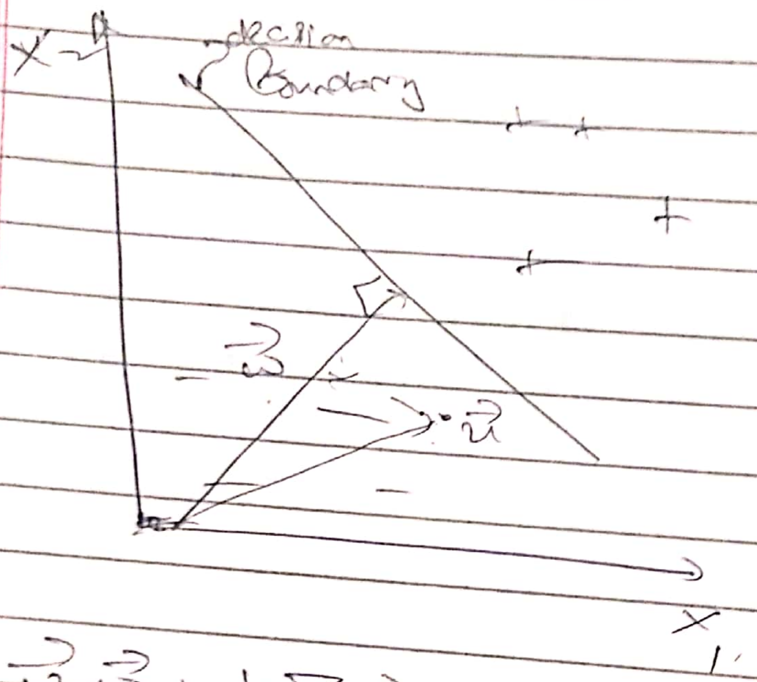
Machine learnig

Sklearn, Opencl, Pandas

~~Regression~~.

SVM

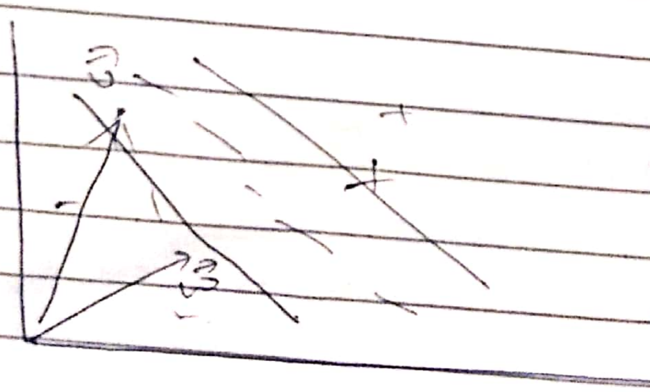


$\vec{u}\cdot\vec{w}+b>0 \rightarrow +$

$\vec{u}\cdot\vec{w}+b<0 \rightarrow -$

$\vec{u}\cdot\vec{w}+b=0 \rightarrow$ decision boundary

~~wide street~~ approach.



$\vec{w}\cdot\vec{u}\geq c \rightarrow$ goes to other side then +ve

$\vec{w}\cdot\vec{u}+b\geq 0$

$$\boxed{\vec{w}\cdot\vec{u}+b\geq 0}$$

Decision Rule

classmate

Date

Page

Assume

$\vec{w}.\vec{x}_+ + b \geq 1 \rightarrow ①$

$\vec{w}.\vec{x}_- + b \leq -1 \rightarrow ②$

$y_i = +1$ for +ve Sample
$\quad -1$ for -ve Samples

Multiply y_i to ① and ②

$y_i(\vec{w}.\vec{x}_+ + b) \geq 1 \rightarrow ③$

$y_i(\vec{w}.\vec{x}_- + b) \geq 1 \rightarrow ④$ (as -1 is Multiplied Change Sine)

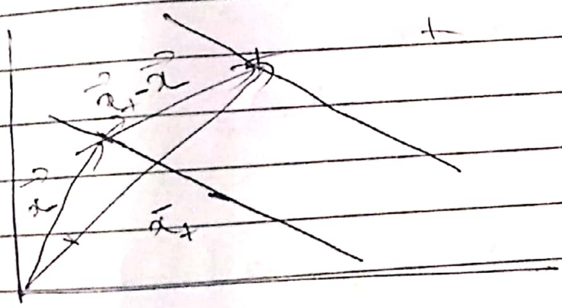
③ = ④

So

$y_i(\vec{x}_i.\vec{w} + b) - 1 \geq 0$

boundary

$\boxed{y_i(\vec{x}_i\vec{w} + b) - 1 = 0} \rightarrow ⑤$



width of street $= (x_+ - x_-) . \frac{\vec{w}}{||w||} \rightarrow ⑥$

$\hookrightarrow$ unit vector $\perp^r$ to line

from ⑤

for +ve Sample $\vec{x}_+ . \vec{w} = 1 - b$

for -ve Sample $-\vec{x}_- . \vec{w} = 1 + b$

So ⑥

$$\frac{(1 - \cancel{b} + (1 + \cancel{b}))}{||w||} = \frac{2}{||w||}$$

Scanned with CamScanner

To increase width maximise $\frac{2}{\|w\|}$

So minimize $\|w\| \Rightarrow$ So minimise $\frac{1}{2}\|w\|^2$

Lagrange Multiplier Note

used for finding extrema given a constraint

ex:

