Chapter 4

Support Vector Machines

SVM

SVC
Support vector classifier

Classification

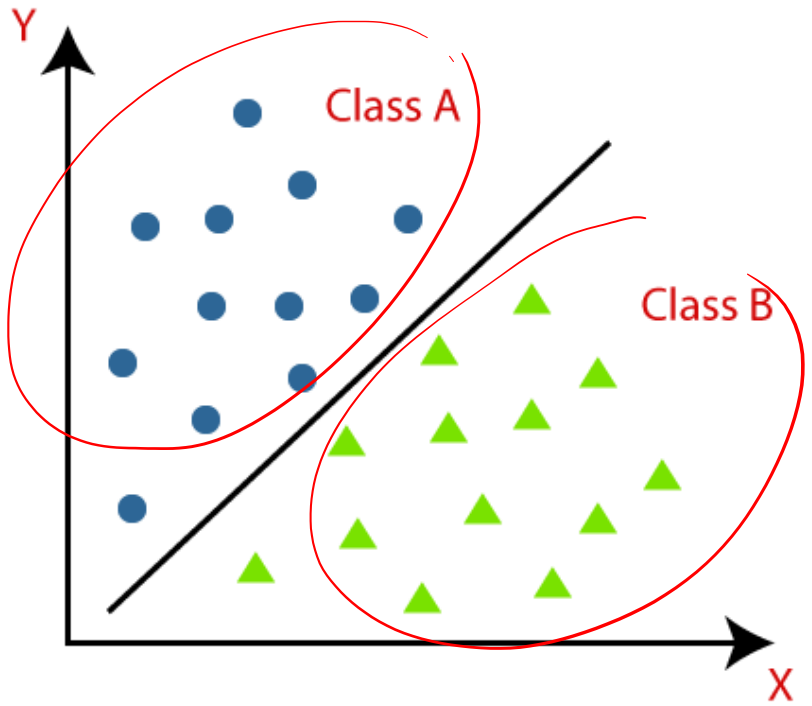
- **Classification** is an important tool in ML.
  - *Determines to which* ***class*** *an observation belongs*

class 0
class 1

class A
class B .

