AKSHAY
202051018

⇨ Margin: Distance of a data point from the decision boundary.

Minimum margin: Distance of the decision boundary to the single and closest data point in the training data.

☆ Our concern is ~~a~~ with minimum margin. Sometimes, people use the word just "margin" instead of "minimum margin". We have to find that particular decision boundary whose minimum margin is maximum from both clusters of data. Such boundary lies in middle.

Those nearest data points are called Support Vectors.

In nutshell,
- Find all decision boundary.
- Find minimum margin corresponding to all boundaries.
- Most optimal boundary will be one with maximum value of minimum margin.

OK! Now, we see how to calculate this margin.

⇨ But before that ...:-

Equation of a hyperplane:

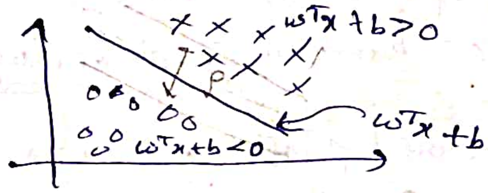
$$\omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \cdots + \omega_n x_n + b = 0$$

$$\Rightarrow \boxed{\vec{\omega}^T \vec{x} = -b}$$

$\vec{\omega}$: weight vector
$\vec{x}$: features vector
n = number of features
b: bias

$\vec{\omega}^T \vec{x} + b \longrightarrow$ decision hyperplane

$\omega^T x + b < 0$ for negative class
$\omega^T x + b > 0$ for +ve class



⇨ Margin 1.0 :-

Taking $\omega^T x + b$ as our discriminant score for margin.

For any point, higher the value of $\omega^T x + b$, farther is the point from decision boundary.

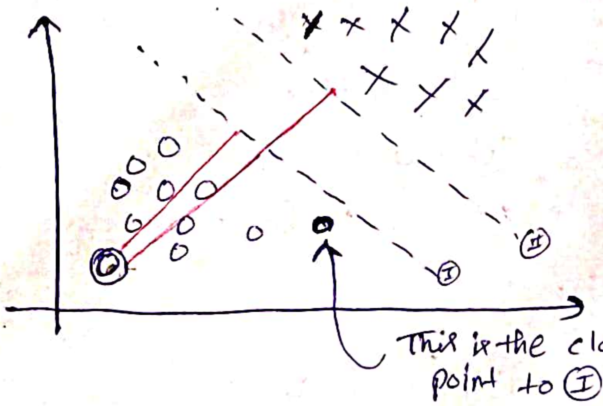
✱ high +ve value ⇒ point is very far and above the decision boundary

✱ high -ve value ⇒ point is very far and ~~above~~ below the decision boundary

$$\text{minimum margin} = \min_{y}(w^T x + b)$$

Drawback of margin 1.0 :-

Minimum margin is always coming from one single point which is in negative class and deepest.



However, minimum margin is defined for the closest point.

⇨ **Bringing constraints for optimal boundary / Need of change :-**

Even closest points to decision boundary should be as away as possible for confident classification.

⇒ For positive class:

$w^T x + b >> 0$ (discriminant scores)

⇒ For negative class:

$w^T x + b << 0$

From now on,

$w^T x + b \geq 1$ for +ve class

and, $w^T x + b \leq -1$ for -ve class

(constraints for decision boundary)

⇨ **Margin 1.1 :-**

We should take magnitude of distance even for negative class.

Taking discriminant score as $|w^T x + b|$

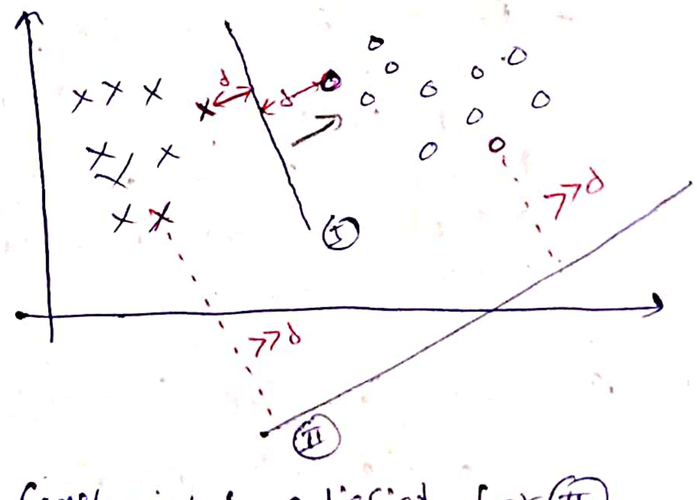
and constraint will become:-
$\boxed{|w^T x + b| \geq 1 \;\; \cancel{\text{For}}}$

$\boxed{\text{Min. margin} = \min(|w^T x + b|)}$

Now, the min margin distance will ~~to~~ not be calculated w.r.t deepest point in -ve class but w.r.t clo the closest point.

