15/2/23

Date / /
Page No.

SVM (SUPPORT VECTOR MACHINE)

(Most powerful & imp algo in ML)

Topics -
- → Formulating Objective
- → Objective as Constraint Convex Function
- → Handling Outliers
- → Weight & Bias Update Rule
- → Hinge loss function
- → Handling non-linearly separable data
- → Kernel Trick Formulation
- → Different types of Kernel

⇒ $X = \{x_1, x_2, x_3, \dots x_n\}$
$Y = \{y_1, y_2, y_3 \dots y_n\}$ ← input data or training data

$y_i \in \{-1, 1\}$

Key idea - Separate data with maximum margin hyperplane

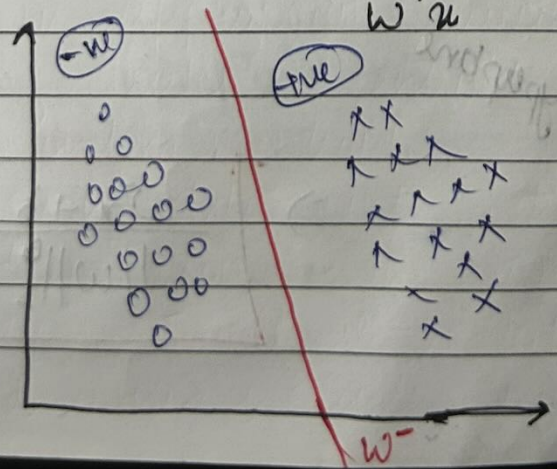
$w^T x + b = 0$ (eq^n of hyperplane)

(hyperplane - having 2 or more features)

$[\theta_0 + \theta_1 x_1^{(1)} + \theta_2 x_2^2 + \theta_3 x_3^3 + \dots]$

$\theta_0 \rightarrow b$

$\theta_1 x_1^{(1)} + \theta_2 x_2^2 + \theta_3 x_3^3 + \dots \rightarrow w^T x$



{ Hyperplane - The best decision boundary }

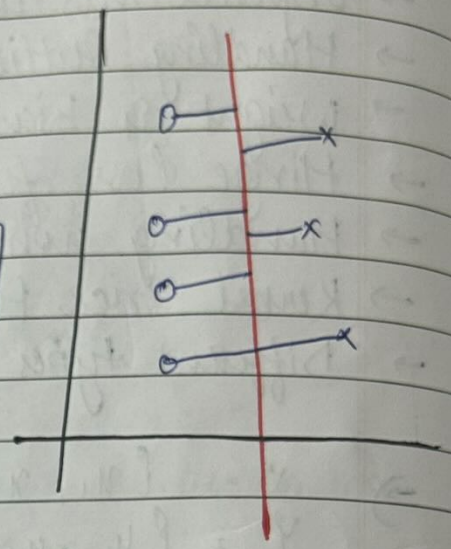
→ $x_i^{(i)}$ → superscript (sample)
→ x_i → subscript (features)

⇒ Optimal Hyperplane

- $w^T x + b > 0$
  if $x^{(i)} \in$ the class
- $w^T x + b \leq 0$
  if $x^{(i)} \in$ -ve class

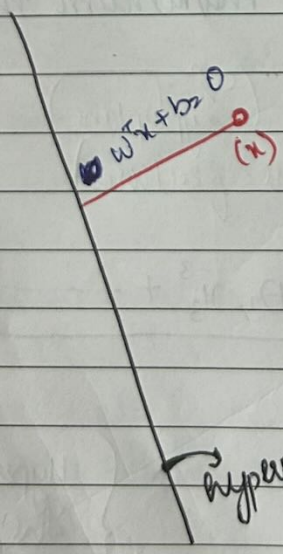
$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Weight matrix Input matrix



$g(x) = +1$ if $z \geq 0$
$g(x) = -1$ if $z < 0$

Distance



⇒ $$\frac{w^T x + b}{\sqrt{w_1^2 + w_2^2 + \cdots - w_n^2}}$$

Euclidean norm of weights
or
L2 norm

⇒ $$\frac{w^T x + b}{\|w\|_2}$$

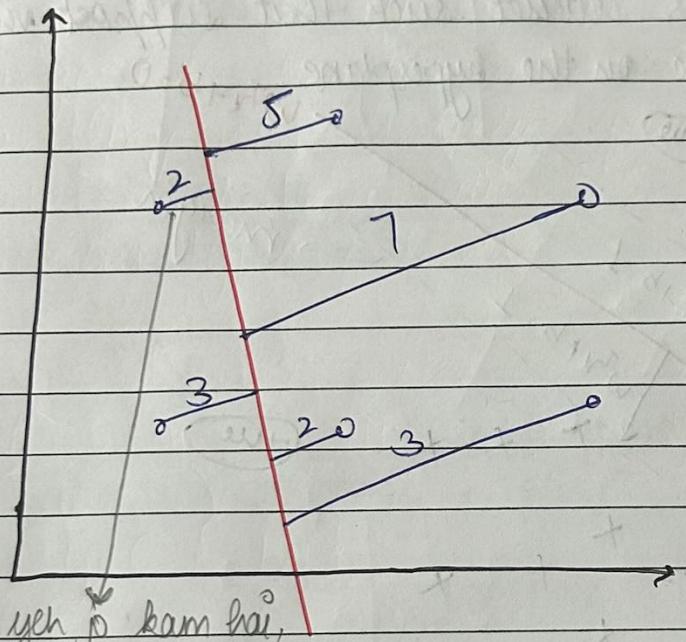
Date / /
Page No.

⇒ Goal - To maximize the min distance of point from the hyperplane

$$\gamma^{(i)} = \frac{w^T x^{(i)} + b}{||w||_2}$$

Yeh is how we find the distance now, try to formulate the goal of SVM using this notation.

Maximise min distance $\quad \gamma = \min \gamma^{(i)}, \quad i = 1, \dots, n$



yeh 5 kam hai, unka distance increase krna hai taaki line distort ho ske & misclassified points correctly classify ho ske.