Binary Classification

- ***Binary Classification***
  - ⇒ 2 classes



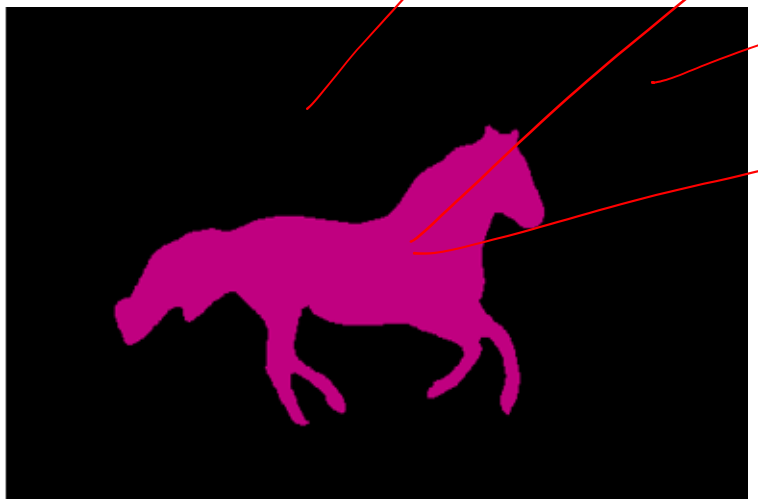
Binary Classification

- ***Binary Classification***
  - ⇒2 Classes

Binary Classification
• Binary Classification
• ⇒2 Classes

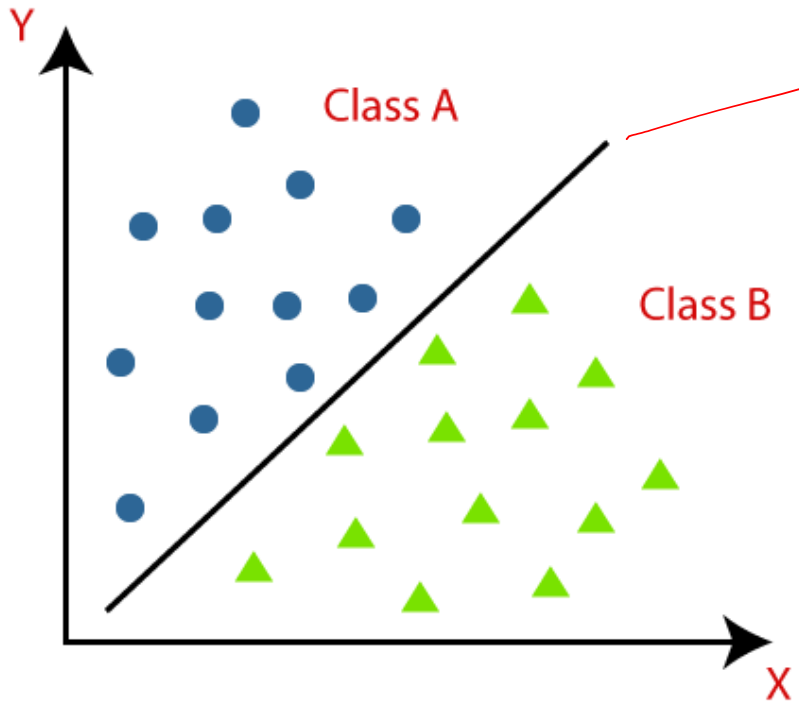
Applications

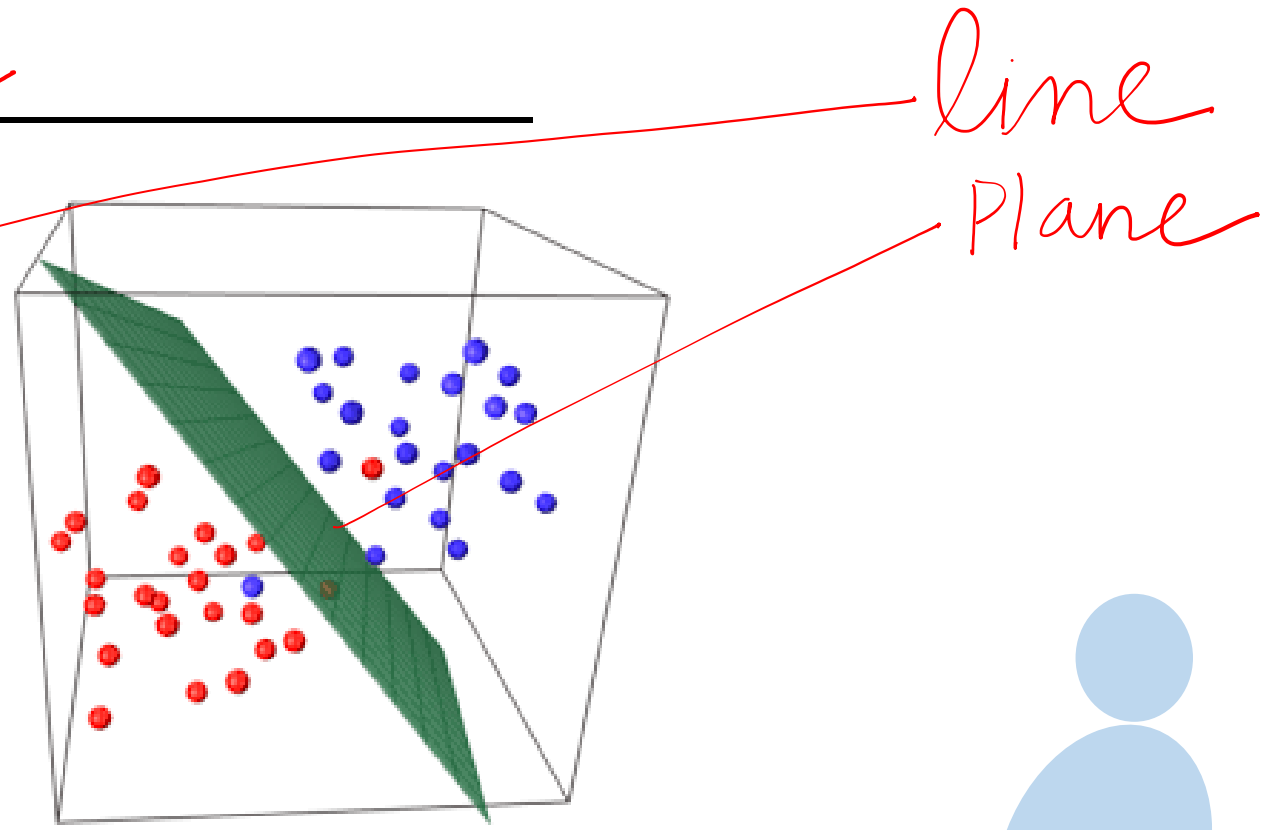
- **Image segmentation**
  - Classify pixels as belonging to **foreground** or **background**



