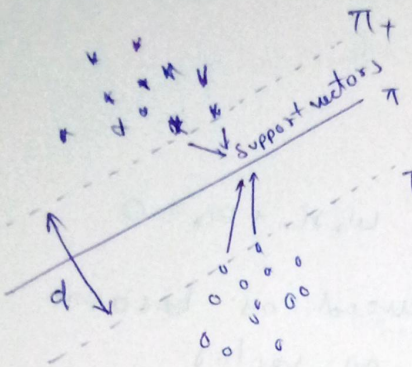
21-Dec-2023

Support Vector Machine



find ~~कोई~~ hyperplane जो margin को maximize करती है

margin = d = Shortest distance b/w $\pi+$ and $\pi-$

margin - maximizing hyperplane

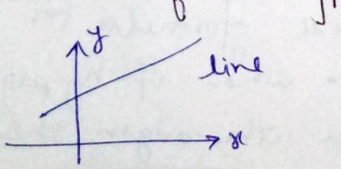
$w^T x + b = 0$

find the value of w and b where d is maximum.

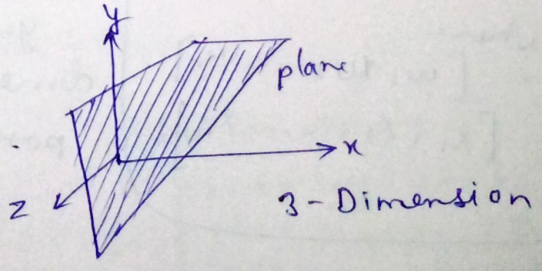
π = hyperplane

Advantages
- SVM is very Robust to outlier. (handle outlier)
- also work on non-linear data (using kernel)
- implied in both classification and regression both

equation of a hyperplane:



2-Dimension

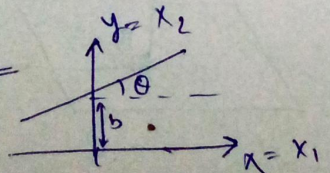


3-Dimension

in 4D, 5D, 6D it is hyperplane.

- most of the dataset in ML are more than the 3 features

in 2D



$y = mx + b$

general form

$ax + by + c = 0$

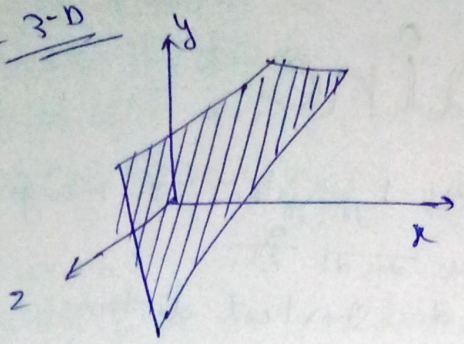
where $y = \frac{-ax}{b} - \frac{c}{b}$ and $m = \frac{-a}{b}$, $b = \frac{-c}{b}$

$ax + by + c = 0$

or $ax_1 + bx_2 + c = 0$

or $w_1 x_1 + w_2 x_2 + w_0 = 0$

in 3-D



$w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0 = 0$

in n-D

$w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_n x_n + w_0 = 0$

In the above equation can be reduced as because the given points can be used as vector.

dot product

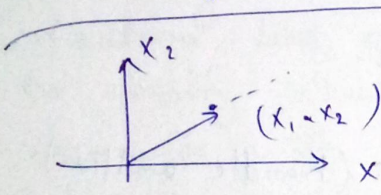
$w \cdot x + w_0 = 0$

$$[w_1\ w_2\ \cdots\ w_n] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + w_0 = 0$$

$$w^T x + w_0 = 0$$

generalize formula for any dimension. when if hyperplane pass from the origin then $w_0 = 0$

then $w^T x = 0$



$w_1 x_1 + w_2 x_2 = 0$

dot product of

$w \cdot x = w_1 x_1 + w_2 x_2$

where

$w = [w_1\ w_2 \cdots w_n]$

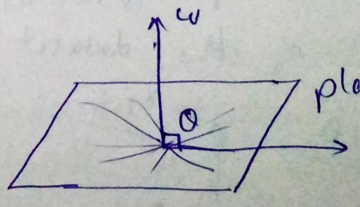
$x = [x_1, x_2, \cdots x_n]$

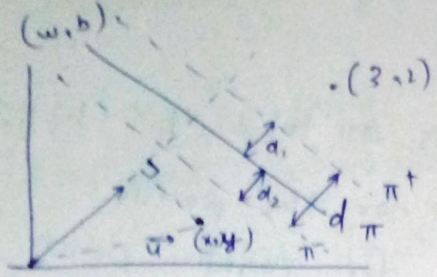
$$w^T x = w \cdot x = \|w\| \|x\| \cos\theta = 0$$