Optimisation / Sum Objective

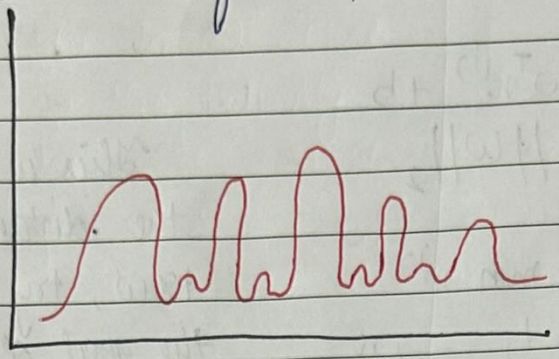
$\max_{\gamma, w, b}$ such that

Absolute distance → $y^{(i)} \cdot w^T x^{(i)} + b \geq \gamma$ for all $i = 1, 2, \dots m$

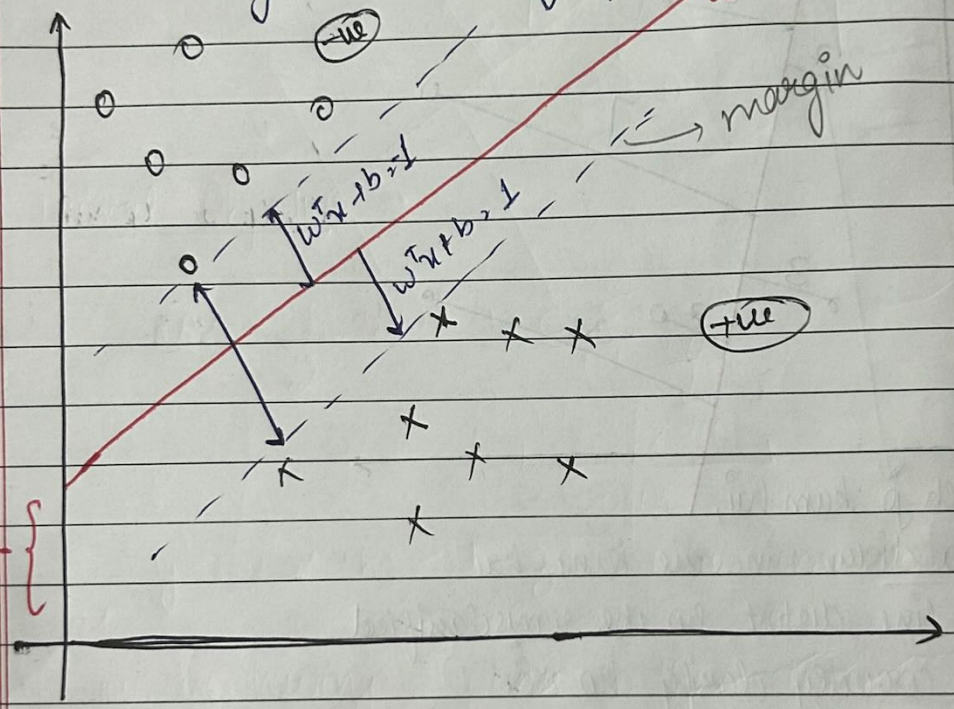
$w^T x^{(i)} + b$ {+ve class}
$-(w^T x^{(i)} + b)$ {-ve class}

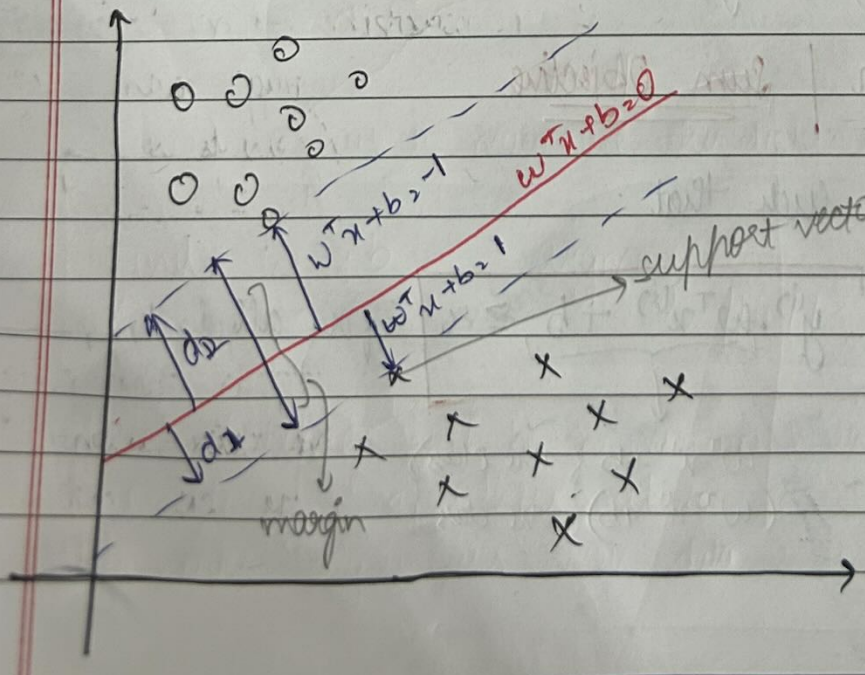
Date / /
Page No.

* The above written absolute distance $(y^{(i)} + w^T x^{(i)} + b \geq 1)$ is a non-convex function.



⇒ Renormalise our dataset such that support vectors should always lie on the hyperplane.