Linear classifier

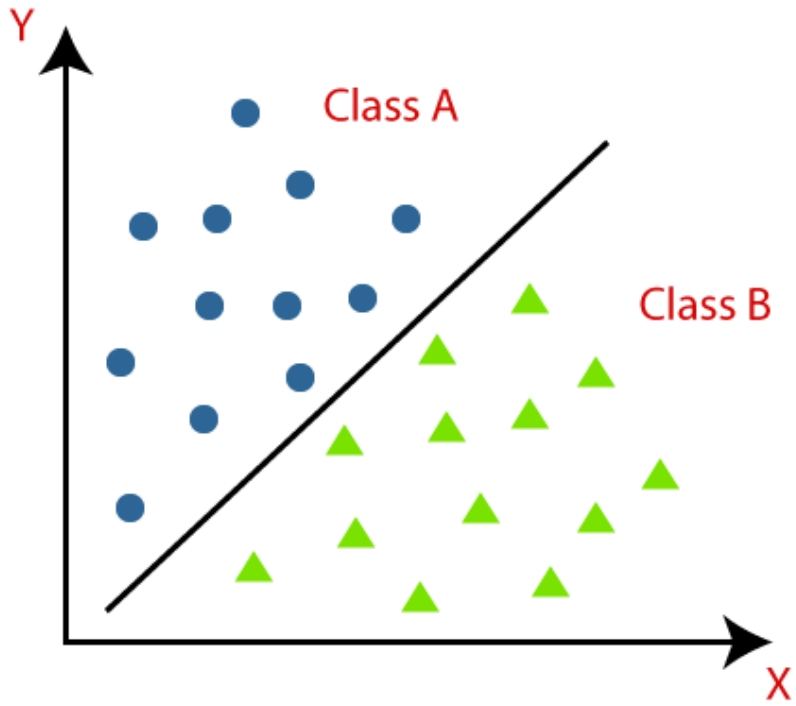
- Linear classifier corresponds to a
- 2D: Line
- 3D: Plane