if $\theta = 90°$

it means by diagram

w is $\perp^r$ to the plane





$\vec{w} \cdot \vec{u} \geq c$

$\vec{w} \cdot \vec{u} - c \geq 0$

$\vec{w} \cdot \vec{u} + b \geq 0$

decision rule

example $2x + 3y + 3 = 0$

$w_1 x_1 + w_2 x_2 + 3 = 0$

$\vec{w} = (2, 3)$ $\quad \vec{u} = (3, 2)$

$2(3) + 3(2) + 3 > 0$

then $(3, 2)$ is lies in +ve side

similarly for $\vec{u} = (-2, 0)$ point is lies in -ve side

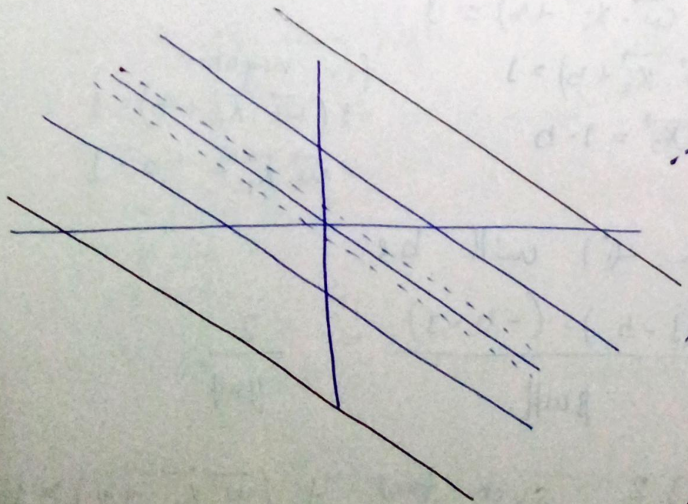
for any x_i

$$\hat{y} = \begin{cases} +1 & \text{if } \vec{w} \cdot \vec{x_i} + b \geq 0 \\ -1 & \text{if } \vec{w} \cdot \vec{x_i} + b < 0 \end{cases}$$

assume that

for π_+: $w^T x + b = 1$

for π^-: $w^T x + b = -1$



Let us example in 2-D

$$\begin{cases} 2.0x + 3.0y + 3.0 = 10 \\ 2.0x + 3.0y + 3.0 = 0 \\ 2.0x + 3.0y + 3.0 = -10 \end{cases}$$

$$\begin{cases} 2x + 3y + 3 = 1 \\ 2x + 3y + 3 = 0 \\ 2x + 3y + 3 = -1 \end{cases}$$

$$\begin{cases} 0.2x + 0.3y + 0.3 = 1 \\ 0.2x + 0.3y + 0.3 = 0 \\ 0.2x + 0.3y + 0.3 = -1 \end{cases}$$

Defining constraints of SVM

$$\begin{cases} \vec{w}\cdot\vec{x} + b \geq 1 \\ \vec{w}\cdot\vec{x} + b \leq 1 \end{cases}$$

If the red are points and label $y_i = +1$, $y_i = -1$

$y_i(\vec{w}\cdot\vec{x_i} + b) \geq 1$ ⟶ for +ve points

$y_i(\vec{w}\cdot\vec{x_i} + b) \geq 1$ ⟶ for negative points

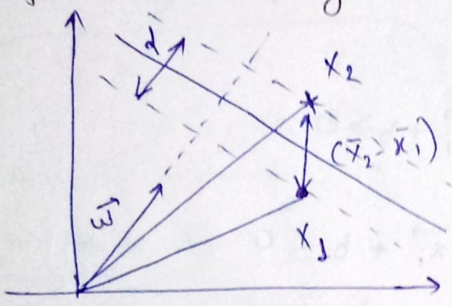
our final constraint is

$$y_i(\vec{w}\cdot\vec{x_i} + b) \geq 1$$

for support vectors

$$y_i(\vec{w}\cdot\vec{x_i} + b) = 1$$

find max margin in SVM:



Now unit vector of $w = \frac{\vec{w}}{\|w\|}$

distance vector b/w x_1 & x_2 = $(\vec{x_2} - \vec{x_1})$

$$d = (\vec{x_2} - \vec{x_1}) \cdot \frac{\vec{w}}{\|w\|}$$

$$d = \frac{\vec{x_2}\cdot w - \vec{w}\cdot x_1}{\|w\|} \quad \text{---(i)}$$

$\because y_i \cdot (\vec{w}\cdot\vec{x_i} + b) = 1$

for +ve $1(\vec{w}\cdot\vec{x_2} + b) = 1$

$\vec{w}\cdot\vec{x_2} = 1 - b$

for negative $-1(\vec{w}\cdot\vec{x_1} + b) = 1$

$\vec{w}\cdot\vec{x_1} = -b - 1$

hence equation (i) will be

$$d = \frac{(1 - b) - (-b - 1)}{\|w\|} = \frac{2}{\|w\|}$$

Hence $\text{argmax}\,(w^*, b^*)\frac{2}{\|w\|}$ such that $y_i(\vec{w}\cdot\vec{x_i} + b) \geq 1$

always true for every point given

all the above calculation and finding the maximum margin are called it Hard margin SVM

Soft Margin SVM:

constraint optimization problem

$$\max f(x) \Leftrightarrow \min \frac{1}{f(x)}$$

$$C \propto \frac{1}{\lambda}$$

Hence,

$$d = \underset{(w^*, b^*)}{\operatorname{argmin}} \frac{\|w\|}{2} + C \sum_{i=1}^{n} \varepsilon_i$$

→ Hinge loss

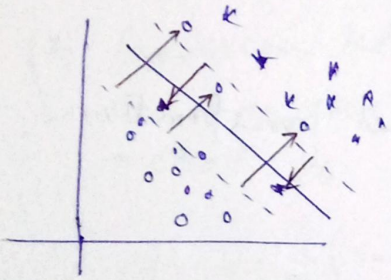
regularisation

Here, ~~the above formula is Regularisation~~

C = hyper parameter

ε = zeta =

if all points are correctly classified. then $\sum_{i=1}^{n} \varepsilon_i = 0$

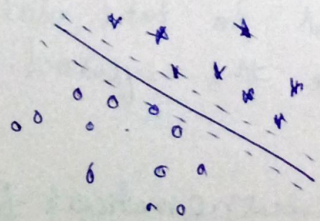


$\sum_{i=1}^{n} \varepsilon_i$ is the sum of all the distance of incorrectly classified point

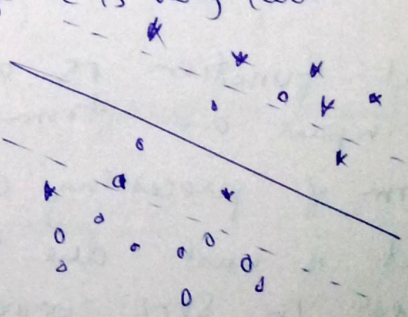
Hard in SVM

Error = Margin error + classification error

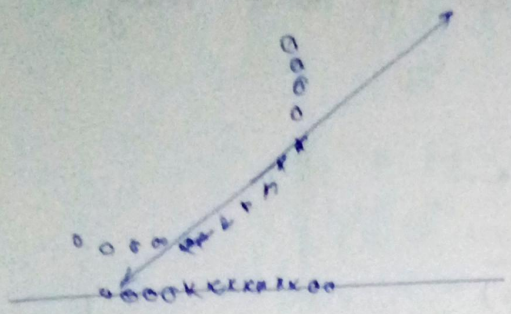
if C is very high.