→ Goal- Maximize d such that $y_i(w^T x^{(i)} + b) \geq 1$

$$d_1 = \frac{|w^T x_i + b|}{||w||} = \frac{1}{||w||} \quad \{\because w^T x_i + b = 1\}$$

$$d_2 = \frac{|w^T x_i + b|}{||w||} = \frac{1}{||w||}$$

$$d = d_1 + d_2 = \frac{1}{||w||} + \frac{1}{||w||} = \frac{2}{||w||}$$ → d-1 maximum

↳ Maximise

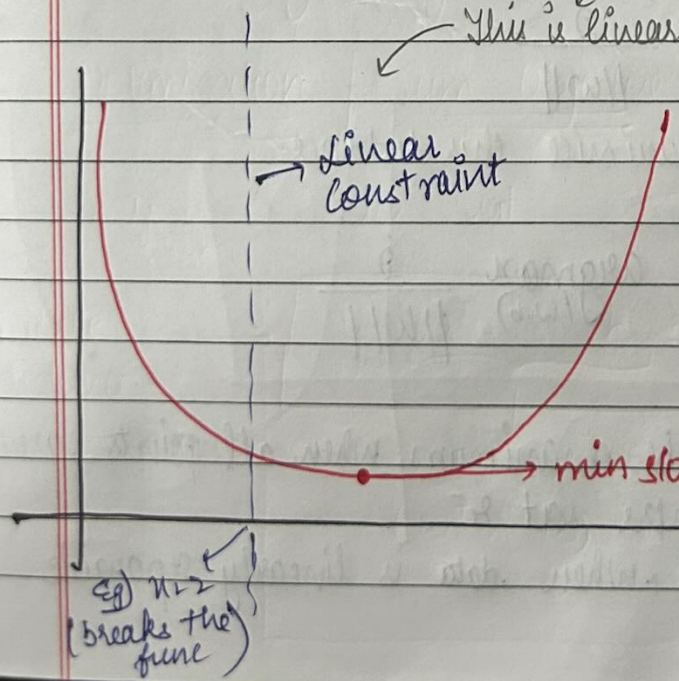
∴ we minimize $= \frac{||w||}{2}$ { reciprocal isliye kiya kyuki hum hmesha minimize krte aa rhe hai }

2)

Final SVM objective → Min { convex function $\frac{1}{2}||w||$ }

$y_i(w^T x^{(i)} + b) \geq 1$ ← This is linear constraint → Quadratic solver

↳ Lagrangian duality (converts constrained optimisation to unconstrained optimisation)

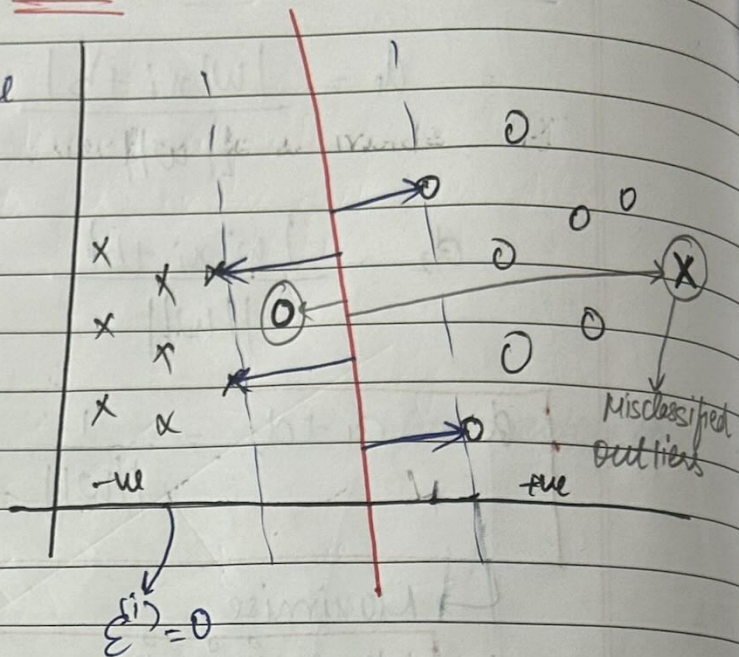


{ Pegasos - linear constraint ko kaise unconstrained bnaa ekte hai }

Date / /
Page No.

Handling Outliers in SVM

Allow one to do some error on the training examples
$(x_i, y_i) \rightarrow \xi^{(i)}$



⇒ Primal Objective →

$$\text{Loss} = \frac{1}{2} ww^T + C \sum_{i=1}^{m} \zeta^{(i)}$$

ζ : zeta

Error

$$y^{(i)}(w^T x^{(i)} + b) \geq 1 - \zeta^{(i)}$$ } Allow some error

If you are having 2 planes so there distance will be,

$$\text{distance} = \frac{2}{||w||}$$

now we have to maximise this distance

$$\text{find } w \text{ & } b = \underset{(w,b)}{\text{argmax}} \frac{2}{||w||}$$

→ This distance will be maximum when all points are on one side of x^+ & others sat x^-

→ This works only in cases, when data is linearly separable

$\underset{w,b}{\text{argmax}} \frac{2}{\|w\|}$ } → Hard Margin SVM (does not work on non-linearly separable data)

↳ $\underset{w,b}{\text{argmin}} \frac{\|w\|}{2}$

Date / /
Page No.

⇒ Hard Margin SVM {also called}

Sol → Soft Margin SVM → for non-linear separable data
(The problem is of outliers)

2) $\underset{w,b}{\text{argmin}} \frac{\|w\|}{2} + C\frac{1}{n}\sum_{i=1}^{n} \xi_i^{(i)}$ → hyperparameter

($\xi_i^{(i)}$: error; $C\frac{1}{n}\sum_{i=1}^{n} \xi_i^{(i)}$: penalty)

Soft Margin SVM (bcoz we added error)

$f(x) \rightarrow \max \rightarrow \min\left(\frac{1}{f(x)}\right)$

Why minimising now?