$f = x_1 x_2 x_3 \qquad g = x_1 + x_2 + x_3 - 1 = 0$

↓

Maximize f and let it follow g

$L = f + \lambda g$

$L = x_1 x_2 x_3 + \lambda(x_1 + x_2 + x_3 - 1)$

$\frac{\partial L}{\partial x_1} = x_2 x_3 + \lambda = 0 \qquad x_2 x_3 = -\lambda$

$\frac{\partial L}{\partial x_2} = x_1 x_3 + \lambda = 0 \qquad x_1 x_3 = -\lambda$

$\frac{\partial L}{\partial x_3} = x_2 x_1 + \lambda = 0 \qquad x_2 x_1 = -\lambda$

$x_1 = x_2 = x_3$

$\frac{\partial L}{\partial \lambda} = x_1 + x_2 + x_3 - 1 = 0$

$x_1 = x_2 = x_3 = 1/3$

$(1/3, 1/3, 1/3)$

To check wether Maxima or Minima

$$\begin{vmatrix} L_{11}-k & L_{12} & L_{13} & g_1 \\ L_{21} & L_{22}-k & L_{23} & g_2 \\ L_{31} & L_{32} & L_{33}-k & g_3 \\ g_1 & g_2 & g_3 & 0 \end{vmatrix} = 0$$

Date ______
Page ______

if k +ve $\rightarrow$ minima

k -ve

$\geq \rightarrow$ Maxima

So for $\frac{1}{2}\|w\|^2$

$$L = \frac{1}{2}\|w\|^2 - \sum \alpha_i [y_i(\vec{w}\cdot\vec{x}_i + b) - 1] \longrightarrow (7)$$

$$\frac{\partial L}{\partial w} = \|w\| - \sum \alpha_i y_i \vec{x}_i = 0$$

$$\boxed{\|w\| = \sum_i \alpha_i y_i \vec{x}_i} \longrightarrow (8)$$

$$\frac{\partial L}{\partial b} = -\sum \alpha_i y_i = 0$$

$$\boxed{\sum \alpha_i y_i = 0} \longrightarrow (9)$$

replace (8) and (9) in (7)

$$L = \frac{1}{2}\left(\sum \alpha_i y_i \vec{x}_i\right)\cdot\left(\sum \alpha_i y_i \vec{x}_i\right) - \sum \alpha_i y_i x_i \left(\sum \alpha_j x_j y_j\right) - \sum \alpha_i y_i b + \sum \alpha_i$$

$$L = \frac{1}{2}\left(\sum \alpha_i y_i x_i\right)\cdot\left(\sum \alpha_i y_i \vec{x}_i\right) - \left(\sum \alpha_i y_i x_i\right)\cdot\left(\sum \alpha_j x_j y_j\right) + \sum \alpha_i$$

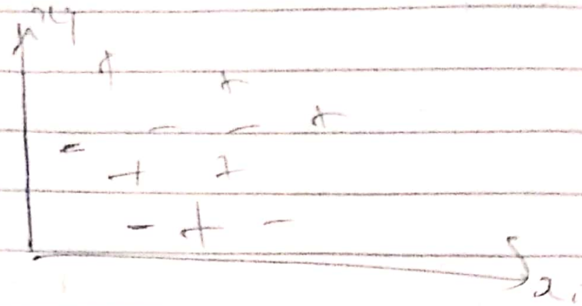
$$\boxed{L = \sum \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j} \longrightarrow (10)$$

So Decision Rule

$$\vec{w}\cdot\vec{u} + b \geq 0$$

$$\sum \alpha_i y_i x_i \cdot \vec{u} + b \geq 0$$

Now if the Samples are linearly inseparable like



then transform it to another space which is more favorable call it $\phi(a)$

$\phi(x_i) \cdot \phi(x_j)$ to max.

$k(x_i, x_j) = \phi(\vec{x_i}) \cdot \phi(\vec{x_j})$

So all we need is k (kernel function)

we need a function which provides . product in other space not the space itself.

linear kernel $(\vec{U} \cdot \vec{V} + 1)^n$ → dot Product in a space

radial basis kernel $e^{-\frac{\|x_1 - x_2\|}{\sigma}}$ → another kernel function

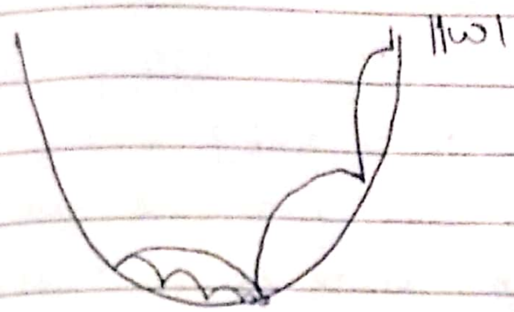
So

$y_i(\vec{x_i} \cdot \vec{w} + b) - 1 = 0$

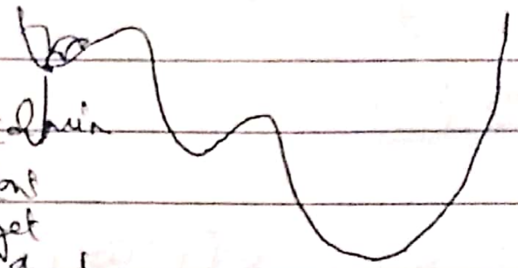
$\text{Sign}(\vec{X_i} \cdot \vec{w} + b)$ → will be decision.

So Minimize $\|\vec{w}\|$ and maximize b

so find $\|\vec{w}\|$ min



Need to find global Minima



optimization of SVM necessary (convex optimisation problem)

(To solve optimisation we can use CVXOPT, LIBSVM)