Drawback of margin 1.1 :-

If we go by this definition of margin the our algorithm can learn some poor maximum margin classifier.



Constraint is satisfied for (II) as well. Our algorithm will declare (II) as more better decision boundary than (I) because min. margin is larger for it.

Here, Negative class is fully misclassified.

⇒ Margin 2.0 aka Functional Margin :-

$$\boxed{\text{margin} = y\,(w^T x + b)}$$

y : output class label
$y = \{-1, +1\}$

> **Constraints:**
> For all points, $y\,(w^T x + b) \geq 1$
> (i.e. margin ≥ 1)

$$\boxed{\text{Minimum margin},\ F = \min\left[y\,(w^T x + b)\right]}$$

Peculiarity of this constraint:-

If a negative class point is above the decision boundary:
margin $= (-ve)(+ve) < 0$
⇒ Such decision boundary will be rejected

If a positive class point is below the decision boundary:
margin $= (+1)(-ve) < 0$
⇒ Such decision boundary will be rejected.

~~For~~ **Following things are successfully achieved :-**

No missclassification

Minimum margin is calculated from the closest point only.
* ∵ margin value is +ve always.

⇒ ~~Optimat~~ Optimization problem so far :-

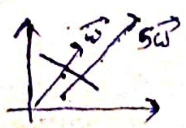
Find decision boundary (i.e. $\vec{w}$, b) for which minimum margin is maximum

or

> $$\max_{\vec{w},\, b} \left[\min\left(y(w^T x + b)\right)\right]$$
> under the constraint:
> $y\,(w^T x + b) \geq 1\ \forall$ points

⇒ Drawback with Functional margin :-

We can maximise the functional margin simply by scaling $\vec{w}$ (normal vector) and b (~~distance from origin or intercept or~~ bias). But by doing so our decision boundary is not changing.



For same decision boundary we can have many values of F. So, functional margin is not a good metric.

Margin 3.0 aka Geometric margin :-

We constraint our margin with unit length normal vector.

$$\boxed{\text{margin} = \frac{y(\vec{\omega}^T\vec{x}+b)}{\|\vec{\omega}\|}}$$

$$\underset{(G)}{\text{Min. margin}} = \frac{\min(y(\omega^T x+b))}{\|\vec{\omega}\|} = \frac{F}{\|\vec{\omega}\|}$$

For convenience, let us choose to require that functional margin of all data points is atleast 1 and that is equal to 1 for at least one data vector. which, means, for all items in the data :-

$$y_i(\vec{\omega}^T\vec{x}_i+b) \geq 1$$

and there exist support vectors for which the inequality is an equality.

$$\therefore\ F = \min(y(\omega^T x+b)) = 1$$

So, $G = \frac{1}{\|\vec{\omega}\|}$

~~Considering margin for both clusters, net margin will be :-~~

$$\cancel{\rho = \frac{2}{\|}}$$

Due to some reasons, a factor of 2 is also introduced

$$\rho = \frac{2}{\|\vec{\omega}\|}$$

Optimisation problem we got :-

maximise $\frac{2}{\|\vec{\omega}\|}$ for $\vec{\omega}, b$

S.T.C $\forall\ (\vec{x}_i, y_i) \in D$

$$y_i(\vec{\omega}^T\vec{x}_i+b) \geq 1$$

Now, problem is that $\frac{1}{\|\omega\|}$ i.e. $\frac{1}{\sqrt{\omega_1^2+\omega_2^2+\cdots\omega_n^2}}$ is a non-convex function (such functions have many local optimas)

Take reciprocal and remove square root to make it convex

$\Rightarrow$

$$\min_{\vec{\omega}, \vec{b}} \frac{1}{2}\|\vec{\omega}\|^2 \quad \text{or} \quad \min_{\vec{\omega}, \vec{b}} \frac{1}{2}\vec{\omega}^T\vec{\omega}$$

Final Optimization problem of SVM :-

$$\boxed{\begin{array}{l}\min_{\vec{\omega},b} \frac{1}{2}\|\vec{\omega}\|^2 \\ \text{S.T.C :-} \\ \text{For all } (\vec{x}_i, y_i), \\ y_i(\vec{\omega}^T\vec{x}_i + b) \geq 1\end{array}}$$

Method of Lagrange Multiplier + KKT conditions :-

It is a method to optimise fxns subjected to constraints.

