $\|x_i\| < R \quad \forall\ i.$

Thm: Perceptron algorithm:

$$w_{i+1} = \begin{cases} w_i + \eta\, y_i x_i & \text{if } x_i \in I \text{ (incorrectly classified)} \\ w_i & \text{if } x_i \notin I \text{ (correct)} \end{cases}$$

learning rate

makes at most $\frac{R^2}{\rho^2}$ updates.

<u>Proof</u>: Mohri et al 2018

<u>Interlude</u>:

Classifier returned by the Perceptron algorithm:

$$h(x) = \text{sgn}(w_T \cdot x)$$
$$= \text{sgn}\Big(\Big(\sum_{i \in I} \eta\, y_i x_i\Big) \cdot x\Big)$$

WLOG: $w_0 = 0 \in \mathbb{R}^d$

linear comb. of incorrectly classified points

$$= \text{sgn}\Big(\sum_{i \in I} \eta\, y_i \underbrace{(x_i \cdot x)}_{\text{dot product}}\Big)$$

Dot product algorithms can be lifted to ~~an~~ a higher-dimensional or infinite-dimensional spaces (kernel methods)

$$\underset{\mathbb{R}^d}{x} \to \Phi(x) \in \mathbb{R}^D \qquad D \gg d$$

or $\Phi(x) \in \mathcal{H}$ function (Hilbert space)

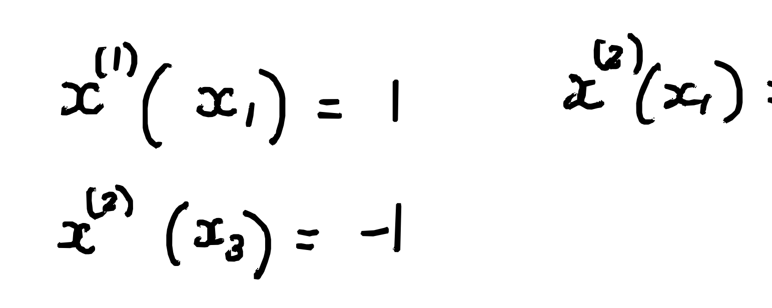
features

if $\langle \Phi(x), \Phi(x') \rangle = \mathcal{K}(x, x')$ (kernel / similarity measure)

then,

$$h(x) = \text{sgn}\Big(\sum_{i \in I} \eta\, y_i \langle \Phi(x_i), \Phi(x) \rangle\Big)$$
$$= \text{sgn}\Big(\sum_{i \in I} \eta\, y_i\, \mathcal{K}(x_i, x)\Big)$$

Example (XOR)



$x^{(1)}(x_1) = 1 \qquad x^{(2)}(x_1) = 1$

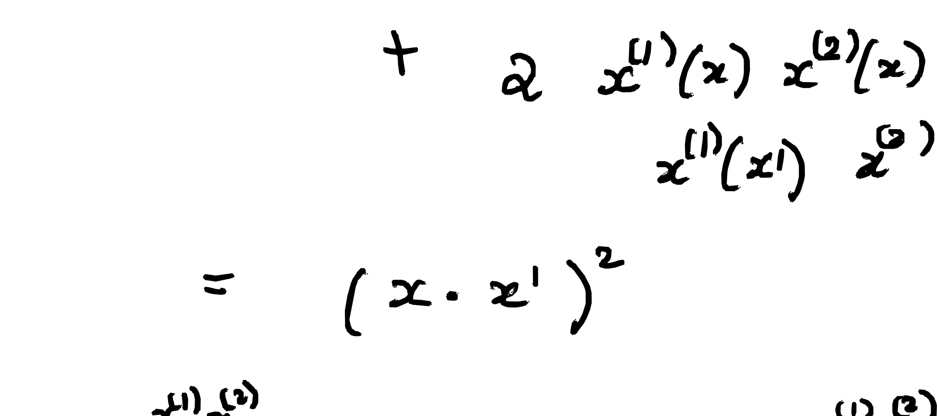
$x^{(2)}(x_3) = -1$

$$\begin{bmatrix} x^{(1)^2}(x) \\ x^{(2)^2}(x) \\ \sqrt{2}\, x^{(1)} x^{(2)}(x) \end{bmatrix} \cdot \begin{bmatrix} x^{(1)^2}(x') \\ x^{(2)^2}(x') \\ \sqrt{2}\, x^{(1)} x^{(2)}(x') \end{bmatrix} =$$

