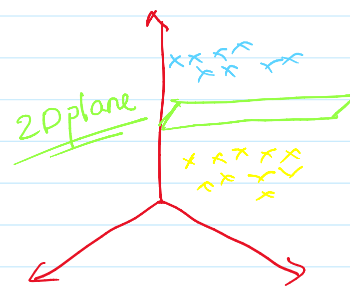
Support Vector Machines (Udemy KN)

13 June 2024 15:40

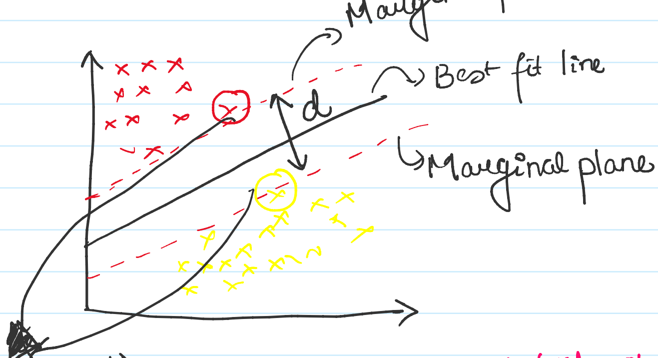
Support Vector Machine

① SVC - (Support Vector Classifier)

② SVR - (Support Vector Regressor)



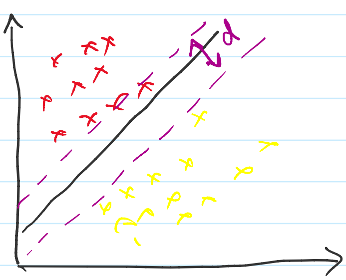
① SVC → Classifier



— They are both equidistant from best fit line

Aim is to ensure the 'd' is maximum

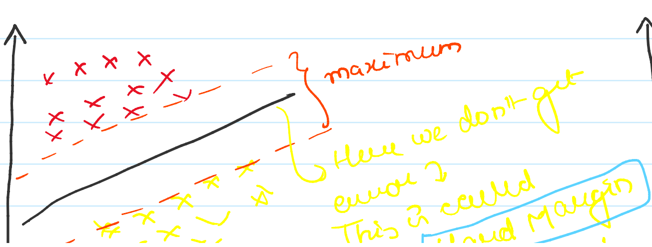
The nearest points to the best fit lines



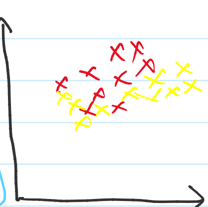
Which of the plane should we select?

- The plane with highest 'd' or distance between the marginal planes

=> Soft Margin and Hard Margin in SVM

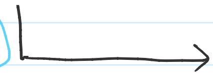


Here we don't get error ? This is called Hard Margin



This is called Soft Margin

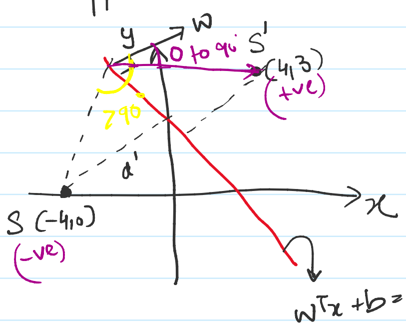
Here error? This is called Hard Margin



So here there cannot be any perfect separation and there will be errors. called Soft Margin

In this case we have a clear boundary but in real-life datapoints will be overlapping

