ML EXAMS 20/01/2020

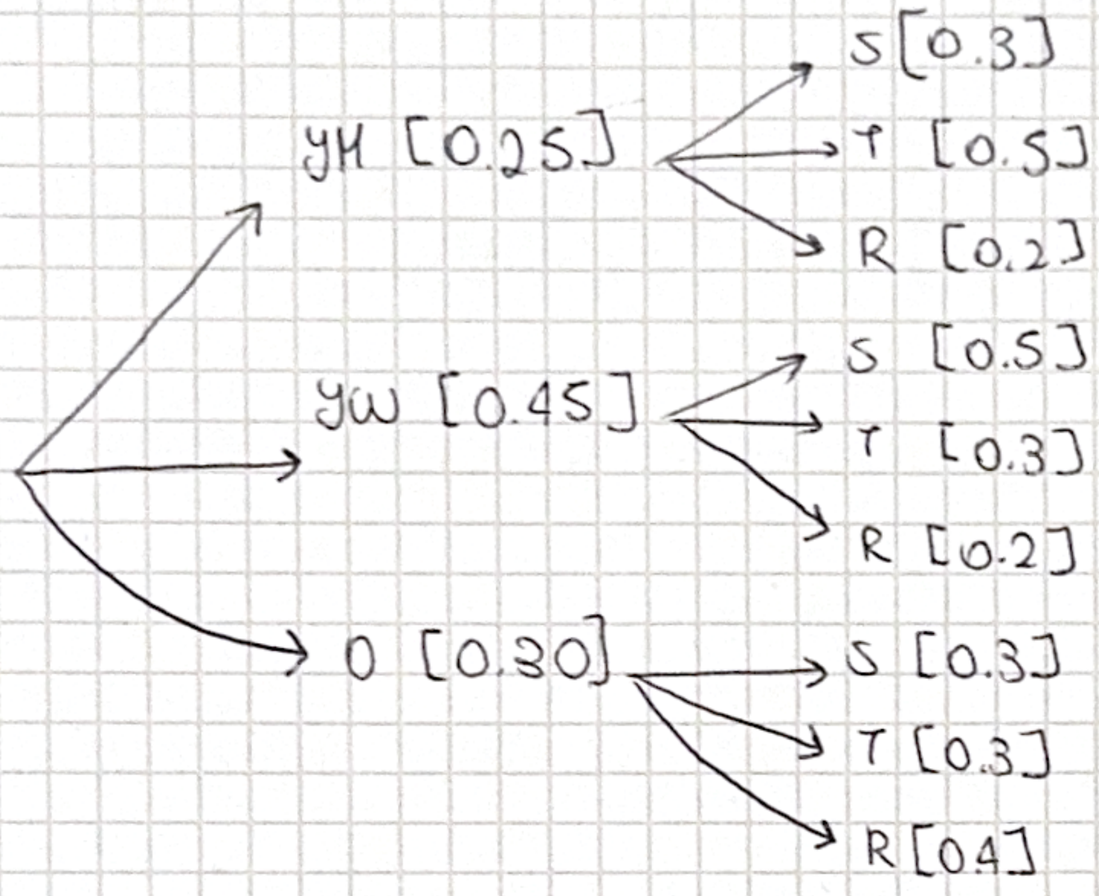
EX. 1

$P(yM) = 0.25 \quad P(yW) = 0.45 \quad P(O) = 0.30$

$P(yM|S) = 0.3 \quad P(yM|T) = 0.5 \quad P(yM|R) = 0.2$

$P(yW|S) = 0.5 \quad P(yW|T) = 0.3 \quad P(yW|R) = 0.2$

$P(O|S) = 0.3 \quad P(O|T) = 0.3 \quad P(O|R) = 0.4$



$v^* = \underset{c \in C}{\operatorname{argmax}} \{P(T|c)P(c)\} = \{(\underbrace{0.25 \cdot 0.5}_{0.125}), (0.3 \cdot 0.45), (0.3 \cdot 0.3)\} \underset{\text{argmax}}{} =$