$f(x)$: objective fxn which is to be optimised

constraints :-

$$g_i \leq 0 \; ; \; i = 1, 2 \ldots k$$

Lagrangian fxn :

$$L(x, \alpha) = f(x) + \sum_{i=1}^{k} \alpha_i g_i$$

uses − in case of $g \geq 0$

α : larange multiplier

- We work on ~~opt~~ L instead of f then..

Primal and Dual Concept :-

Mini. $f(x)$ (Primal Problem)

S.T. $g_i(x) \leq 0, \; i = 1, 2, \ldots k$

$$L(x, \alpha) = f(x) + \sum_{i=1}^{k} \alpha_i g_i$$

~~Mini~~ $\boxed{\alpha_i \geq 0}$ (from perceptron algorithm)

$$\text{Mini}_{x} f(x) \equiv \min_{x}\left[\max_{\substack{\alpha \\ \text{S.T } \alpha \geq 0}} L(x, \alpha)\right]$$

(Primal)

Reason :

$\alpha_i \, g_i(\omega) = (+ve)(-ve) = -ve$

So, in order to maximise L, we have to set $\alpha = 0$. That will yield original problem only.

↳ when α becomes 0, $\alpha_i g_i = 0$

In general :-

$$\max_{\alpha}\left[\min_{x} L(x, \alpha)\right] \leq \min_{x}\left[\max_{\alpha} L(x, \alpha)\right]$$

dual $\leq$ Primal

We solve dual problem ∵ :-

In most cases dual is easier to solve.

Since dual is less than or equal primal, it is good ~~of~~ for us as we are ~~were~~ already in search of minimum value in primal problem. Here, we are getting even more less value, which favours us.

Primal = Dual when KKT conditions are satisfied

KKT conditions :-

1) Convert to Lagrangian fxn. Take ~~parial~~ partial derivatives w.r.t variables and equate them with 0.

2) $\alpha_i g_i = 0$

(Practical see that either $\alpha_i = 0$ or $g_i = 0$ is found, not both)

3) $g_i \leq 0$

4) $\boxed{\alpha_i \geq 0}$

Take derivative of L w.r.t X and equate to 0. Eliminat X from dual, then we will left only to minimise w.r.t α.

Illustration :

(i) Minimise $w_1^2 + w_2^2$

S.T.C $2w_1 + 3w_2 \geq 1$

Sol:-

$1 - 2w_1 ~~*~~ - 3w_2 \leq 0$ {$g \leq 0$ form}

$L(w_1, w_2, \alpha) = w_1^2 + w_2^2 + \alpha(1 - 2w_1 - 3w_2)$

Dual:- $\max\limits_{\alpha} \left[\min\limits_{w_1, w_2} L(w_1, w_2, \alpha) \right]$

S.T. ~~KKT conditions~~ $\alpha \geq 0$

$$\begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \\ \frac{\partial L}{\partial w_2} \end{bmatrix} = \begin{bmatrix} 0 \\ \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 2w_1 - 2\alpha \\ \\ 2w_2 - 3\alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \\ 0 \end{bmatrix}$$

$\Rightarrow w_1 = \alpha$ and $w_2 = \frac{3}{2}\alpha$

Put these w_1 and w_2 in L :-

$$L = \alpha^2 + \frac{9\alpha^2}{4} + \alpha - 2\alpha^2 - \frac{9\alpha^2}{2}$$

