# Session-5

$$\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$\left.\begin{array}{l} \alpha_i \geq 0 \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{array}\right]$$

$\alpha_i = 0$ for non SV

$\alpha_i > 0$ , S.V.

Let there be 5 SV.-

$$\max_{\alpha_i} (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5) - \frac{1}{2} \Big[ \_ + \_ + \_ \;\ldots\; + \Big]$$

25 terms

| i \ j | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | – | – | – | – | – |
| 2 | – | – | – | – | – |
| 3 | – | – | – | – | – |
| 4 | – | – | – | – | – |
| 5 | – | – | – | – | – |

→ Cosine similarity $= \dfrac{A \cdot B}{|A||B|}$

if magnitude is 1 then $A \cdot B$

So in ~~SV~~ dual form we have $x_i \cdot x_j$
can be called cosine similarity

So in dual form we are maximizing the similarity of SV based on there sign (due to $y_i$ & $y_j$)

we can use other similarity in place of $x_i \cdot x_j$

$$x_i \cdot x_j \Rightarrow sim(x_i, x_j)$$

here now we use kernal

· Kernel SVM

$$\max_{\alpha_i} \sum \alpha_i - \frac{1}{2} \sum\sum \alpha_i \alpha_j y_i y_j \, \underline{K(x_i, x_j)}$$

· Polynomial kernel -

$$K(x_i, x_j) = (r + x_i \cdot x_j)^d$$

d is degree

assume $r = 1$ & $d = 2$

| $x_1$ | $x_2$ |
|---|---|
| $x_{11}$ | $x_{12}$ |
| $x_{21}$ | $x_{22}$ |

$$(1 + x_{11} x_{21} + x_{12} \cdot x_{22})^2$$

$$= 1 + x_{11}^2 x_{21}^2 + x_{12}^2 \cdot x_{22}^2 + 2 x_{11} x_{21} + 2 x_{12} x_{22} + 2 x_{11} x_{21} x_{12} x_{22}$$

Now we can use this in svm eqn in place of $K(x_i, x_j)$.

· The above eqn can be written as a product of two vectors.

$$\left[1 \; x_{11}^2 \; x_{12}^2 \; \sqrt{2}x_{11} \; \sqrt{2}x_{12} \; \sqrt{2}x_{11}x_{21}\right]$$

$$\left[1 \; x_{21}^2 \; x_{22}^2 \; \sqrt{2}x_{21} \; \sqrt{2}x_{22} \; \sqrt{2}x_{12}x_{22}\right]$$

To get $\epsilon_i^n$ 1 there are two ways -

1) $x_i\ (x_{11}\ x_{12}) \xrightarrow{\text{transform}} x_i'\ (6d)$ ⟩ $x_i' \cdot x_j'$ = get exp.

$x_j\ (x_{21}\ x_{22}) \longrightarrow x_j'\ (6d)$

2) $x_i, x_j$ ⟩ $K(x_i, x_j)$ = exp.

In 1st we have to transform data in 6d space and then do the dot product its very time & space consuming

Rather we can just give $x_i$ & $x_j$ to kernel fn and it will give us the exp.
Thats why we call it a trick

- In circular or sphere kind of data the square term i.e $x_{11}^2\ x_{21}^2$, $x_{12}^2\ x_{22}^2$ is needed but if we have shapes like ~~conic~~ hyperbola or conic section then linear term $2x_{11}x_{21}$ etc are helpful.

• RBF kernal (Radial Basis fn) (like normal dist.)
↳ popular
↳ best out of the box kernal
(means if we don't know ~~what~~ which kernel to use then use this)

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

$$= e^{-\gamma\|x_i - x_j\|^2} \qquad \gamma = \frac{1}{2\sigma^2}$$

$||x_i - x_j||$ is euclidean dist

So

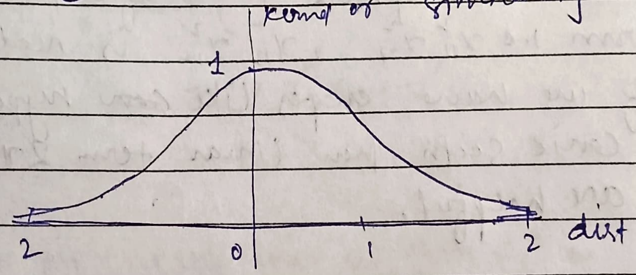
$$K \propto \frac{1}{dist}$$

Advantage -

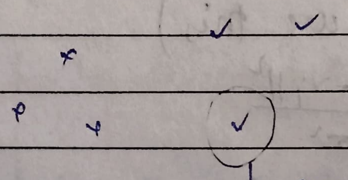
1) It can do Non-Linear transformation.
2) it can do Local ~~distance~~ decision
3) we can increase or decrease the value of $\gamma$ to increase or decrease complexity of decision boundary
4) Universal approximate property
   ↳ means it can approximate any continuous fn