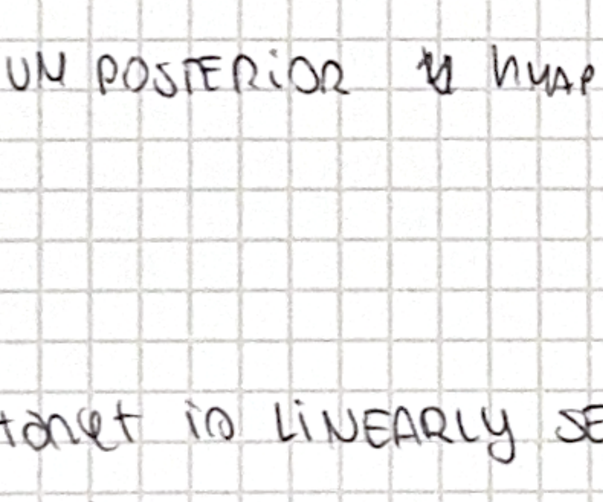
$= \operatorname{argmax} \{0.125, 0.09, 0.135\} = yW$

2 in MAXIMUM POSTERIOR $h_{MAP} = \underset{h \in H}{\operatorname{argmax}} (P(D|h)P(h))$

EX. 2

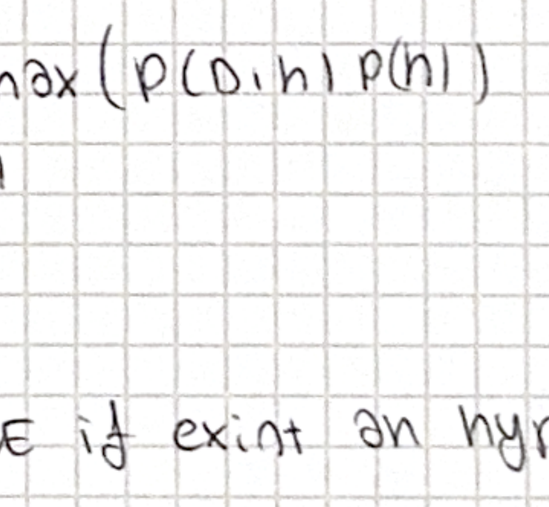
1. a dataset is LINEARLY SEPARABLE if exist an hyperplane (surface) that splits our instance into two regions such that different classified are separated

2.



SEPARABLE

3.



NOT SEPARABLE

4.

the first dataset is linearly separable and SVM aims at maximum margin providing a better accuracy

given a dataset D and an hyperplane h the margin is computed

$$\min_{n=1,\dots,N} \frac{1}{\|w\|} \cdot |y(x_n)| = \dots = \frac{1}{\|w\|} \min_{n=1,\dots,N} [t_n (w^T x_n + w_0]$$

no given a dataset D and an hyperplane $h^* = w^{*T} x + w^*$ maximum margin is computed as:

$$w^*, w_0^* = \arg\max_{w, w_0} \frac{1}{\|w\|} \min_{n=1,\dots,N} [t_n (w^T x_n + w_0)] .$$

a more stable version is obtain by averaging all support vector

$$w_0^* = \frac{1}{|SV|} \sum_{x_k \in SV} \left(t_k - \sum_{x_j \in S} a_j^* t_j x_k^T x_j \right)$$

