Support Vector Machines
&
Kernels

Anil Kumar M P

MTech CSE (DS & AI) PT

Semester 2

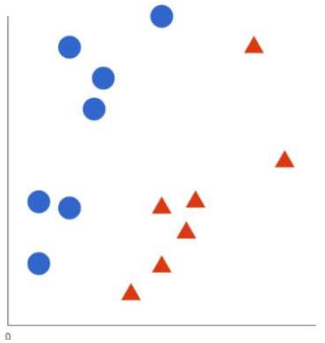
Support Vector Machines

- **Support Vector Machine** is a **Supervised Machine Learning** Algorithm
- SVM can be used for both **Classification** and **Regression** Tasks. But more prefers for Classification Tasks
- Support Vector Machines may be used for both **linear separable** and **non-linear separable** data
- The objective of SVM is to find a **hyperplane** that **maximally separates** the different classes in the training data

Types of SVMs

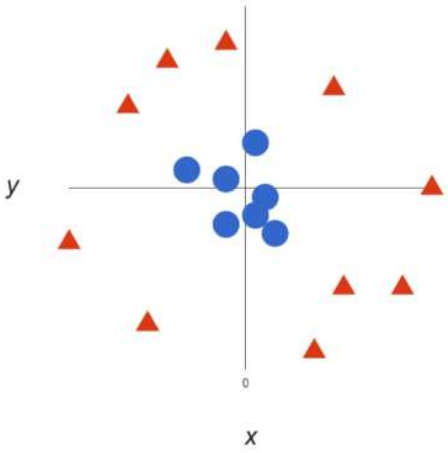
Simple (Linear) SVM

- A linear SVM refers to the SVM type used for classifying linearly separable data.
- This implies that when a dataset can be segregated into categories or classes with the help of a single straight line



Kernel (Non-Linear) SVMs

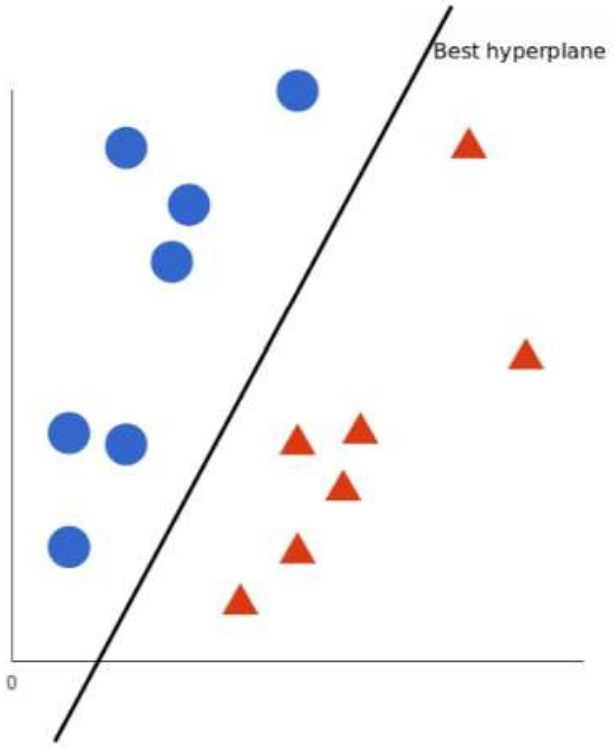
- Non-linear data that cannot be segregated into distinct categories with the help of a straight line is classified using a kernel or non-linear SVM



Hyperplane

- Hyperplane is plane that classifies the data into multiple classes
- Hyperplane may be 2D, 3D… (n-1) Dimensional etc depending on the number of dimensions of the data

No of Features (No of Dimensions)	Feature Representation	Hyperplane
1	(x_1)	Point (Zero Dimension)
2	(x_1,x_2)	Line (1 Dimensional)
3	(x_1,x_2,x_3)	2D Plane (2 Dimensional)
n	$(x_1,x_2,x_3,....x_n)$	(n-1) Dimensional