⊛ Support Vector Machine Maths intuition



How → Eq^n of a straight line

$ax + by + c = 0$

↓

$w_1 x_1 + w_2 x_2 + b = 0$

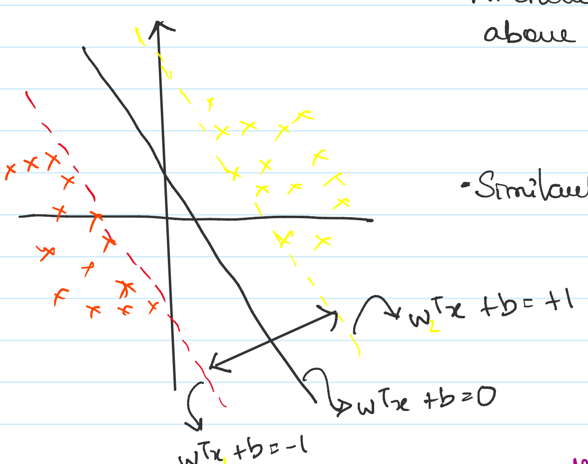
↓

$w^T x + b = 0$ → if b pass through origin ⇒ $\boxed{w^T x = 0}$

∴ Whenever for a point, the angle b/w $\vec{w}$ and the point connecting w to S, is greater than 90°, the distance of the point S is (-ve) from the plane. (points below line)

∴ In case it is b/w 0 to 90°, the distance of pt from line is (+ve.) (points above line)

d = -ve below plane
d = +ve above plane



- Whenever we go up from the hyperplane towards the pts. above it, we are going to get positive values always

$\therefore w^T x + b = +1$

- Similarly, if we go down the hyperplane, we get negative values.

$\therefore w^T x + b = -1.$

We need the distance b/w marginal planes

$$w^T x + b = +1$$
$$\underline{\overset{(-)}{} w^T x \underset{(-)}{+ b} = \underset{(+)}{-1}}$$
$$w^T(x_1 - x_2) = 2$$

A vector whose magnitude is 1

We convert the vector to the $\boxed{\text{unit vector}}$, divide the vector by its magnitude

$$\frac{w^T(x_1 - x_2)}{\|w\|} = \frac{2}{\|w\|} \rightarrow \text{Cost function}$$

distance b/w marginal planes

$\Rightarrow$ Maximize $\frac{2}{\|w\|}$
$\therefore$ w, b

under constraint
$$y_i = \begin{cases} +1 & w^Tx + b \geq 1 \\ -1, & w^Tx + b \leq -1 \end{cases}$$

Correctly classified points

For all correct points

constraint $\Rightarrow$ $\boxed{y_i * (w^Tx + b) \geq 1}$

SVC Cost function

$$\underset{w,b}{\text{maximize}} \frac{2}{\|w\|} \Rightarrow \underset{(w,b)}{\text{minimize}} \frac{\|w\|}{2}$$

final cost fn for SVC $\rightarrow$ $\boxed{\underset{w,b}{\min} \frac{\|w\|}{2}} + \boxed{C_i \sum_{i=1}^{n} \xi_i}$

Hinge Loss

$\Downarrow$

Soft Margin

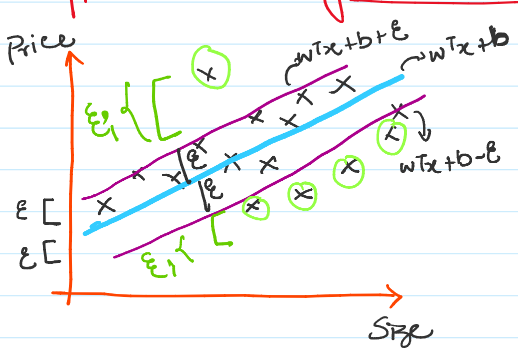
(because in soft margin we expect overlapping classes and misclassification)

How many points we want to avoid misclassification?