$$\Phi(x) \cdot \Phi(x')$$
$$= \langle \Phi(x), \Phi(x') \rangle$$

$$x^{(1)^2}(x)\, x^{(1)^2}(x') + x^{(2)^2}(x)\, x^{(2)^2}(x') + 2\, x^{(1)}(x)\, x^{(2)}(x)\, x^{(1)}(x')\, x^{(2)}(x'))$$

$$= (x \cdot x')^2$$



<u>Polynomial kernel</u>: $x, x' \in \mathbb{R}^d$

$$\mathcal{K}(x, x') = (x \cdot x' + b)^m$$

Can be written as a dot product of features

$b \neq 0, \quad d = 2, \quad D = 6$ (dim of feature space), $m = 2$

$$(x \cdot x' + b)^2 = \Big(\sum_{i=1}^{d} x^{(i)}(x)\, x^{(i)}(x') + b\Big)^2$$

$$\begin{bmatrix} x^{(1)^2} \\ x^{(2)^2} \\ \sqrt{2}\, x^{(1)} x^{(2)} \\ \sqrt{b}\, x^{(1)} \times \sqrt{2} \\ \sqrt{b}\, x^{(2)} \times \sqrt{2} \\ \sqrt{b} \end{bmatrix}$$

D = 6 dim feature space.

$$\big(x^{(1)}(x)\, \underline{x}^{(1)}(x') + x^{(2)}(x)\, x^{(2)}(x') + \underline{b}\big)^2$$

$$= \sum_{i=1}^{2} x^{(i)^2}(x)\, x^{(i)^2}(x') + b^2 + \sum_{i=1}^{2} 2b\, x^{(i)}(x)\, x^{(i)}(x')$$

$$= \Phi(x) \cdot \Phi(x').$$

To evaluate $\Phi(x) \cdot \Phi(x')$ (which consists of nonlinear functions), we only need $\mathcal{K}(x, x')$

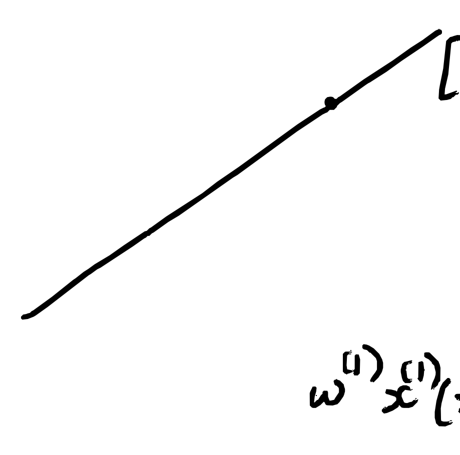
$$\mathcal{K}(x, x') = \Phi(x) \cdot \Phi(x')$$
$$= \begin{bmatrix} x^{(1)^2}(x) \\ \vdots \\ \sqrt{b} \end{bmatrix} \cdot \begin{bmatrix} x^{(1)^2}(x') \\ \vdots \\ \sqrt{b} \end{bmatrix}$$
$$= \underline{(x \cdot x' + b)^m}$$

kernel Perceptron : $h(x) = \text{sgn}\big(\sum \eta\, y_i\, \mathcal{K}(x_i, x)\big)$

Kernel evaluation $\Longleftrightarrow$ inner product on feature space

- doesnot require explicit knowledge of features

- computationally expensive
- have to know features



$$[w \;\; b] \begin{bmatrix} x^{(1)}(x) \\ x^{(2)}(x) \\ 1 \end{bmatrix} = 0$$

$$w^{(1)} x^{(1)}(x) + w^{(2)} x^{(2)}(x) + b = 0$$

$$\underline{x^{(2)}(x)} = -\frac{w^{(1)}}{w^{(2)}}\, x^{(1)}(x) - b$$