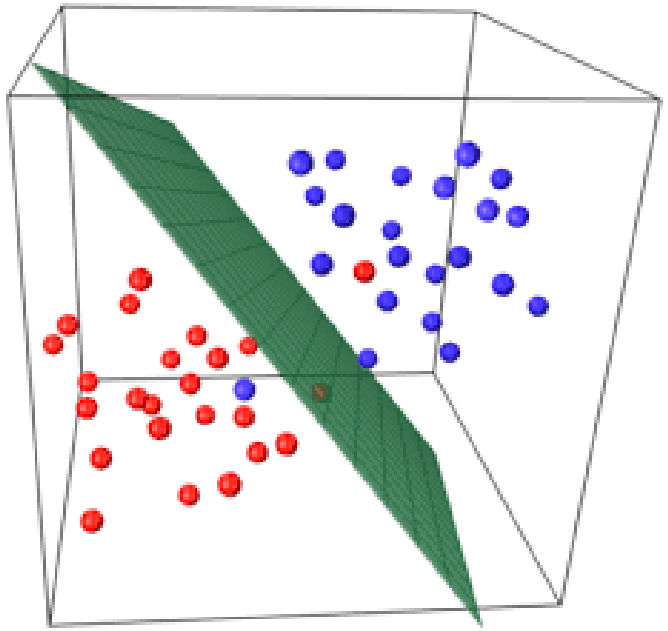




Linear classifier

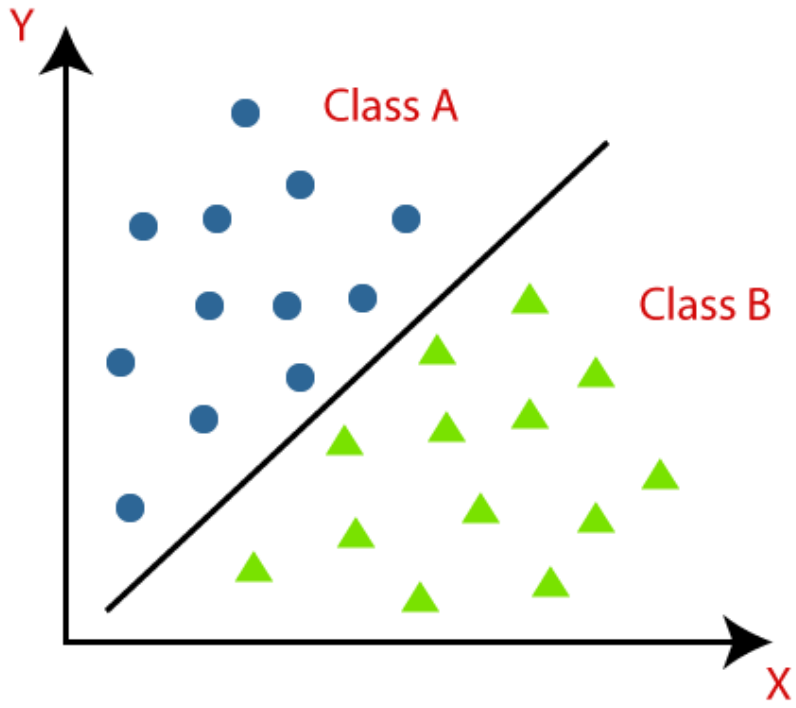
- Linear classifier corresponds to a
- 2D: line
- 3D: plane





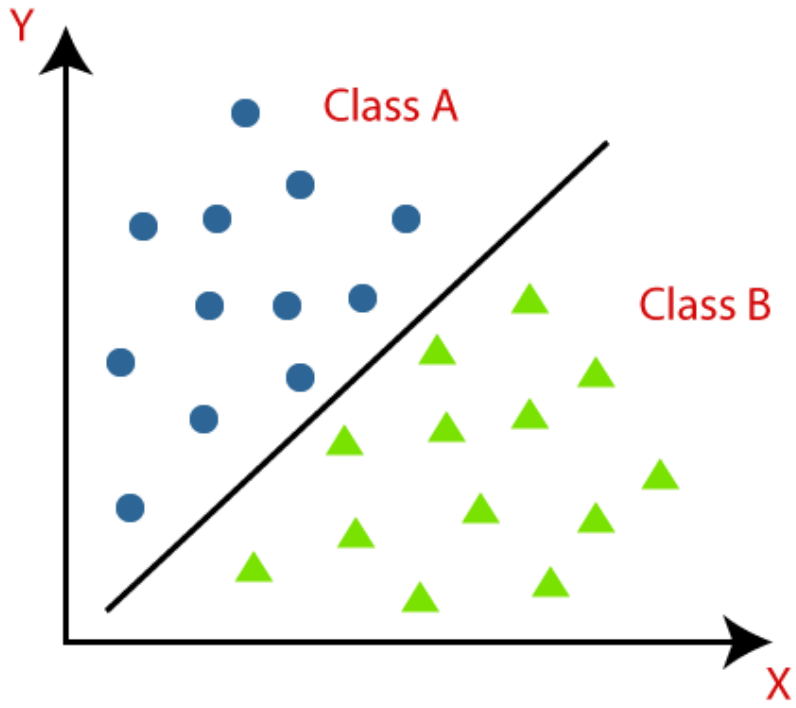
Linear classifier

- Linear classifier corresponds to a hyperplane in N dimensions
  - Easy to determine and analyse!