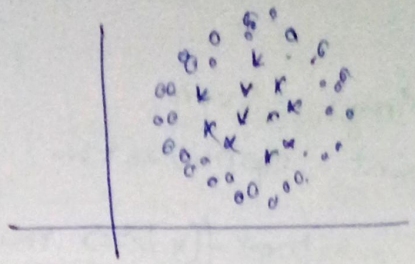


if C is very low



Kernel trick in SVM

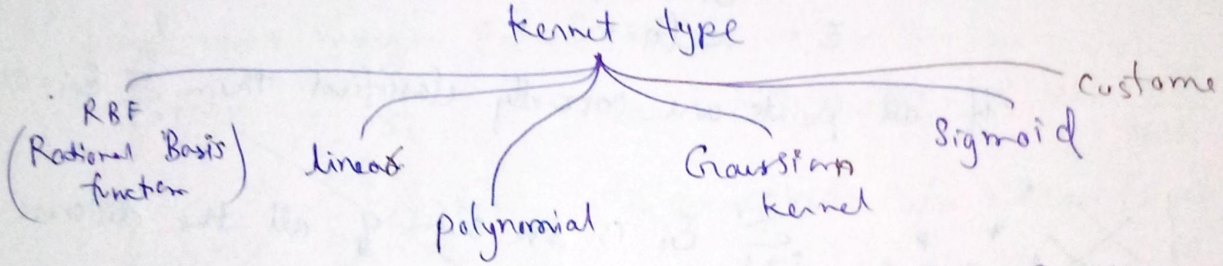




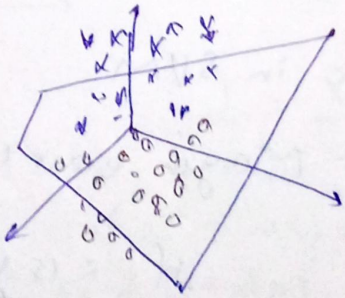
Non-linear data

(Low-dimension → f(x) → higher dimension

kernel



- The transformation is called kernel transformation.



in 3-D

- Kernel function is a method used to take data as input and transform it into the required form of processing data.
- Kernel is used due to a set of mathematical function used in SVM providing the window to manipulate the data.
- So kernel function generally transforms the training set of data so that a non-linear

decision surface is able to transform to a linear equation in higher number of dimension spaces.

- Basically ϕ returns the inner product b/w two point in a standard feature dimension.

standard kernel function equation

$$K(\vec{x}) = \begin{cases} 1 & \text{if } \|x\| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

(i) Gaussian kernel:

- it is used to perform transformation when there is no prior knowledge

$$K(x, y) = e^{-\left(\frac{\|x-y\|^2}{2\sigma^2}\right)}$$

(ii) Gaussian kernel Radial Basis function: (RBF):

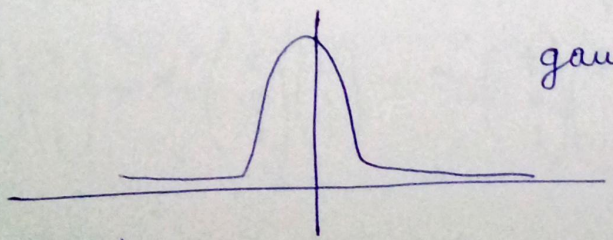
- Same as above kernel function, adding radial basis method to improve the transformation.

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

$K(x, x_1) + K(x, x_2)$ (Simplified - formula)

$K(x, x_1) + k(x, x_2) > 0$ (Green)

$K(x, x_1) + k(x, x_2) = 0$ (Red)



gaussian kernel graph

code:

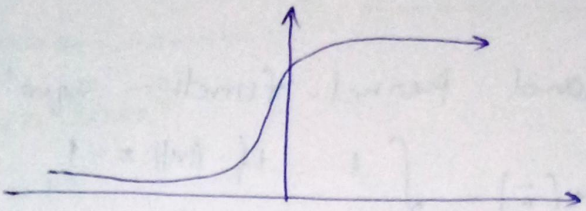
```
from sklearn.svm import svc
classifier = svc(kernel = 'rbf', random_state = 0)
classifier.fit(x_train, y_train)
```

* if γ is very high then the model overfits the training data

(iii) Sigmoid kernel.

- This function is equal to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial neurons

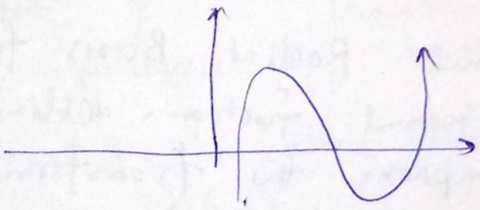
$$K(x, y) = \tanh(\gamma . x^T y + \gamma)$$



(iv) polynomial kernel:

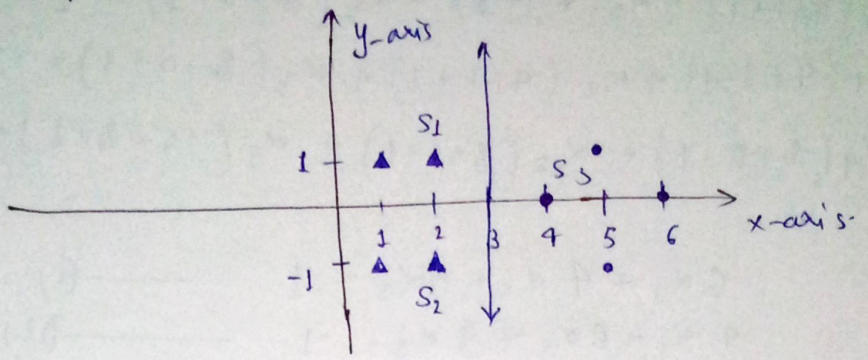
- It represents the similarity of vector in the training set of data in a feature space over polynomials of the original variables used in the kernel.

$$K(x, y) = \tanh(\gamma . x^T y + \gamma)^d \quad . \quad \gamma > 0$$



(v) linear kernel: used when data is linearly separable.