(2) $\frac{1}{n}\sum_{i=1}^{n} z_i$ → Average distance of misclassified points

(to those points which are not correctly classified, that distance average it is)

(3) Pegasos Algorithm for unconstrained optimisation

* $w^T x_1 + b = 1$
$w^T x_2 + b = -1$
$w^T(x_1 - x_2) = 2$
$\frac{w^T(x_1 - x_2)}{\|w\|} = \frac{2}{\|w\|}$

$\therefore \boxed{(x_1 - x_2) = \frac{2}{\|w\|}}$

* SVM - find w & b such where $\frac{\|w\|}{2}$ is minimum

16/2/23

Pegasos Algorithm for Unconstrained Optimisation

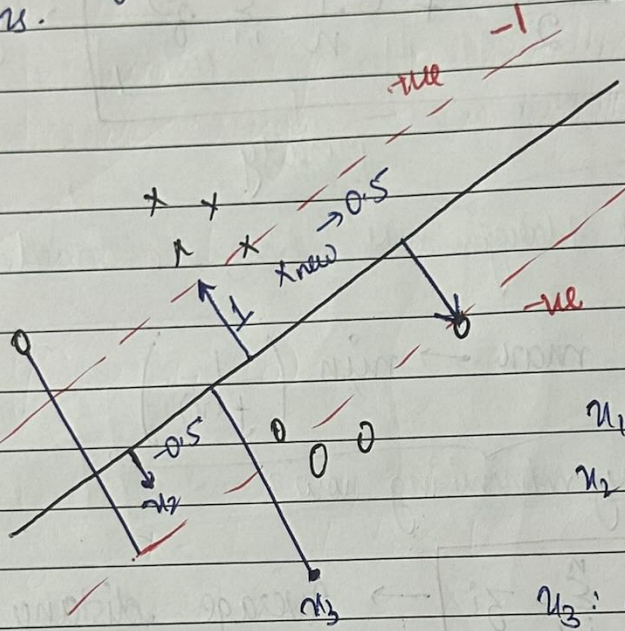
→ Objectives-

i) Develop a gradient descent algorithm for SVM

ii) Introduce sub-gradients for convex but non-differentiable cost functions.

$y_i w^T x > 0$ then correctly classified

if $y_i w^T x_i > 1$ misclassified



x_1: d = 0.5 (1-0.5)

x_2: d = -0.5

1-(-0.5) = 1.5

x_3: d = -1.5

1-(-1.5) = 2.5

Distance from correct plane or margin

$$\underset{w,b}{\operatorname{argmin}} \frac{||w||}{2}$$

* SVM mein non-convex function kyun hai, hum maan lete hyperplanes

⇒ Non Convex

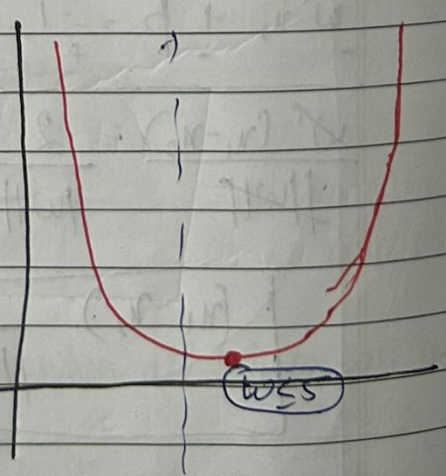
↓

Convex with Constraints

↓

Convex with Constraints + Outliers

$$\left[\min \frac{||w||}{2}\right] \text{s.t. } y^{(i)}(w^T x + b) \geq 1$$



⇒ Due to constraints, we're unable to differentiate the function

* Constraint function

↓ conversion

Unconstrained function

⇒ When it will get converted, the differentiation problem will get solved. Gradient Descent will be able to optimise

→ When $d > 1$, this is the case of misclassification (This is the error)

⇒ d should be less than 1 (for correct classification)

⇒ Misclassified distance from correct plane:

1 - (distance of misclassified points from correct plane

↓

ζ_i (zeta)

Note: ζ_i will increase as point is further away from correct plane

ζ_i → is for outliers (~~to~~ error)

$y_i w^T x_i > 0$ ⇒ correctly classified

Class label. distance ≥ 0

* we have to tune the hyperparameter.

Date / / Page No.

⇒ Loss Function / Objective

$$w, b = \underbrace{\text{argmin} \frac{\|w\|}{2}}_{\text{Regulariser}} + \underbrace{C}_{\text{Hyperparameter / penalty}} \cdot \frac{1}{n} \sum_{i=1}^{n} \underbrace{z_i}_{\text{error}}$$