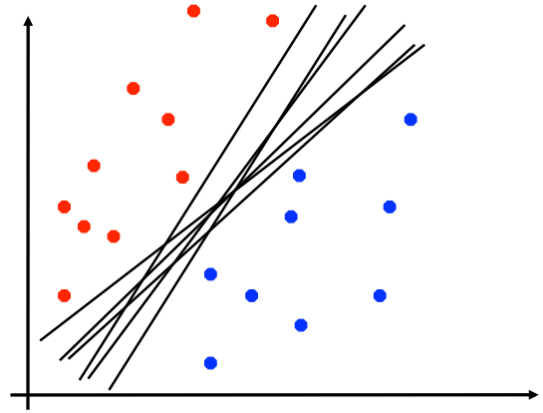
Margin & Support Vectors

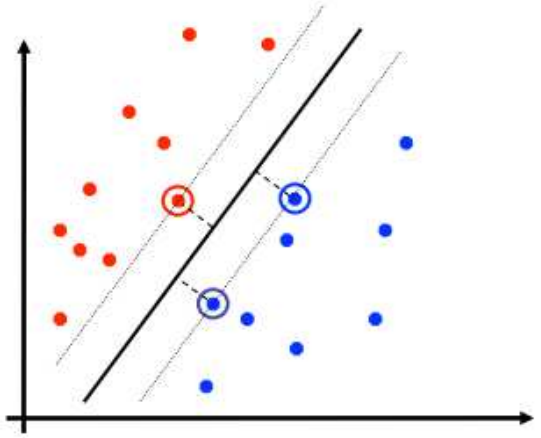
Margin: The distance between the Hyperplane and the closest data points from each class

Optimal Separating Hyperplane: The hyperplane with the maximum margin is called the Optimal Separating Hyperplane

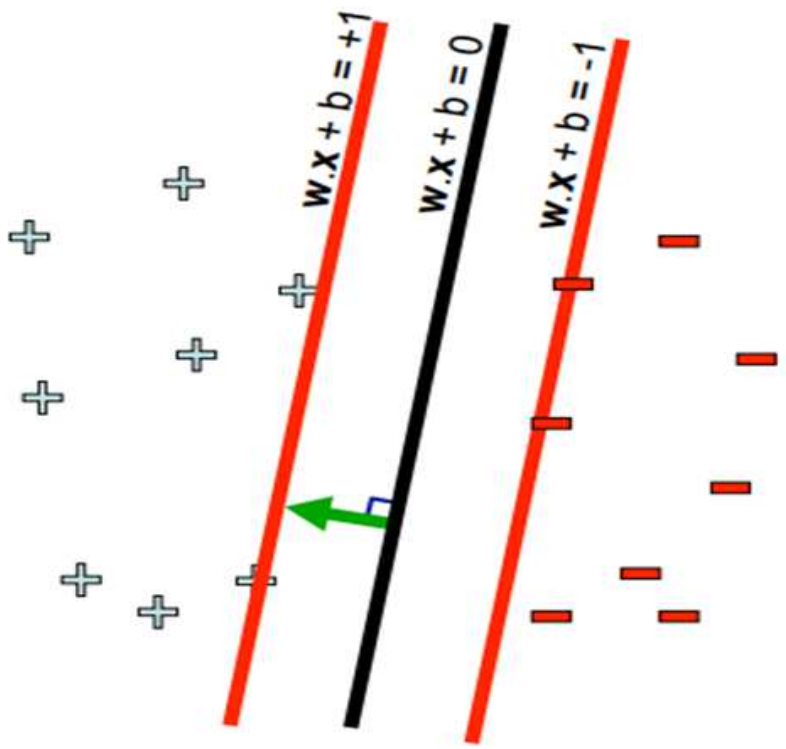
Support Vectors: The data points that lies closest to the optimal separating hyperplane are called Support Vectors

SVM Chooses the linear separator with the **largest margin**





Mathematical Formulation of SVM



For every data point (x_t, y_t), enforce the constraint

for $y_t = +1$, $w \cdot x_t + b \geq 1$

and for $y_t = -1$, $w \cdot x_t + b \leq -1$

Equivalently, we want to satisfy all of the linear constraints

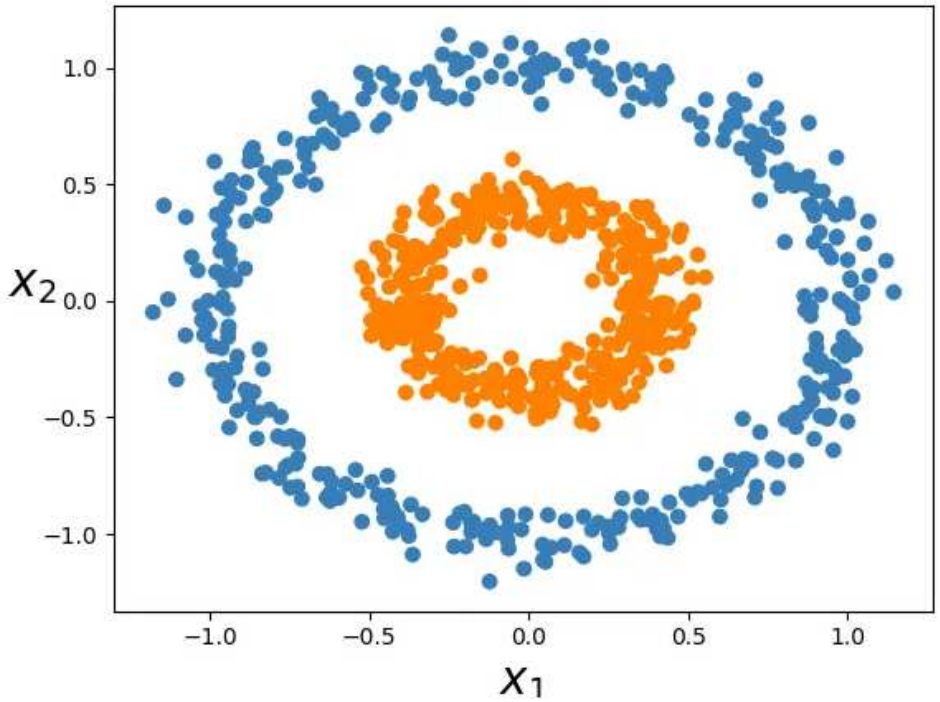
$$y_t \, (w \cdot x_t + b) \geq 1 \quad \forall t$$

