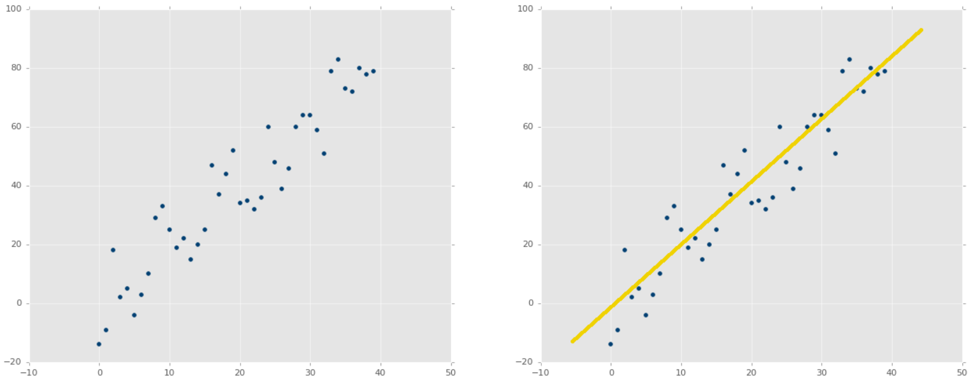
Regression

03 June 2024 18:57

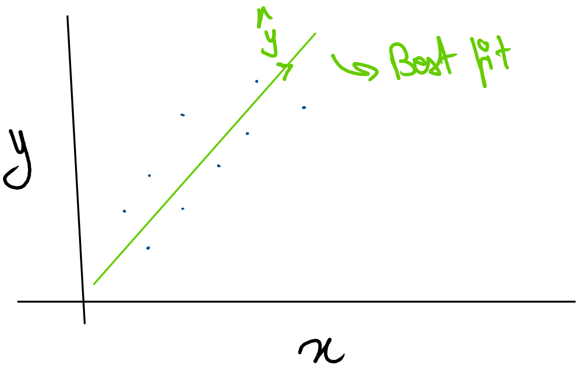
Simple Regression is used to find the best fit line of a dataset.



y-intercept

$y = mx + b$

Slope



Slope(m) : $m = \dfrac{\bar{x} \cdot \bar{y} - \overline{xy}}{(\bar{x})^2 - (\overline{x^2})}$

y-intercept(b) : $b = \bar{y} - m\bar{x}$

$\{ \because y = mx + c \}$ $\therefore$ $\hat{y} = \dfrac{\bar{x} \cdot \bar{y} - \overline{xy}}{(\bar{x})^2 - (\overline{x^2})} \times x + (\bar{y}) - \left(\dfrac{\bar{x} \cdot \bar{y} - \overline{xy}}{(\bar{x})^2 - (\overline{x^2})} (\bar{x}) \right)$

Equation of Regression Line($\hat{y}$)

→ R - Squared

The distance between Regression line Values ($\hat{y}$) & data's y values is the error, which we then sq.

Lines Squared error is either a mean or Sum of this Values.

$$r^2 = 1 - \frac{\text{Squared Error}(\hat{y})}{\text{Squared Error}(\bar{y})}$$

regression line

mean of y data points

Coefficient of determiment

Q) Why are we squaring errors? Why not just adding them up?

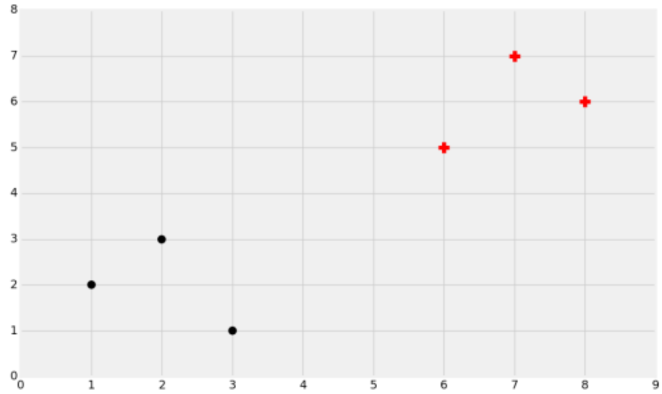
Ans) First, we want a way to normalize the error as a distance, so the error might be -5, but, when squared, that's a positive number. Another reason is to further punish for outliers. The "further" off something is, the more the it affects the error. This is simply the standard people use.