↳ Loss function

* If C ko increase krenge → overfitting
  C ko decrease krenge → underfitting

⇒ $d = 1 - y_i w^T x_i \geq 1$

1) $z^{(i)} = 1 - y^{(i)}(w^T x_i + b)$

2) $z^{(i)} \geq 0$

   $z^{(i)} \geq 1 - t_i$ if $t_i \leq 1$

   $z^{(i)} \geq 0$ if $t_i \geq 1$

⇒ $z^{(i)} = 1 - t_i$ if $t_i \leq 1$

$z^{(i)} = 0$ if $t_i > 1$

$$z^{(i)} = \max(0, 1 - t_i)$$

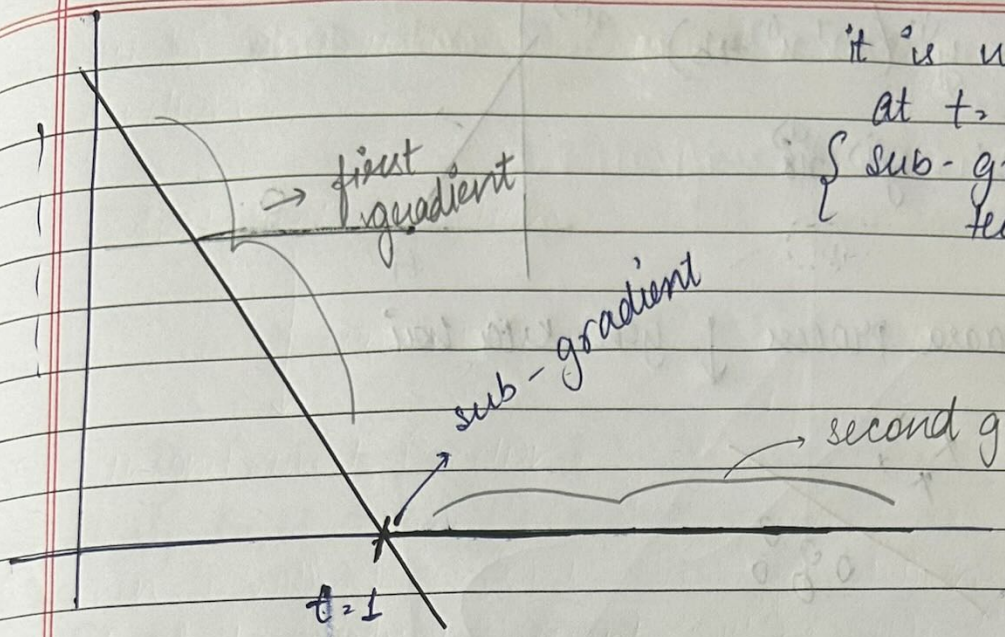
Loss / objective

$$\min_{w,b} \frac{1}{2} \frac{\|w\|}{2} + C \sum_{i=1}^{m} \max(0, 1 - t_i)$$

where, $t_i = y^{(i)}(w^T x^{(i)} + b)$

→ unconstrained convex optimisation

↳ Hinge loss



it is not differentiable at $t=1$

{ sub-gradient technique }

$$w + C \sum_{i=1}^{m} \nabla_w [\max(0, 1 - t_i)]$$

$$= w + C \sum_{i=1}^{m} \nabla_{t_i} [\max(0, 1 - t_i)]$$

$\nabla_w t_i$ ← CHAIN RULE

$$y = w + C \sum_{i=1} \begin{cases} 0 & \text{if } t_i \geq 1 \\ -1 & \text{if } t_i < 1 \end{cases} g^{(i)} x^{(i)}$$

⇒ Differentiation of Hinge Loss

$$\Delta w = w + C \cdot \sum_{i=1}^{m} \nabla_w [\max(0, 1 - t_i)]$$

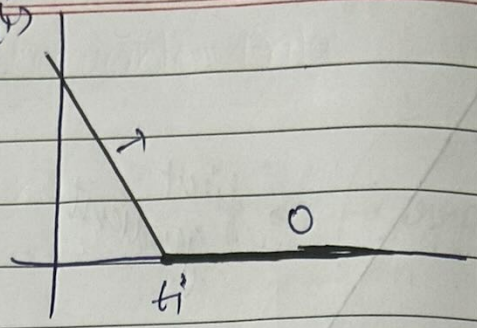
$$= w + C \sum_{i=1}^{m} \nabla_t \left[|\max|(0, 1 - t_i) \right] \nabla_w t_i$$

↓ CHAIN RULE

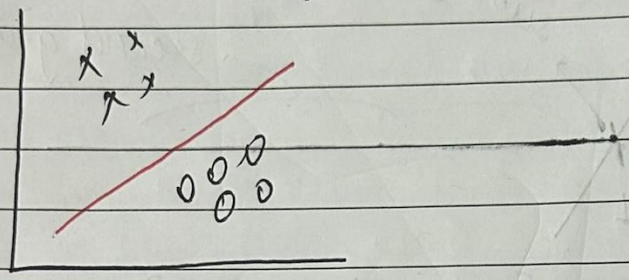
$$= w + C \cdot \sum_{i=1}^{m} \begin{cases} 0 & \text{if } t_i \geq 1 \\ -1 & \text{if } t_i < 1 \end{cases}$$

$t_i = y^{(i)}(w^T x^{(i)} + b)$

$\Delta_w t_i = y^{(i)} x_i^{(i)}$



yeh saara process ↓ yeh karta hai



20/2/23.

SVM : Weight & Bias Update Rule [The final update rule]

$$L(w) = \frac{1}{2}\|w\| + C \sum_{i=1}^{m} \max(0, 1 - t_i)$$

where, $t_i = y^{(i)}(w^T x + b)$

Convex loss function will be Gradient Descent to find optimal values of w & b.

i) $w = w - \eta \cdot \Delta_w L$ (derivative of Loss w.r.t w)

ii) $b = b - \eta \Delta_b L$

[(i) & (ii) are two equations we have to find]

$$\Rightarrow \Delta_w L = w + C \begin{bmatrix} 0 & t_i \geq 1 \\ -1 & t_i < 1 \end{bmatrix} y_i x_i$$

(after derivating eq (i) & (ii) w.r.t w & b resp.

Date / /
Page No.

$$\Rightarrow \Delta_b L = 0 + C \begin{bmatrix} 0 & t_i \geq 1 \\ -1 & t_i < 1 \end{bmatrix} y_i$$

(after derivating eq (i) & (ii) w.r.t w & b resp.)

$$\Delta_w t_i = y_i (w^T x^{(i)} + b)$$

i) $\Delta_w t_i \rightarrow y^{(i)} w^{(i)}$ in w.r.t w
ii) $\Delta_b t_i \rightarrow y^{(i)}$ w.r.t b

→ **Final weight update rule**

$$w = w - \eta \left[w + C \begin{bmatrix} 0 & t_i \geq 1 \\ -1 & t_i < 1 \end{bmatrix} y_i x_i \right]$$

expanding above eqn

Simplifying :

$$\Rightarrow w = w - \eta w \begin{bmatrix} 0 & t_i \geq 1 \\ nc[y_i x_i] & t_i < 1 \end{bmatrix}$$

$$\Rightarrow b = b - \eta \left[C \begin{bmatrix} 0 & t_i \geq 1 \\ -1 & t_i < 1 \end{bmatrix} y_i \right]$$

$$\begin{cases} b = b + 0 & \text{if } t_i \geq 1 \\ b = b + nCy_i & \text{if } t_i < 1 \end{cases}$$

* Margin maximisation works well on linear separable data.