- Local decision -

$$y = e^{-dist^2}$$



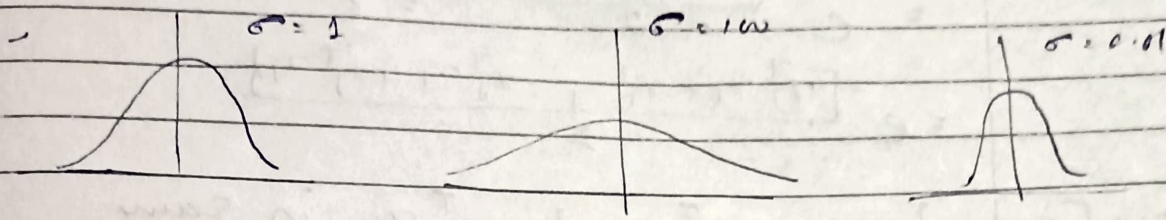

if $-2 < dist < 2$ then there is similarity, else no similarity exist.



↳ any pt lying within this region is similar to ✓

- Effect of $\gamma$ –

$$\gamma = \frac{1}{2\sigma^2}$$




If we increase value of $\sigma$ then width increase i.e locality increases it means it can be $-50 < dist < 50$

If we decrease $\sigma$ then width decreases so locality decreases.

now $\gamma \propto \frac{1}{\sigma}$   $\gamma \uparrow \rightarrow$ loc $\downarrow$
$\gamma \downarrow \rightarrow$ locality $\uparrow$

- $\sigma \downarrow \Rightarrow \gamma \uparrow \Rightarrow$ overfitting
  $\sigma \uparrow \Rightarrow \gamma \downarrow \Rightarrow$ underfitting

- $\gamma$ is hyperparameter.

- Relationship b/w RBF & polynomial kernel -

$x_1$ $x_2$ degree 2

in poly. $\Rightarrow$ $x_1^2$ $x_2^2$ $2x_1x_2$ $x_1$ $x_2$

in RBF $\rightarrow$ $x_1^2$ $x_2^2$ $x_1x_2$ $x_1,x_2$ $x_1^3$ $x_2^3$

---

in RBF we can make infinite dimension feature space, so we can ~~map~~ generate complex boundaries also

Page No.
Date

$$k(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2}}$$

$$= e^{-\frac{(x_i - x_j)^T (x_i - x_j)}{2}} = e^{-\frac{(x_i^T - x_j^T)(x_i - x_j)}{2}}$$

$$= e^{-\frac{[x_i^T x_i - x_i^T x_j - x_j^T x_i + x_j^T x_j]}{2}}$$

$x_i = [x_{11}\ x_{12}]$

$x_j = [x_{21} \cdot x_{22}]$

$x_j^T x_i$ & $x_i^T x_j$ is same

so,

$$e^{-\frac{1}{2}[x_i^T x_i + x_j^T x_j - 2 x_i^T x_j]}$$

$$= \underbrace{e^{-\frac{1}{2}[x_i^T x_i + x_j^T x_j]}}_{c} e^{x_i^T x_j}$$

$$c\, e^{x_i^T x_j}$$

$$c\, e^{1 + x_i^T x_j - 1}$$

$$= \underbrace{c e^{-1}}_{c'} e^{1 + x_i^T x_j}$$

$$c' e^{1 + x_i^T x_j}$$

$$= c' \sum_{k=0}^{\infty} \frac{(1 + x_i^T x_j)^k}{k!}$$

$$= c' \sum_{k=0}^{\infty} \frac{k \cdot \text{Poly}(x_i, x_j)^k}{k!}$$

from here we see that rbf kernel can map to any degree.

→ custom kernels

↳ we have many kernels that we can use & we can make our own also