You could use the power of 4, 6, 8, or whatever. You could also just use the absolute value of the errors. If you have a unique challenge, maybe where some extreme outliers do exist, but you don't care to map them, you could consider doing something like an absolute value. If you care a lot about outliers, you could use much higher exponents. We'll stick with squared, as that is the standard almost everyone uses.

KNN

10 June 2024 20:24

- It is a type of classification algo.
- Type of Supervised Learning.
- Datasets are labeled
- Lazy Learner, no learning phase



- Algo:-

Step-1: Select the number K of the neighbours
Step-2: Calculate the Euclidean distance of **K number of neighbors**
Step-3: Take the K nearest neighbours as per the calculated Euclidean distance. → $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$
Step-4: Among these k neighbours, count the number of the data points in each category.
Step-5: Assign the new data points to that category for which the number of the neighbour is maximum.
Step-6: Our model is ready.

→ Working:-

1. Pass train_data & single data point from test_set.
2. Calculate Euclidean distance for data point to all the data present in train_data.
3. Save the distance and the group

for data point to train_data.

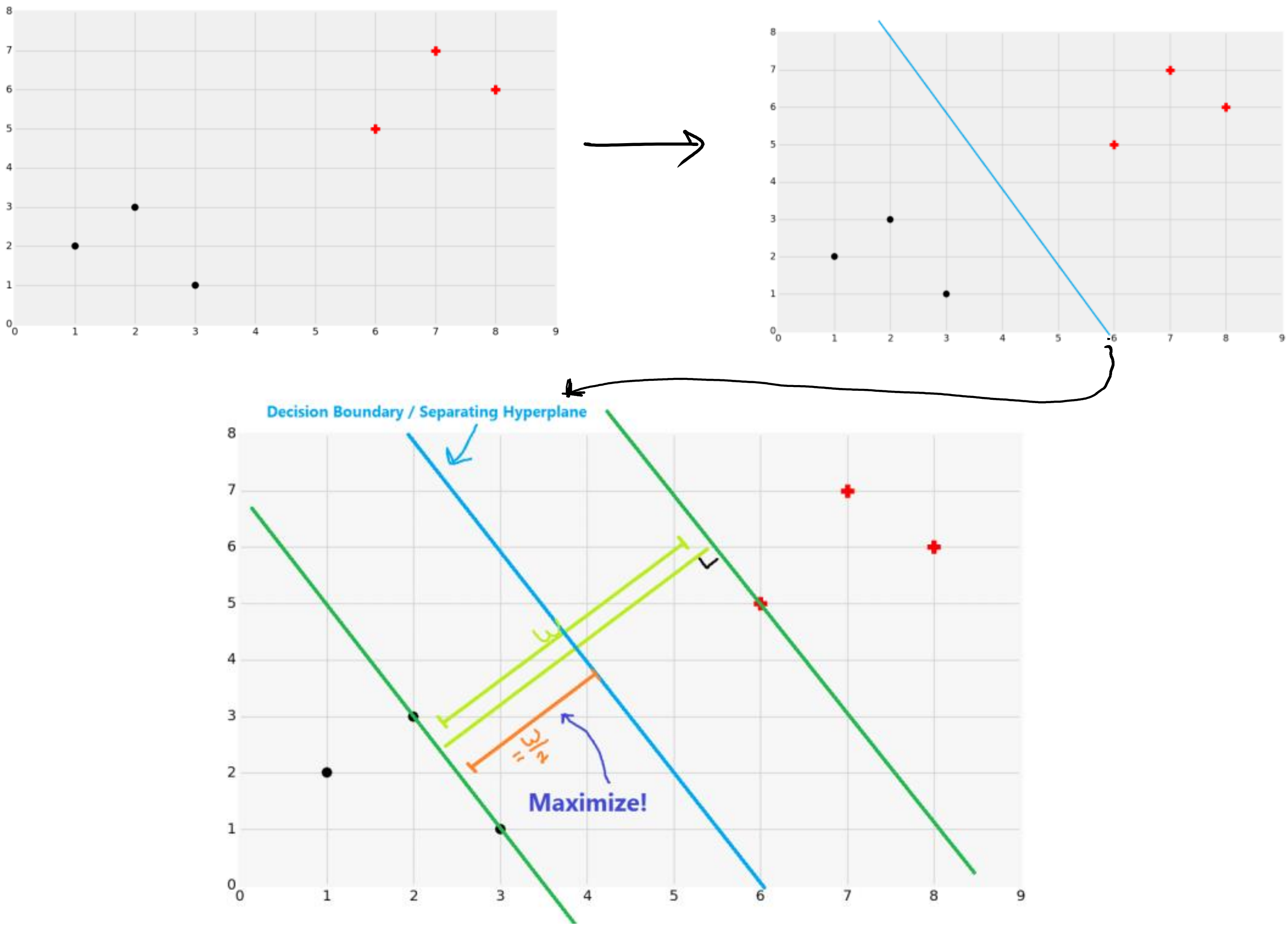
(4) Sort and select 'K' from the saved point.