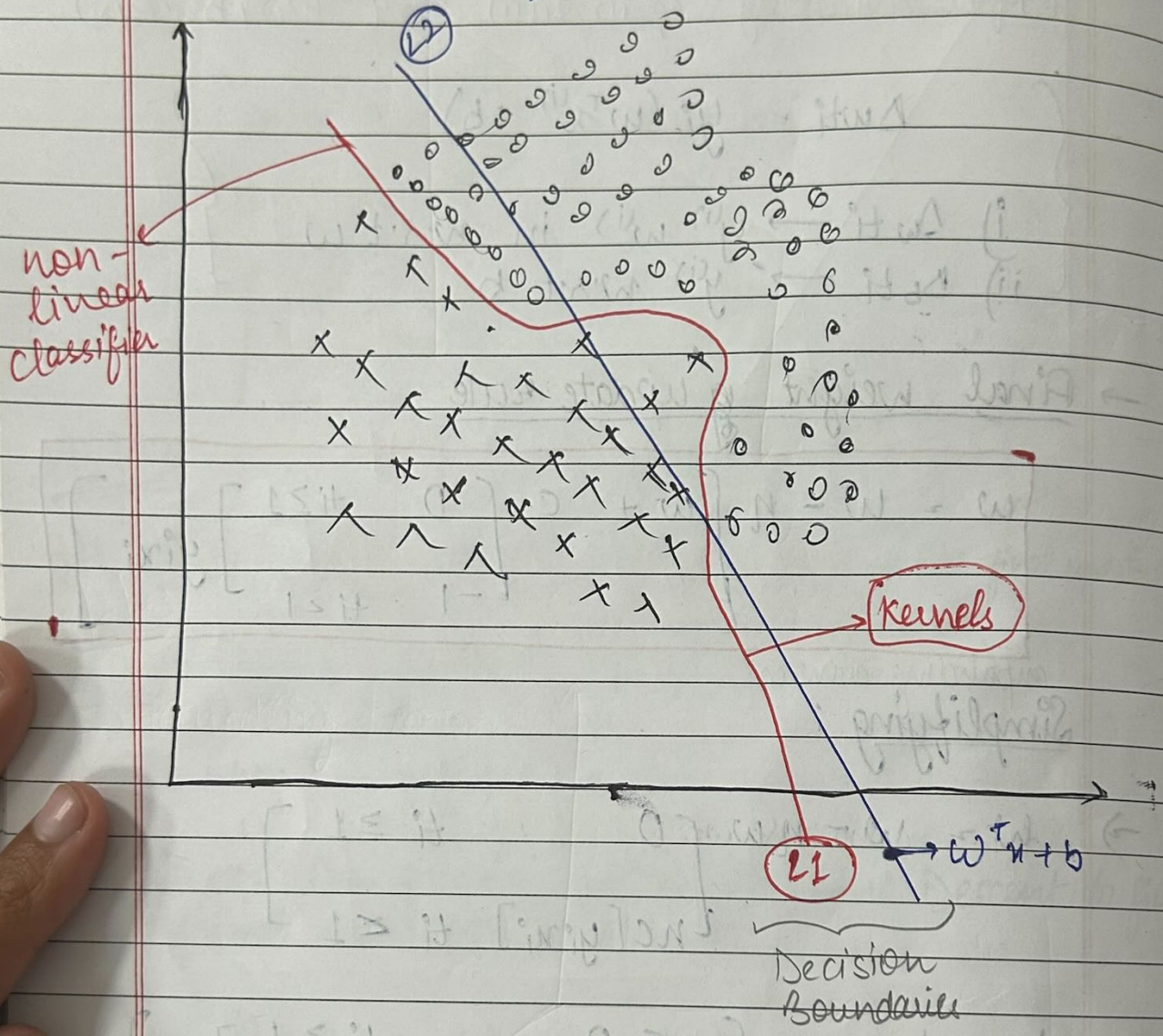
SVM is for both regression & Classification

→ SVCA support vector classifier

Date / /
Page No.

Handling Non-linear Separable Data

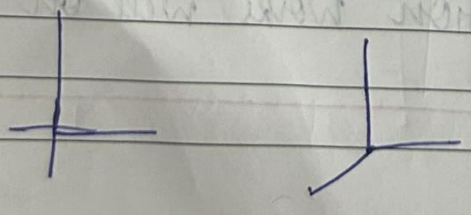
Non-Linear Classification



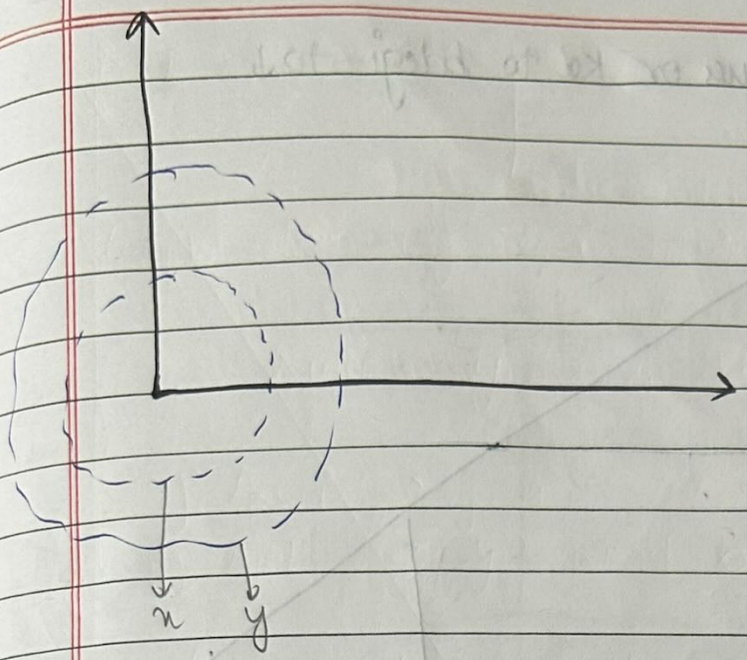
⇒ Send this data to higher dimension & separate the classes using linear classifier.