(Basically it means I can ignore C misclassified points)

Summation of the distance of the incorrect data points from the marginal plane

Support Vector Regressor (SVR)



$\varepsilon \rightarrow$ marginal error

Cost fn

Hinge Loss

$$\underset{w,b}{\text{Min}} \frac{\|w\|}{2} + \boxed{C_i \sum_{i=1}^{n} \xi_i}$$

The marginal error we take to accomodate the points outside marginal plane

Constraints

margin error

$$|y_i - w_i x_i| \leq \varepsilon + \xi$$

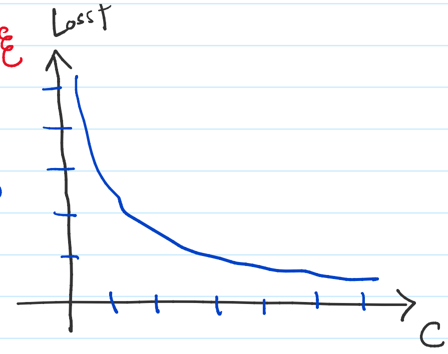
Loss fn

$$|y_i - w_i x_i| \leq \varepsilon + \xi$$

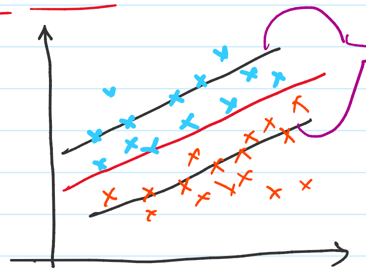
true — predicted — error above the margin

Basically all point should fall b/w marginal plane and the difference should be as low as possible

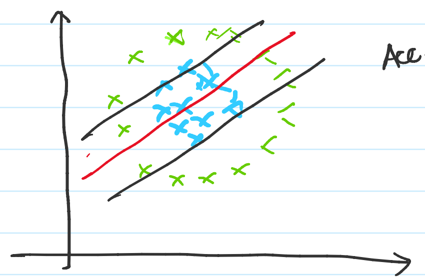
C↑↑
Loss fn ↓



SVM Kernels

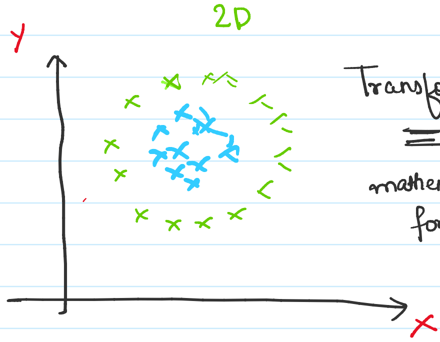


Linear SVC

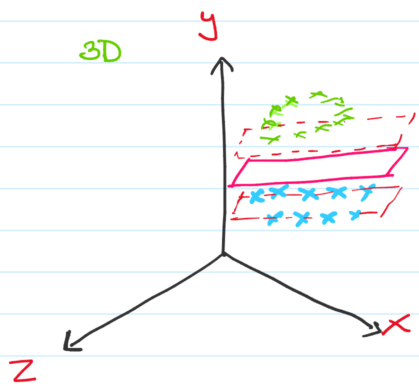


Acc↓↓ if we use Linear SVC

So in these scenarios, where our data is not linearly separable, we use SVM Kernels.



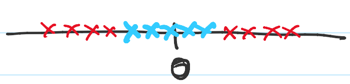
Transformation ⟹ mathematical formula



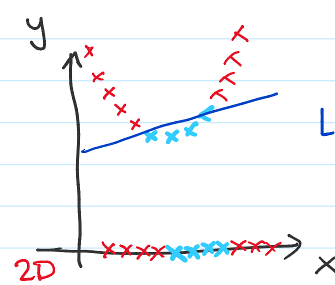
Create plane with linear SVC

Dataset : 1D

Cannot be solved by linear SVC



⟹ Transformation $y = x^2$



Acc ↑↑
Linear SVC

1. Polynomial Kernel
2. RBF Kernel
3. Sigmoid Kernel

} major SVM Kernels

Linear Kernel

used for text classification

Advantages: — Fastest
— Only one hyperparameter so can be easily obtained

Polynomial Kernel

formula: $(\underbrace{a \times b}_{\text{Observation from data}} + r)^d$ d = degree of polynomial

r → coefficient of polynomial

r, d can be found using cross validation

RBF Kernel

↳ Radial Basis function

formula: $e^{-\gamma(\underbrace{x-l}_{\text{2 data points}})^2}$

$\gamma \to$ scaling factor