(5) From the 'k' selected groups, vote for most common classifier.

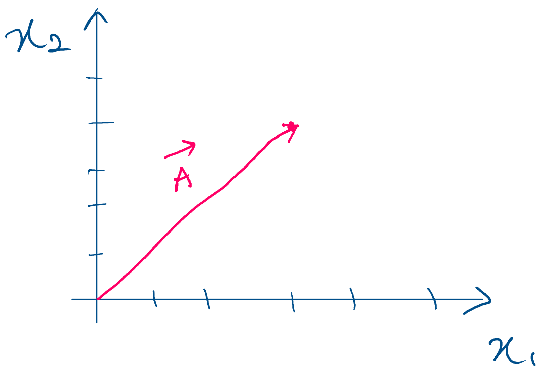
SVM

https://see.stanford.edu/materials/aimlcs229/cs229-notes3.pdf https://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf

14 June 2024 20:55



→ Vector Basics



$\vec{A} = [3,4]$

$|A| = 5$

$\vec{A} = [1,3]$

$\vec{B} = [4,2]$

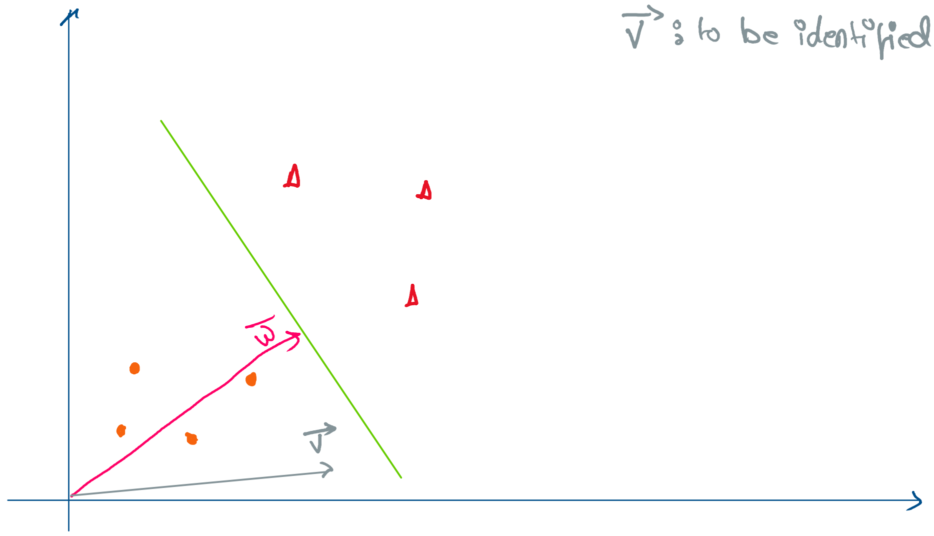
$\vec{A} \cdot \vec{B} = [(1 \times 4) + (3 \times 2)]$

$= 4 + 6$

$= 10$

→ SVM Working :-

$\vec{v}$: to be identified



$(\vec{v} \cdot \vec{w}) + b \begin{cases} \geq 0 & \Delta \\ \leq 0 & \bullet \end{cases}$

if $= 0$ then $\vec{v}$ lies on the desision boundry

$\Rightarrow \quad \bar{X}_{\bullet SV} \cdot \bar{w} + b = -1 \qquad \bar{X}_{\Delta SV} \cdot \bar{w} + b = 1$

$y_i \begin{cases} \Delta = 1 \\ \bullet = -1 \end{cases}$

(class)

