Page No.
Date

RBF Kernel

- Radio Basis Function (RBF)

$e^{-\gamma (a-b)^2}$

→ This is the correct and standard form of the RBF (Gaussian) kernel used in SVMs, ~~expo~~ especially for 1-dimensional data

- a and b are two different data points (scalars or vectors)
- γ (gamma) is a parameter that determines how much influence a single training example has.
- The formula computes a similarity score: the closer a and b the closer the kernel value is to 1; the further apart, the closer is the value is to 0

→ For higher-dimensional data $(a-b)^2$ is replaced by the squared Euclidean distance i.e $\|a-b\|^2$

$K(x,y) = e^{-\gamma \|x-y\|^2}$

$\|x-y\|^2$ means:

$(x_1-y_1)^2 + (x_2-y_2)^2 + \ldots\ldots + (x_n-y_n)^2$

⇒ $k(x,y) = e^{-\gamma [(x_1-y_1)^2 + (x_2-y_2)^2 + (x_3-y_3)^2 + (x_4-y_4)^2]}$

[If I have 4 inputs :- x & y]

The RBF kernel is a way to measure how similar two data points are, based on their distance in the input space.

RBF :- Radio Basis Function (In-depth Math)

Step1: Create a New Dataset

Let's imagine data with 2 inputs (features) for each point and a class label:

Point	Input 1	Input 2	class
E	3	5	0
F	4	7	0
G	9	13	1
H	8	12	1

Step2 : Write down the RBF Kernel Formula

$\Rightarrow K(x, y) = e^{-\gamma \|x - y\|^2}$

Let's set $\gamma = 0.1$ for easy calculation

Step3 : Calculate the RBF Kernel value for two points

Let's compare Point E and F :

Step 3.1 : Write the vectors :

E : x =

F : y =

Step 3.2 : Subtract features :

Input 1 : $3 - 4 = -1$

Input 2 : $5 - 7 = -2$

Step 3.3 : Square each difference :

$(-1)^2 = 1$

$(-2)^2 = 4$

Step 3.4 : Add squared differences :

$1 + 4 = 5$

Step 3.5 : Multiply by ~~formula~~ gamma and put in formula :

$K(E,F) = e^{-0.1 \times 5} = e^{-0.5}$

$e^{-0.5} = 0.6065$

This result means :

E and F are "fairly similar", as kernel value is close to 1

Step 4 : Compare points that are far apart

- Compare E and G :
  - E : $3 - 9 = -6$
  - G : $5 - 13 = -8$
  - Squares : $(-6)^2 + (-8)^2 = 32 + 64 = 100$
  - Multiply by gamma : $0.1 \times 100 = 10$
  - Kernel : $K(E, G) = e^{-10} = 0.000045$

Result : E and G are "are not similar" as kernel value is almost 0

Step 5 : Try Points in the same class (G and H)

- G :- $9 - 8 = 1$
- H :- $13 - 12 = 1$
- Squares : $(1)^2 + (1)^2 = 1 + 1 = 2$
- Multiply by gamma : $0.1 \times 2 = 0.2$
- Kernel : $K(G, H) = e^{-0.2} = 0.8187$

Result : G and H are "very similar" as kernel value is close to 1

Step 6 : What Does SVM learn from these values ?

- High kernel value (close to 1) : Points are very similar
- Low kernel value (close to 0) : Points are very different

- SVM uses all these kernel values to learn a boundary that separates class 0 (E, F) from class 1 (G, H)

- With RBF kernel, this boundary can be curved - not just a straight line.

Step 7: Visualizing the Points

- If you plot all four points on a graph:
  - E (3, 5) and F (4, 7) will be close together (Class 0)
  - G (9, 13) and H (8, 12) will be close together (Class 1)
- The SVM, using the RBF kernel will easily draw a smooth boundary that separates two groups

Step 8: In Words

- RBF kernel measures how far two points are - closer means higher kernel value
- SVM uses this info to build a flexible, non-linear boundary for classification

★ Similar or Not (Range)

- 0.0 to 0.5 → not similar
- 0.51 to 1 → similar

⇒ If two points are similar it tends to place them on the same side of the decision boundary - so they're likely to be classified into the same class

⇒ If two points are not similar - its easier for SVM to separate them into different classes by drawing a boundary line between them.