$$L = -\frac{13\alpha^2}{4} + \alpha$$

dual problem :- $\max\limits_{\alpha} L(\alpha)$

i.e. $\max -\frac{13}{4}\alpha^2 + \alpha$

S.T. KKT conditions

$$\frac{\partial L}{\partial \alpha} = 0$$

$$\Rightarrow -\frac{13}{2}\alpha + 1 = 0$$

$$\Rightarrow \alpha = \frac{2}{13} \quad \{\alpha \geq 0\}$$

Satisfied

$\therefore w_1 = \frac{2}{13}$ and $w_2 = \frac{3}{13}$

Minima:- $\left(\frac{2}{13}, \frac{3}{13}\right)$

Checking if KKT conditions are ~~b~~ satisfies :-

$$1 - 2w_1 - 3w_2 = 1 - 2\times\frac{2}{13} - 3\times\frac{3}{13}$$
$$= 0$$

$\because g \leq 0$ satisfied ✓

$\alpha_i g_i = 0$ $\{\because g = 0\}$ ✓

ii) Mini. $w_1^2 + w_2^2$

S.T.C $2w_1 + 3w_2 \leq 1$

sol:-

$$2w_1 + 3w_2 - 1 \leq 0 \quad (g \leq 0)$$

$$L(w_1, w_2, \alpha) = w_1^2 + w_2^2 + \alpha(2w_1 + 3w_2 - 1)$$

$$\max_{\alpha} \left[\min_{w_1, w_2} L(w_1, w_2, \alpha) \right]$$

S.T $\alpha \geq 0$

$$\frac{\partial L}{\partial w_1} = 0 \Rightarrow 2w_1 + \cancel{2}\alpha = 0$$

$$\Rightarrow \cancel{w_1}\, w_1 = -\alpha$$

$$\frac{\partial L}{\partial w_2} = 0 \Rightarrow 2w_1 + 3\alpha = 0$$

$$\Rightarrow w_1 = -\frac{3\alpha}{2}$$

So $L = -\frac{13}{4}\alpha^2 - \alpha$

dual:- $\max_{\alpha} \; -\frac{13\alpha^2}{4} - \alpha$

S.T $\alpha \geq 0$

$$\frac{\partial L}{\partial \alpha} = -\frac{13\alpha}{2} - 1 = 0$$

$$\Rightarrow \alpha = \frac{-2}{13}$$

But $\alpha \geq 0$

So take $\alpha = 0$

$\Rightarrow w_1 = 0, \quad w_2 = 0$

Minima :- $(0, 0)$

$g \leq 0$ $\quad \alpha g = 0 \quad (\because \alpha = 0)$

$-1 \leq 0$ ✓ $\quad$ ✓

⇒ **Dealing with our SVM optimilation problem :-**

$$\min_{\vec{w}, b} \frac{1}{2} \|\vec{w}\|^2 \quad \text{(Primal)}$$

S.T.C $y_i(\vec{w}^T \vec{x}_i + b) \geq 1 \; \forall (x_i, y_i)$

Lagrange fxn :-

$$L(\vec{w}, b, \alpha) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\vec{w}^T x_i + b) - 1]$$

$$\min_{\vec{w}, b} \left[\max_{\alpha_1, \alpha_2 \ldots \alpha_n} L(\vec{w}, b, \alpha_1, \alpha_2 \ldots \alpha_n) \right]$$

S.T. $\alpha_i \geq 0$ $\quad$ (Primal)

$$\max_{\alpha_1, \alpha_2 \ldots \alpha_n} \left[\min_{\vec{w}, b} \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\vec{w}^T x + b) - 1) \right] \quad \text{--- (1)}$$

S.T. $\alpha_i \geq 0$

Taking partial derivative w.r.t $\vec{w}$, b and equating them to 0.

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \vec{w}} = 0 \Rightarrow \vec{w} = \sum_{i=1}^{n} \alpha_i y_i X_i$$

features vector

Now, plug in the value of of $\vec{w}$ and $\sum \alpha_i y_i = 0$ in (1)

$$\max_{\alpha_1,\alpha_2\ldots\alpha_n} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j X_i^T X_j$$

S.T.C. $\forall (x_i, y_i)$

$$\alpha_i \geq 0$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

Dual Problem of SVM

Solution from SVM dual problem

$$\vec{w} = \sum \alpha_i y_i X_i$$

$b = y_k - \vec{w}^T\vec{x_k}$ for any $\vec{x_k}$ such that $\alpha_k \neq 0$

$\Rightarrow$ support vector

Each non-zero α_i indicates that corresponding $\vec{x_i}$ is support vector

All KKT conditions under which primal = dual in SVM :-

1.) $\forall (x_i, y_i),\ y_i(\vec{w}^T\vec{x_i} + b) - 1 \geq 0$

{ Primal problem constraint }

2.) $\frac{\partial L}{\partial \vec{w}}(\vec{w}^*, b^*, \alpha^*) = 0$

3.) $\frac{\partial L}{\partial \alpha}(\vec{w}^*, b^*, \alpha^*) = 0$

sufficient conditions which are sufficient to find maxima or minima

4.) $\forall_i\ \alpha_i \geq 0$ (from perceptron algorithm)

&

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

$\alpha = 0$ for almost all data points.
$\alpha \neq 0$ for only support vectors
α_i are weights
$\alpha = 0 \Rightarrow$ no ~~penalty~~ weight
$\alpha \neq 0 \Rightarrow$ weight is increased to avoid misclassification