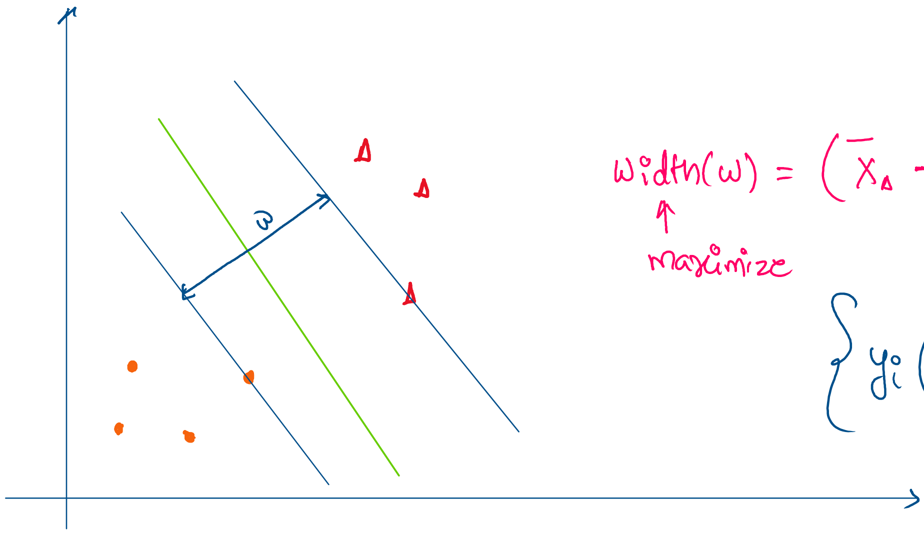
$\Rightarrow$ Δ class $\quad \bar{x}_i \cdot \bar{w} + b = 1 \Rightarrow y_i(x_i \cdot \bar{w} + b = 1) = (1) 1$

$\Rightarrow$ △ class $y_i = 1$ $\quad \bar{x}_i \cdot \bar{w} + b = 1 \Rightarrow y_i(x_i \cdot \bar{w} + b = 1) = (1)1$

• class $y_i = -1$ $\quad \bar{x}_i \cdot \bar{w} + b = -1 \Rightarrow y_i(\bar{x}_i \cdot \bar{w} + b = -1) = (-1)(-1)$

△ class $y_i = 1$ $\Rightarrow y_i(x_i \cdot \bar{w} + b) - 1 = \cancel{1-1} \to 0$

• class $y_i = -1$ $\Rightarrow y_i(\bar{x}_i \cdot \bar{w} + b) - 1 = \cancel{1-1} \to 0$



$$\text{width}(w) = (\bar{x}_\triangle - \bar{x}_\bullet) \cdot \frac{\bar{w}}{|w|}$$

↑ maximize

$$\{ y_i(\bar{x}_i \cdot \bar{w} + b) - 1 = 0 \}$$

$$\Rightarrow \left((1-b) - (1+b)\right) \cdot \frac{\bar{w}}{|w|}$$

☆

$$\boxed{\text{width} = \frac{(2) \cdot \bar{w}}{|w|}}$$

width (maximize) $= \dfrac{2}{|w|}$ $\leftarrow$ minimize

* * we want to minimize $\frac{2}{|w|}$ then we also want to minimize $|w|$.
* Lets says we want to minimize $\frac{1}{2}|w|^2$ as we alerdy want to minimize $|w|$

$$\frac{1}{2}|w|^2 \quad y_i(x_i \cdot w + b) - 1$$

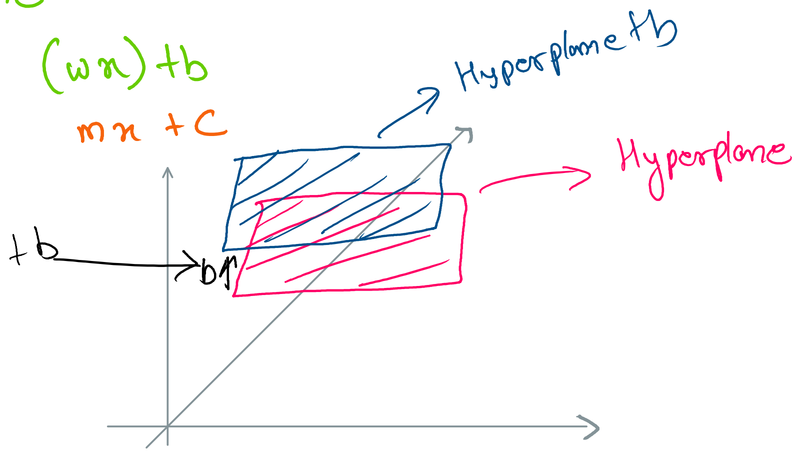
Lagrangian formulation:

$$L(\overset{\text{min}}{w}, \overset{\text{max}}{b}) = \frac{1}{2}|w|^2 - \sum \alpha_i \left[y_i(\bar{x}_i \, \bar{w} + b) - 1 \right]$$