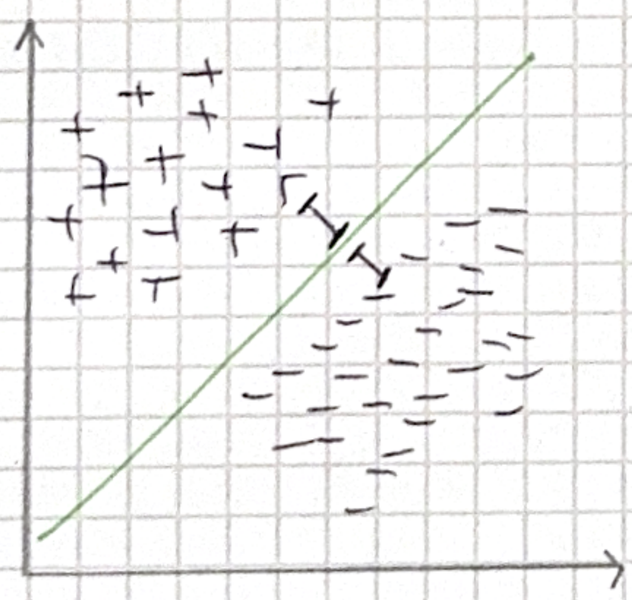
where a_j^* are lagrange multipliers used to solve maximization problem write before

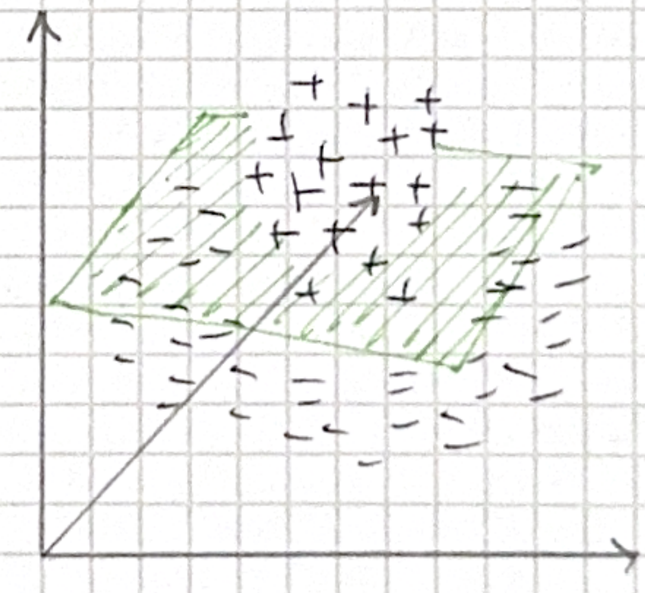
$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} a_n a_m t_n t_m x_n^T x_m$$

and:

$$SV = \{ x_k \in D \mid t_k y(x_k) = 1 \}$$

for the second dataset because is not linear separable we need to use some basis function like polinomial so result will be $y(x) = w^T \phi(x) + w_0$





EX. 3

1.

in PCA we want maximize the data variance after the projection to some direction u_1 the projected points are $x_n^T u_1$

variance of projected points

$$\frac{1}{N}\sum_{n=1}^{N}[u_1^T x_n - u_1^T \bar{x}]^2 = u_1^T S u_1$$

where S (dxd is covariance matrix define:

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T = \frac{1}{2}x^T x$$

we want maximite $\max_{u_1} u_1^T S u_1$ with a lagrange multiplier

$$\max_{u_1} u_1^T S u_1 + \lambda(1 - u_1^T u_1)$$

solution $\lambda_1 = u_1^T S u_1$

variance is maximum when ~~compressed~~ u_1 correspond to a largest eigenvector called PRINCIPAL COMPONENT

2. yes because the minimni dimensionality of data are 3 so we can write images $\mathbb{R}^{w \times h \times c} \to \mathbb{R}^3$ in a point represent in 3D dimension