Question Find the ~~pot~~ optimal hyper-plane for following. data points negatively labelled data points. (1,2), (2,1), (2,-1), (2,-1) and positively labelled data points (4,0), (5,1), (5,-1), (6,0)



Step-1 find the support vectors

$$S_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad S_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \quad S_3 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$$

Step-2. finding augumented vector

$$S_1 = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \quad S_2 = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \quad S_3 = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$

Step 3 finding expressions.

$$\alpha_1 \bar{S}_1 \cdot \bar{S}_1 + \alpha_2 \bar{S}_2 \cdot \bar{S}_1 + \alpha_3 \bar{S}_3 \cdot \bar{S}_1 = -1$$

$$\alpha_1 \bar{S}_1 \cdot \bar{S}_2 + \alpha_1 \bar{S}_2 \cdot \bar{S}_2 + \alpha_3 \bar{S}_3 \cdot \bar{S}_2 = -1$$

$$\alpha_1 \bar{S}_1 \cdot \bar{S}_3 + \alpha_2 \bar{S}_2 \cdot \bar{S}_3 + \alpha_3 \bar{S}_3 \cdot \bar{S}_3 = 1$$

put the value of in above equation

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 9 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = -1$$

Now,

$$\alpha_1(9+1+1) + \alpha_2(9-1+1) + \alpha_3(8+0+1) = -1$$
$$\alpha_1(9-1+1) + \alpha_2(9+1+1) + \alpha_3(8+0+1) = -1$$
$$\alpha_1(8+0+1) + \alpha_2(8+0+1) + \alpha_3(16+8+1) = -1$$

$$6\alpha_1 + 4\alpha_2 + 9\alpha_3 = -1 \quad \text{——(i)}$$
$$4\alpha_1 + 6\alpha_2 + 9\alpha_3 = -1 \quad \text{——(ii)}$$
$$9\alpha_1 + 9\alpha_2 + 17\alpha_3 = 1 \quad \text{——(iii)}$$

By solving the above equation we get.

$$\alpha_1 = \frac{-13}{4} = -3.25 \; ; \; \alpha_2 = \frac{-13}{4} = -3.25 \; ; \; \alpha_3 = \frac{7}{2} = 3.5$$

Step. 4 Calculate the weight vector:

$$w = \sum \alpha_i \vec{s_i}$$
$$w = \alpha_1 \vec{s_1} + \alpha_2 \vec{s_2} + \alpha_3 \vec{s_3}$$
$$(w,b) = -3.25 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + -3.25 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + 3.5 \begin{pmatrix} 9 \\ 0 \\ 1 \end{pmatrix}$$
$$(w,b) = \begin{pmatrix} -6.5 + 6.5 + 19 \\ -3.25 + 3.25 + 0 \\ -3.25 + -3.25 + 3.5 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

$$(w,b) = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

Step. 5 Hyperplane equation

$$y = mx + b$$
$$y = wx + b$$

$w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $b = -3$

Hence hyper plane line will pass through at $x = 3$ which will be parallel to y-axis.

1. What are kernels in SVM? Can you list some popular SVM kernels?

Description:
Kernel in SVM are functions that transform data into a higher-dimensional space, making it easier to find a hyperplane (decision boundary) that separates data points. These functions enable SVMs to handle non-linear data by projecting it into higher dimensions, essential for capturing complex patterns.

Advantages:
Kernels are versatile and effective in capturing complex patterns, allowing SVMs to handle non-linear data. They transform data to enable the separation of complex patterns.

Disadvantage:
Choosing the right kernel and its parameters can be challenging and training with complex kernel can be computationally expensive. proper kernel selection is crucial.

Numerical example:
In a binary classification problem, an RBF kernel projects data into a higher-dimensional space, making it separable by a hyperplane and allowing the SVM to find a non-linear decision boundary.

Popular SVM kernels:
(i) linear kernel: computes the dot product in the original space, suitable for linearly separable data.

(ii) Polynomial kernel: Raises the dot product to a power, introducing non-linearity and is suitable for data with polynomial boundaries.

(iii) Radial Basis Function (RBF) kernel:
uses a Gaussian-like function to capture complex non-linear relationships, and is widely used and versatile.

(iv) Sigmoid kernel:
Applies the hyperbolic tangent function to the dot product. useful for data with sigmoid-shaped boundaries.

(v) Custom kernels:
Custom kernels can be defines based on domain knowledge or problem-specific features.