• Hyperplane

$(wx) + b$

$mx + c$



=) on differenciating wrt w & also differenciating wrt. b.

$$\frac{dL}{dw} = \bar{w} - \sum \alpha_i \, y_i \, x_i = 0 \Rightarrow \bar{w} = \sum \alpha_i y_i x_i$$

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

$$\boxed{\underset{\text{max}}{L} = \sum_i \alpha_i - \frac{1}{2}\sum_{ij} \underbrace{\alpha_i \alpha_j}_{\text{quadratic}}\, y_i y_j \cdot (\bar{x}_i \cdot \bar{x}_j)}$$

→ Concept :-

$(X \cdot w + b)$ ⇐ Hyperplane Equation

$= 1$ +class $\qquad = -1$ -class

$\Rightarrow \text{Sign}(x_i \cdot w + b)$

< 0 +class

0 Decision Boundary

> 0 -class

* The optimization objective is to minimize $|w|$ & maximize b. $\{ |w| \downarrow \ \& \ b \uparrow \}$

Known feature

$y_i (x_i \cdot w + b)$

$y_i(x_i \cdot w + b)$

Know...

Class

Unknown

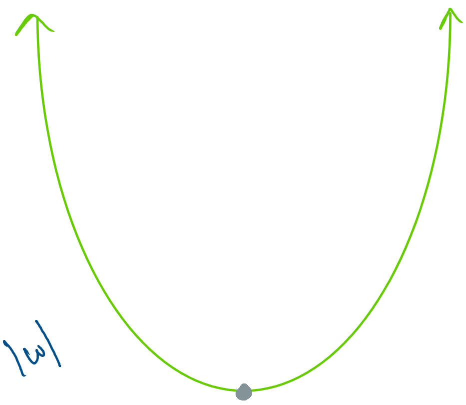
$$\boxed{L_{max} = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j \, y_i y_j \cdot (\bar{x}_i \cdot \bar{x}_j)}$$

quadratic

SVM is a optimization problem, |w| is a Convex shape.

{Convex}

$\vec{w}_{1\times 2}$



|w|

Finding minima is the target.

Since we are trying to find minima of |w| we need to check values in all the 4 quadrants for $\vec{w}$.

Eg: $\vec{w} = [10, 10]$

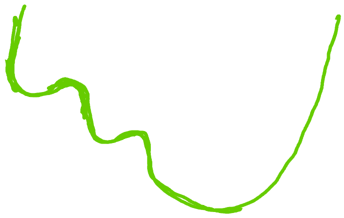
$y_i(x_i w + b) \geq 1$

$mag = \{ |w| : [\bar{w}, b] \}$

(finding lowest magnitude is target.

* local minima problem might occur

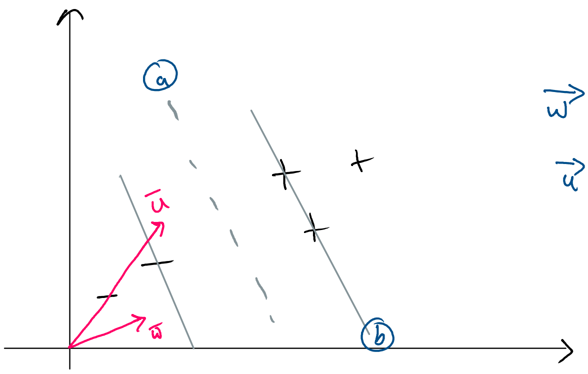
* local minima Problem might occur



is target.

SVM - MIT Notes

25 June 2024 20:02



$\vec{w}$ is $\perp$ to (a)

$\vec{u}$ Points an unknown value, lies on - or + side

$\bar{w} \cdot \bar{u} \geq c$ {c, Some Constant 'c'}

⇒ dot product has taken projection on $\bar{w}$, the bigger the projection is the further the projection lie on $\bar{w}$.

$$\boxed{\bar{w} \cdot \bar{u} + b \geq 0 \quad \text{Then } +}$$

Decision Rule

$\bar{w} \cdot \bar{x}_+ + b \geq 1$ {for pt to lie out the the decision boundry (b)}