Linear classifier

- Linear classifier corresponds to a **hyperplane** in N dimensions
  - Easy to determine and analyse!



Linear classifier

$\bar{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}$

- General structure of a hyperplane is

$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$

2D line: $a_1x_1 + a_2x_2 = b$

$2x - 5x_2 = 7$

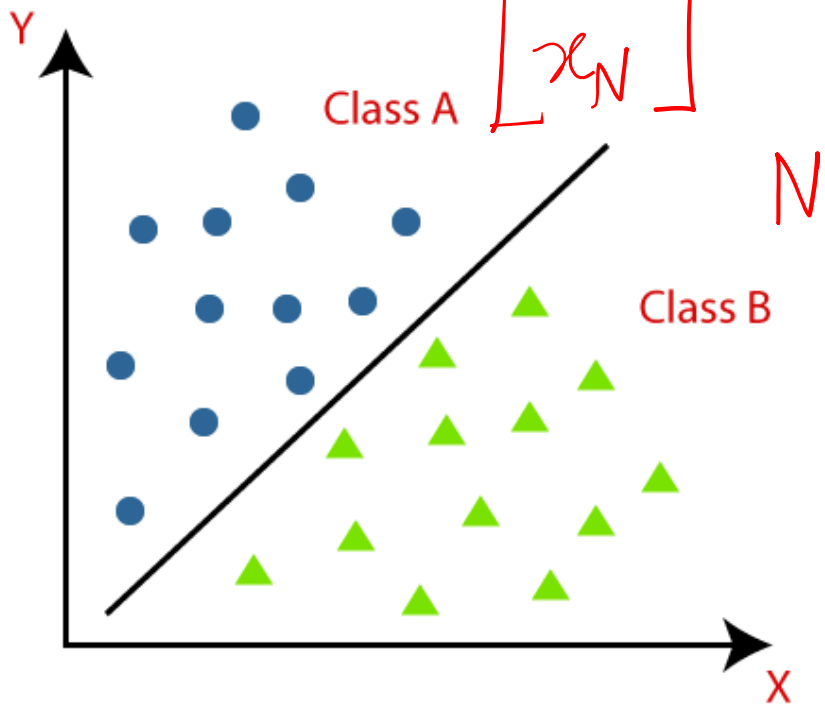
3D Plane: $a_1x_1 + a_2x_2 + a_3x_3 = b$

constant

$3x_1 - \frac{1}{2}x_2 + \frac{5}{7}x_3 = \sqrt{2}$

NDim: $a_1x_1 + a_2x_2 + \cdots + a_Nx_N = b$

$\bar{a}^T\bar{x} = b$



For different values of b these are parallel.