Page No.
Date

RBF kernel

- RBF kernel separates data points based on it computes similarity scores between all pairs of data points using the RBF formula
- The SVM algorithm then uses these scores (for the full dataset) to find the boundary that best separates the classes

SVM kernel handles multiclass:

- SVM breaks down the problem into multiple binary classifications
- One-vs-One or One-vs-Rest

Every classifier it uses RBF kernel formula

$K(x, y) = e^{-\gamma \|x - y\|^2}$

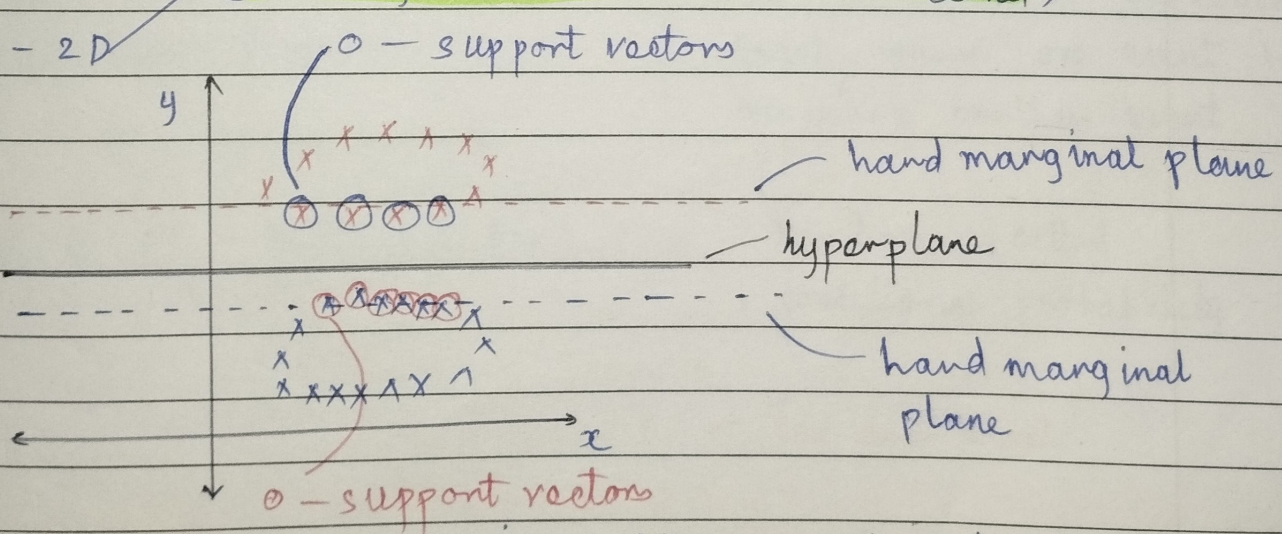
- it compares input features for all data points to measure similarity (same as binary)

By Default SVM is a binary classifier (can separate two classes at a time)

↳ for multiple class or multiclass :- SVM algorithm uses technique (like one-vs-one or one-vs-rest) to efficiently handle more than two classes

(for Polynomial kernel and RBF kernel)

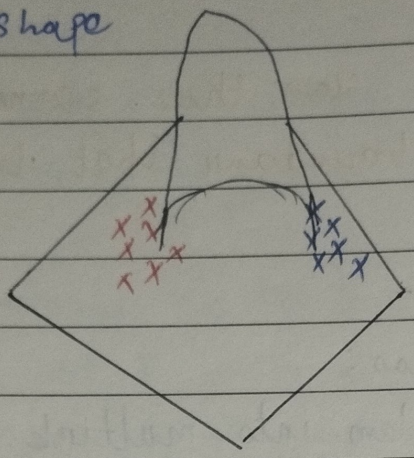
* 1D - 2D



1Dimensional to 2D imensional (Polynomial and RBF kernel)

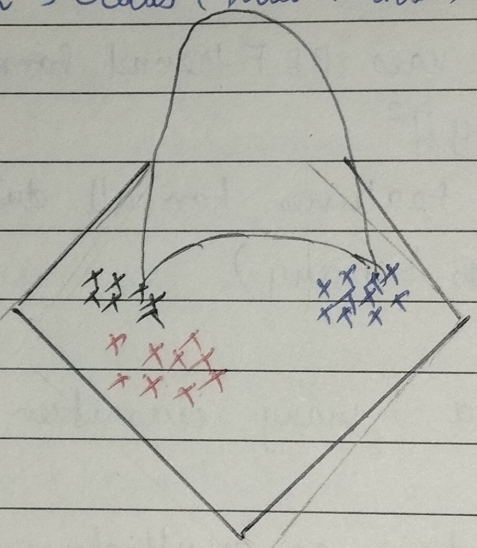
① RBF Kernel with 2 class (Data Points)

→ Bell curve shape



② RBF Kernel with 3 class (Data Points)

→ Bell curve shape



⇒ Each class data points tends to form clusters (or round shapes) in different regions of the plot