$\bar{w} \cdot \bar{x}_- + b \leq 1$

⇒ let y_i be such that $y_i = +1$ for + Samples

$y_i = -1$ for − Samples

(+) $y_i(\bar{w} \cdot \bar{x}_i + b) \geq y_i(1)$ ⇒ $y_i(\bar{w} \cdot \bar{x}_i + b) \geq 1$

(−) $y_i(\bar{w} \cdot \bar{x}_i + b) \leq y_i(-1)$ ⇒ $y_i(\bar{w} \cdot \bar{x}_i + b) \geq 1$

(−) $(y_i)(-1)$

(-)

(- 1)

(+)

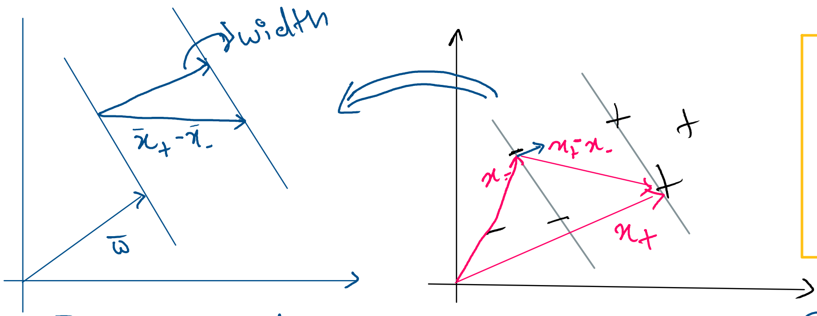
(-1) (-1)

(+1)

$\geq$ (1)

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

$\boxed{y_i(\bar{x}_i \cdot \bar{w} + b) = 0}$; for x_i in gutter



$\bar{w}$ is a normal

$\frac{\bar{w}}{|w|}$ is an unit vector

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

{ using $y_i(\bar{x}_i \cdot \bar{w} + b) = 0$ }

$$\Rightarrow \; = \left(\bar{x}_+ - \bar{x}_-\right) \cdot \frac{\bar{w}}{|w|}$$

$$(1 - b - 1 + b) \cdot \frac{1}{|w|}$$

$$\Rightarrow \boxed{\text{width} = \frac{2}{|w|}} \quad \text{— } 3$$

To get the max dist b/w descision boundry we need to maximize $\frac{2}{|w|}$

$\hookrightarrow$ maximize $\frac{1}{\;}$ $\rightarrow$ minimize $|w|$

$\hookrightarrow$ maximize $\frac{1}{|w|} \rightarrow$ minimize $|w|$

$\hookrightarrow$ minimize $\frac{1}{2}|w|^2$

for mathematical Convinience.

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

Lagrange Multipliers

$$\frac{\partial L}{\partial \bar{w}} = \bar{w} - \sum \alpha_i y_i x_i = 0 \Rightarrow \boxed{\bar{w} = \sum \alpha_i y_i x_i}$$

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

$$L = \frac{1}{2}\left(\sum \alpha_i y_i \bar{x}_i\right)\left(\sum \alpha_j y_j \bar{x}_j\right) - \sum \alpha_i y_i x_i\left(\sum \alpha_j y_j x_j\right) - \underbrace{\sum \alpha_i y_i b}_{0} + \sum \alpha_i$$

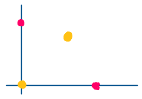
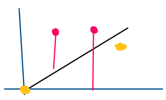
$$\boxed{L = \sum \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i \cdot x_j}$$

$$\boxed{\sum \alpha_i y_i \boxed{\bar{x}_i \cdot \bar{u}} + b \geq 0 \quad \text{Then} +}$$

Descision Rule only depends on the dot product of Sample Vectors $(\bar{x}_i)$ & unknown vector $(\bar{u})$

Note: this a Convex Space, so we can't get stuck to a local maxima.

For Non-linear Seprable Data:-

we can seprate Non linear data we change space.

$\phi(\bar{x}_i) \cdot \phi(\bar{x}_j)$ To maximize

$\phi(\bar{x}_i) \cdot \phi(\bar{u})$

$$\boxed{K(x_i, x_j) = \phi(\bar{x}_i) \cdot \phi(\bar{x}_j)}$$

Kernal function

① Linear Kernal - $(\bar{u} \cdot \bar{v} + 1)^n$

② Radial Basis - $e^{-\frac{|\bar{x}_i - \bar{x}_j|}{\sigma}}$