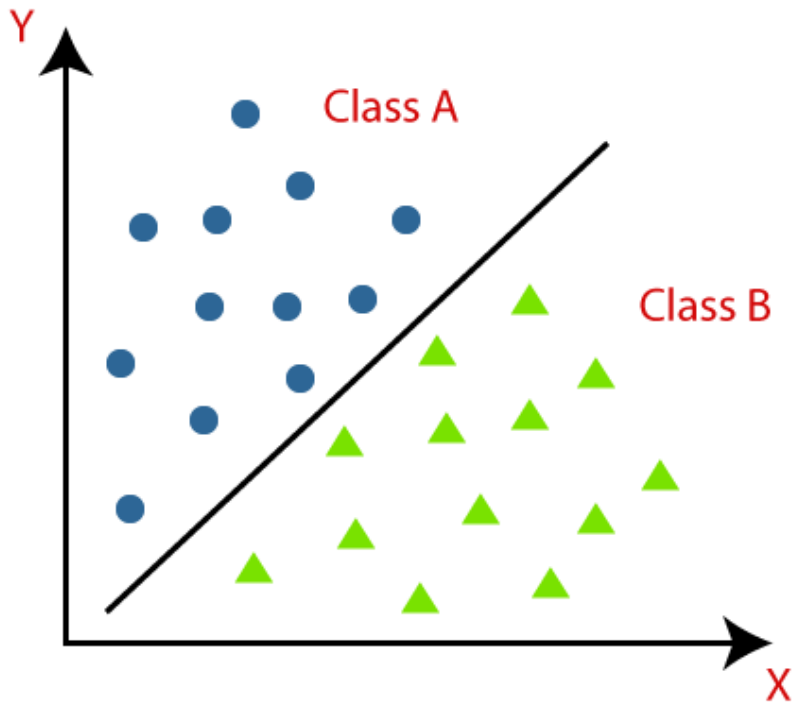
Linear classifier

- General structure of a hyperplane is

$$a_1x_1 + a_2x_2 + \cdots + a_Nx_N = b$$

$$\bar{\mathbf{a}}^T\bar{\mathbf{x}} = b$$: Hyperplane

$$\bar{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$: N dimensional vector



Linear classifier

Halfspace: Class.

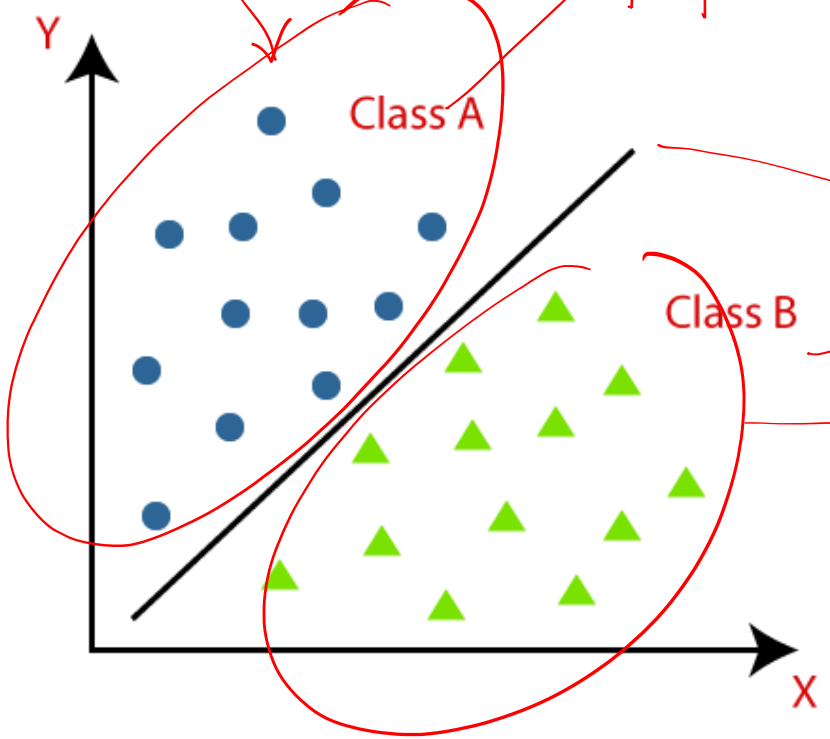
- General structure of a linear classifier is