Then the margin is $\frac{1}{||w||}$

or Minimizing $\min_{w,b} \frac{1}{2} ||w||^2$ as per Quadratic Programming

Kernel (Non-Linear) SVMs

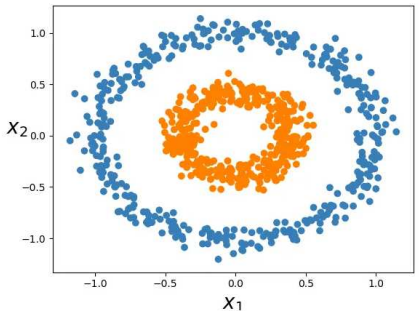
- **Kernel Machines** are a class of algorithms for Pattern Analysis
- The methods involve using linear classifiers to solve non-linear problems
- The general task of Pattern Analysis is to find and study general types of relations (eg: Clustering, Correlations, Classifications etc) in datasets



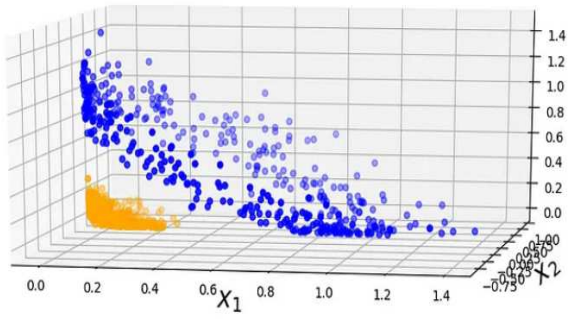
Non-Linear Separable Data

Kernel (Non-Linear) SVMs

- If the data is not linearly separable in the original, or input, space then we apply transformations to the data, which map the data from the original space into a higher dimensional feature space.
- The goal is that after the transformation to the higher dimensional space, the classes are now linearly separable in this higher dimensional feature space.
- We can then fit a decision boundary to separate the classes and make predictions.
- The decision boundary will be a hyperplane in this higher dimensional space.



after Data Transformations



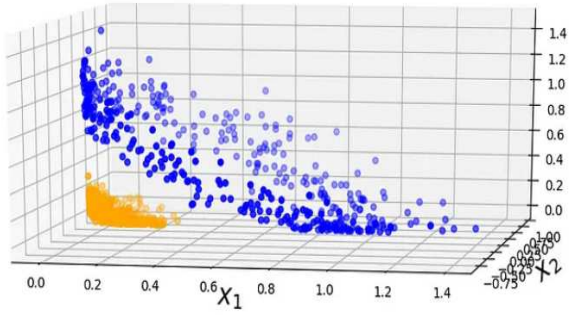
Kernel Functions

A function that takes as its inputs vectors in the original space and returns the dot product of the vectors in the feature space is called a *kernel function*

More formally, if we have data $\mathbf{x}, \mathbf{z} \in X$ and a map $\phi : X \rightarrow \Re^N$ then

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$$

is a kernel function



Kernel Trick: The "Kernel trick" is that the kernel methods represent the data only through a set of pairwise similarity comparisons between the original data observations x

$$\phi(\mathbf{a})^T \cdot \phi(\mathbf{b}) = \begin{pmatrix} a_1^2 \\ \sqrt{2}\, a_1 a_2 \\ a_2^2 \end{pmatrix}^T \cdot \begin{pmatrix} b_1^2 \\ \sqrt{2}\, b_1 b_2 \\ b_2^2 \end{pmatrix} = a_1^2 b_1^2 + 2 a_1 b_1 a_2 b_2 + a_2^2 b_2^2$$

$$= (a_1 b_1 + a_2 b_2)^2 = \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}^T \cdot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)^2 = \left(\mathbf{a}^T \cdot \mathbf{b} \right)^2$$

Thank You