EX. 4

1. $W_{in} = 32 \quad h_{in} = 32 \quad d_{in} = 3$

~~$W_{out1} = \frac{32 - 5 + 2 \cdot 2}{1} + 1 = 32$~~

~~$H_{out1} = \frac{32 - 5 + 2 \cdot 2}{1} + 1 = 32$~~

~~** parametri~~

** parametri $1 = 5 \times 5 \times 3 \times 16 + 16 = 1216$

** parametri $2 = 3 \times 3 \times 3 \times 32 + 32 = 896$

** parametri $3 = 5 \times 3 \times 64 + 64 = 2944$

** $f_{c1} = 200 \cdot 10 = 2000$

2. I will use cross-entropy loss

$$J(\theta) = E_{x,c \sim D}[-\ln(P(t|x,\theta))]$$

assuming additive noise

$$J(\theta) = \mathop{E}_{x,t\sim D}\left[\tfrac{1}{2}\|t - f(x,\theta)\|^2\right]$$

EX. 5

1.

the model can be defined as $y(x,w) = \sum_{n=1}^{N} w_n \phi_n(x)$

we know that target value are $t = y(x,w) + \varepsilon$ with some noise ε

if noise is gaussian $p(\varepsilon|\beta) = \mathcal{N}(\varepsilon|0,\beta^{-1})$

$p(t|x_1,\dots,x_n,\beta) = \mathcal{N}(t|y(x,w),\beta^{-1})$

now if we assume iid hypothesis $p(t_1,\dots t_n|x_1,\dots,x_n,w,\beta) =$

$= \prod_{n=1}^{N} \mathcal{N}(t_n|w^T\phi(x_n)\beta^{-1}) = \underbrace{-\beta\,\frac{1}{2}\sum_{n=1}^{N}(t_n - w^T\phi(x_n))^2}_{E_D(w)} - \frac{N}{2}\ln(2\pi/\beta^{-1})$

2.

we consider error function

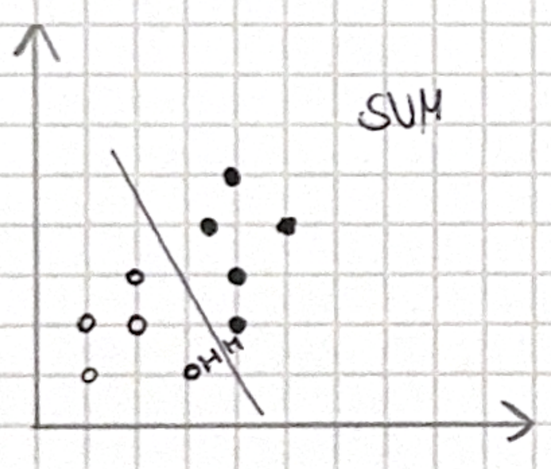
$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T x_n)^2$$

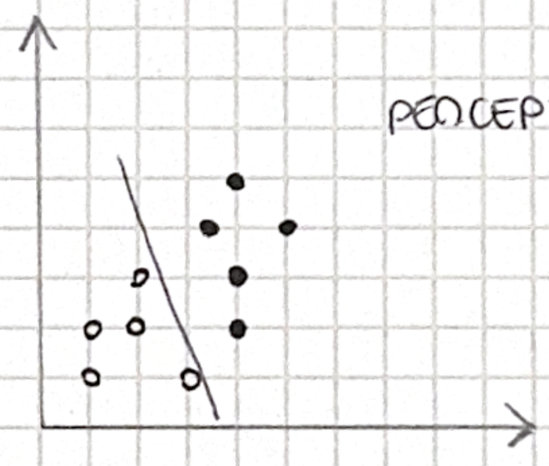
we can use sequential algorithm to find $w^* = \arg\min E_D(w)$

$$\hat{w} \leftarrow \hat{w} + \eta[t_n - w^T\phi(x_n)]\phi(x_n)$$

EX. 6

1.





2

the main different is that SVM ~~tries to maximize~~ aims to maximum with the better accuracy while perceptron no.

Perceptron stop iterations when finds a possible separation surface simply summing the vector related to the separation surface with the feature vector that misclassified

3. I prefer SVM because with linear separable dataset due the fact aims to maximum with better accuracy obtain better result while with perceptron we need to find the η correct η for have a convergence.