{ 4.) is basically the Dual problem constraint }

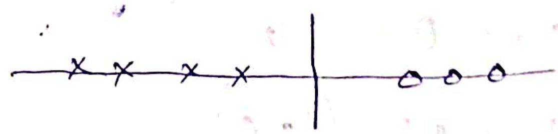
5.) $\forall_i\ \alpha_i[y_i(w^T x_i + b) - 1] = 0$

Most Important Advantage of Dual Problem in SVM :-

We are getting the dot product of feature vectors X_i and X_j

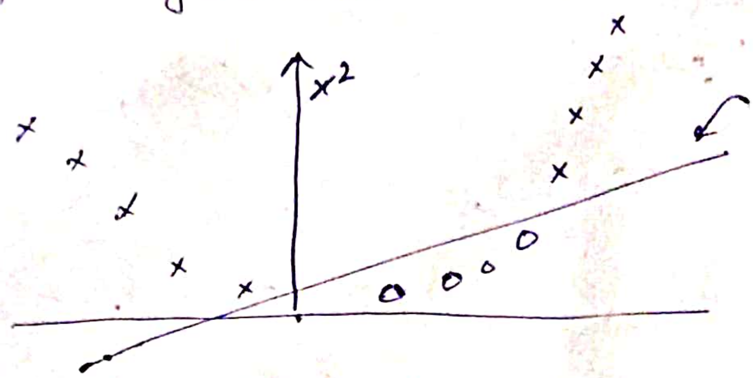
It is known that linear classifier fails to provide good classification. We do feature engineering to map our ~~fea~~ data to higher dimensions. Then we are able to draw linearly separating hyperplane.

* Linearly separable:



* Non-linearly separable:-

Projecting into higher dimension



This feature engineering involves hand coding which takes lot of efforts and it increases computional complexity for finding discriminant score.

However, in SVM, $X_i^T X_j$ naturally yields us engineered features without having the need to hand code them just with slight modification while providing fast calculation of dot product at the same time.

~~At~~

Kernel function: It is a function of vectors which projects the two vectors into original or higher dimensional space and takes their dot product (like $X_i^T X_j$) after projecting them and return the result of dot product.

Suppose $X_i = \begin{bmatrix} x_1^i \\ x_2^i \end{bmatrix}$ & $X_j = \begin{bmatrix} x_1^j \\ x_2^j \end{bmatrix}$

and modify $X_i^T X_j$ to Quadratic kernel $(X_i^T X_j + c)^2$

$$K(X_i, X_j) = (X_i^T \cdot X_j + c)^2$$

$$= (x_1^i \cdot x_1^j + x_2^i \cdot x_2^j + c)^2$$

$$= (x_1^i \cdot x_1^j + x_2^i \cdot x_2^j)^2 + c^2 + 2 \cdot c \cdot (x_1^i \cdot x_1^j + x_2^i \cdot x_2^j)$$

(On rearranging further)

$$= (x_1^i)^2 \cdot (x_1^j)^2 + (x_2^i)^2 \cdot (x_2^j)^2 + \sqrt{2} \cdot x_1^i \cdot x_2^i \cdot \sqrt{2} \cdot x_1^j \cdot x_2^j + c \cdot c$$
$$+ \sqrt{2c}\, x_1^i \cdot \sqrt{2c} \cdot x_1^j + \sqrt{2c} \cdot x_2^i \cdot \sqrt{2c} \cdot x_2^j$$

Now, we can see feature engineering.

$$X_i = \begin{bmatrix} x_1^i \\ x_2^i \end{bmatrix} \longrightarrow X_i' = \begin{bmatrix} (x_1^i)^2 \\ (x_2^i)^2 \\ \sqrt{2}\, x_1^i \cdot x_2^i \\ c \\ \sqrt{2c} \cdot x_1^i \\ \sqrt{2c} \cdot x_2^i \end{bmatrix} \qquad X_j' = \begin{bmatrix} (x_1^j)^2 \\ (x_2^j)^2 \\ \sqrt{2}\, x_1^j \cdot x_2^j \\ c \\ \sqrt{2c}\, x_1^j \\ \sqrt{2c}\, x_2^j \end{bmatrix} \longleftarrow X_j = \begin{bmatrix} x_1^j \\ x_2^j \end{bmatrix}$$

This implicit projection of the two feature vectors into a higher dimensional space while computing different slight modification with constant time complexity, is what we call the KERNEL TRICK.

Now, if we ~~com~~ add kernel trick to our optimization dual problem

$$\boxed{\begin{gathered} \max_{\alpha_1, \alpha_2 \ldots \alpha_n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(X_i, X_j) \\ \text{S.T.C} \quad \forall_i \quad \alpha_i \geq 0 \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{gathered}}$$