⇒ Key Idea - Project Data to High Dimension

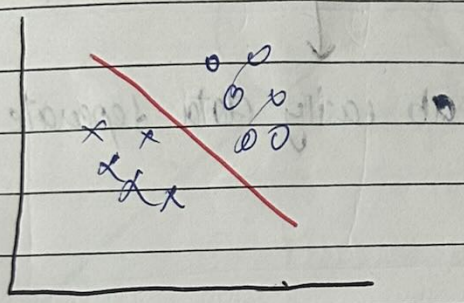
$x = [x_1, x_2]$ → Transform → $x = [x_1, x_2, x_3]$



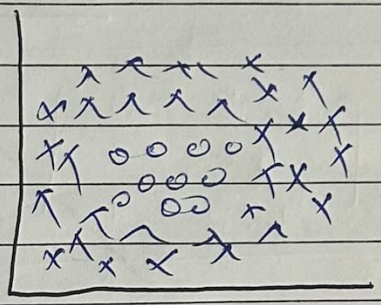
$x_4 = x^2$ . . .



*

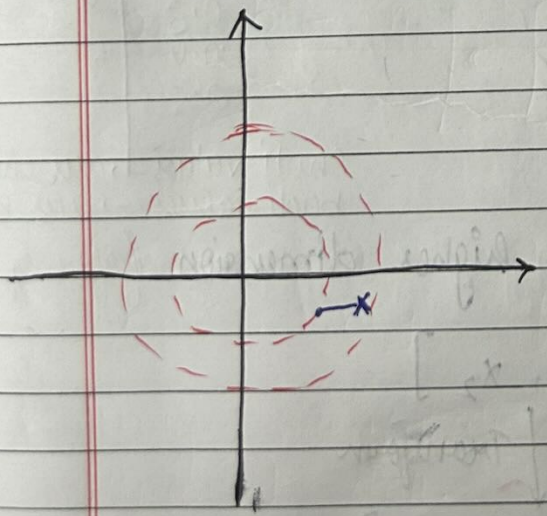


Linear separable data
(Margin maximisation is used)



Non-linear separable data
(Kernel tricks is used)

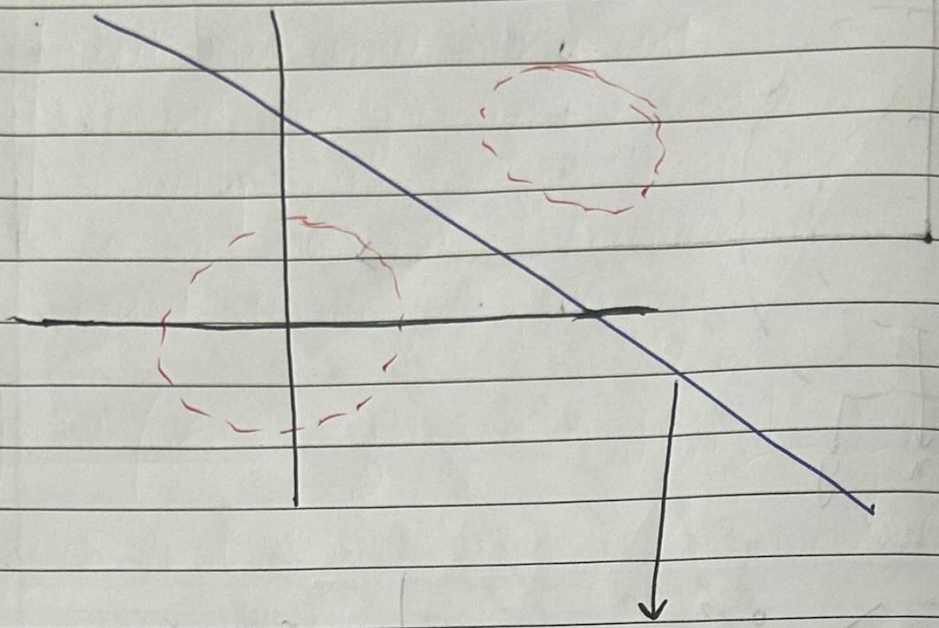
* Kernel trick uses math to separate data



x_1	y (x_2)
0.1	2
0.2	4
↓	↓

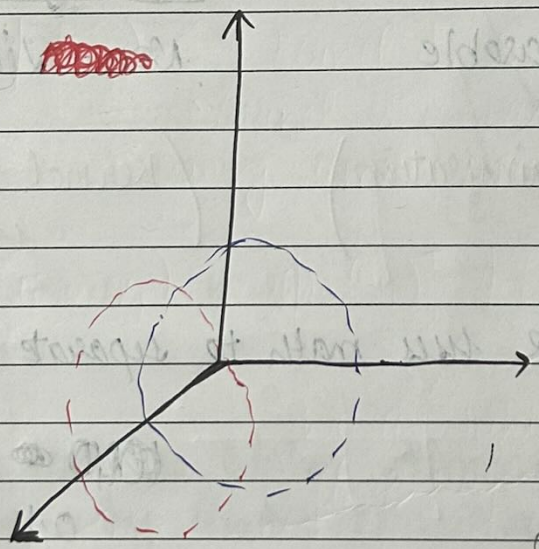
ab; $x_3 = x_1^2 + x_2^2$

x_1^2 krenge to values or ka to bdegi to



ab easily data seperate hojayega

22/2/23



$x_3 = x_1^2 + x_2^2$

x_1	x_2
0.2	2
0.3	4
0.4	6
0.5	8

choti values → aur choti
badi values → aur badi

Key Idea - Project data to higher dimension (after SVM)

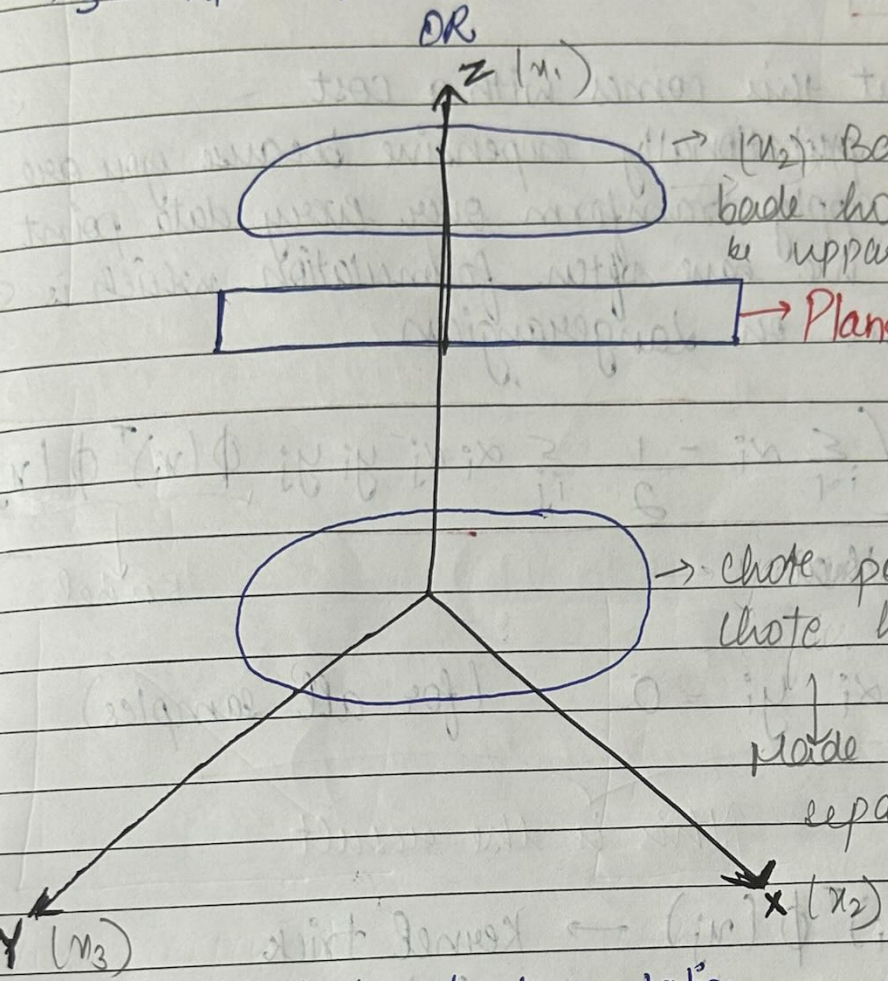
$$x = [x_1, x_2]$$

↓ Transform

$$x = [x_1, x_2, x_3]$$

Date / /
Page No.

⇒ $x_3 = x_1^2 + x_2^2$



'Kernel Trick' based formulation

Ques) How can you apply svm to non linearly separable data?

⇒ The first change you have to make to the formulation is

$$\min \frac{1}{2}||w|| + C \sum_{i=1} \xi_i \; y_i (w^T x_i + b) \geq 1 - \xi_i$$

loss function

$x_i \rightarrow \phi(x_i)$

will transform x_i & change x_i

Date / /
Page No.

⇒ $\phi(x_i)$

↳ but this comes with a cost
↳ computationally expensive because you are going to apply transform over every data point
↳ so we have often formulation which is SVM based on Langerangian

⇒ $\max\left(\sum_{i=1} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j \underbrace{\phi(x_i)^T \phi(x_j)}_{\text{Kernel}}\right)$

such that

$\sum \alpha_i y_i = 0$ (for all samples)

This is the result

⇒ $\phi(x_i)^T \phi(x_j)$ → Kernel trick
↳ kernel is a particular function having some property.

⇒ $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$

We are taking product of 2 projected vectors

⇒ RBF
Polynomial
Sigmoid
Linear
} Different types of Kernel

Different kernels decision boundary banate hai

Date / /
Page No.

⇒ RBF :- Radial Bias Kernel

$$K(x_i, x_j) = e^{-\gamma |x_i - x_j|^2}$$

{γ = gamma}

→ coefficient hai (yeh control karta hai)

↓ This does transformation

⇒ Polynomial Kernel :-

$$K(x_i, x_j) = (\gamma x_i^T \cdot x_j + r)^c$$

↓ degree of polynomial

⇒ Sigmoid -

$K(x_i, x_j)$

↓

$$K \text{ of } x_i, x_j = \frac{1 - e^{-2(\gamma x_i \cdot x_j + r)}}{1 - e^{-2(\gamma x_i \cdot x_j + r)}}$$

Ques) Why these kernels helps in simple calculation?

⇒ ϕ is a map from n-dimension to m-dimension space

Compute ϕ(x), ϕ(y)

↓ do dot product

→ this will be expensive, isliye Kernel trick dekha

⇒ Without kernel trick,

Eg) Suppose X = (1, 2, 3)

Y = (4, 5, 6)

$f(x) = (1, 2, 3, 2^2, 4, 6, 3, 6, 9)$

$f(y) = (16, 20, 24, 20, 25, 30, 30, 36)$

$\langle f(x), f(y) \rangle = 16 + 40 + 72 + 40 + 100 + 180 + 72 + 180 + 324 = 1024$

↓

lot of algebra, mainly because it is mapping from 3D to 4D

⇒ **With Kernel**

$K(x, y) = (4 + 10 + 18)^2 = 32^2 = 1024$

same result, but calculation is easier

(took dot product of x, y)

* $(1, 2, 3) \quad (4, 5, 6)$

$(1 \times 4) + (2 \times 5) + (3 \times 6)$

$(4 + 10 + 18)^2 = 32^2 = 1024$

* Suppose we have Imbalanced samples

$n = 10$

7 male, 3 female

So we give more weightage to less samples

(Eg - female ko 4 weightage de diya)

↳ Isse overfitting nhi hoti