$\bar{a}^T\bar{x} > b$ / C_0 HalfSpace

$C_0 : \bar{a}^T\bar{x} \geq b$

$C_1 : \bar{a}^T\bar{x} < b$

Halfspace



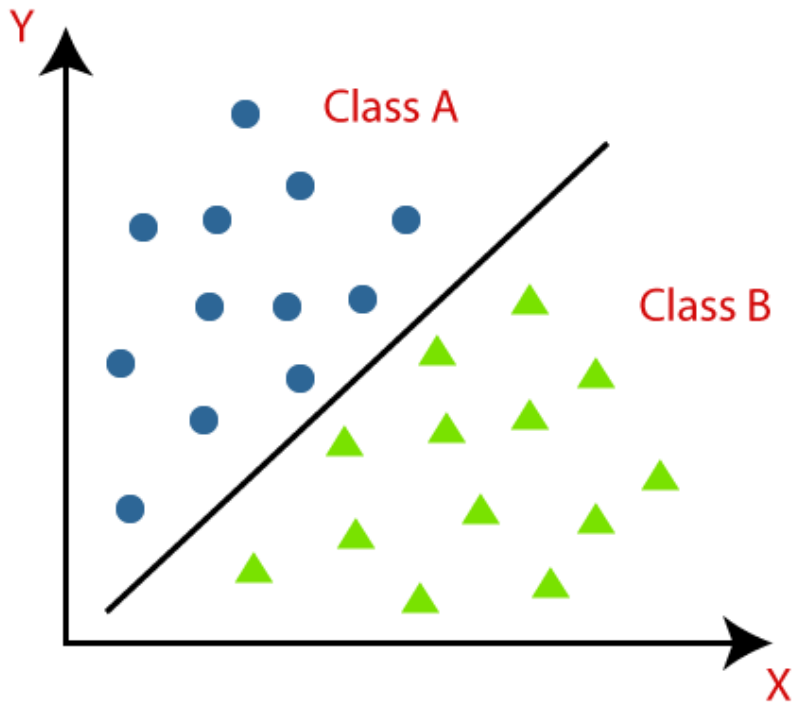
$\bar{a}^T\bar{x} = b$

$\bar{a}^T\bar{x} < b$

Linear classifier

- General structure of a linear classifier is

$$C_0: \bar{\mathbf{a}}^T \bar{\mathbf{x}} \geq b : c_0$$
$$C_1: \bar{\mathbf{a}}^T \bar{\mathbf{x}} < b : c_1.$$



SVM

- In SVM we consider parallel hyperplanes

Positive hyperplane: $\bar{a}^T\bar{x} + b = 1$

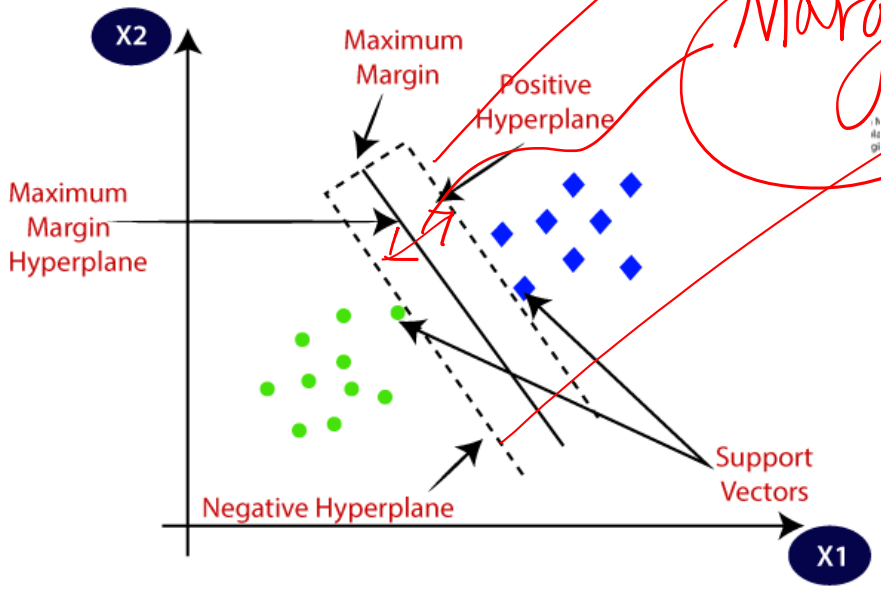
Negative hyperplane: $\bar{a}^T\bar{x} + b = -1$

Parallel hyperplanes.

$\bar{a}^T\bar{x} + b = 1 \Rightarrow \bar{a}^T\bar{x} = 1 - b$ — c_1

$\bar{a}^T\bar{x} + b = -1 \Rightarrow \bar{a}^T\bar{x} = (-1 - b)$ — c_2

Margin

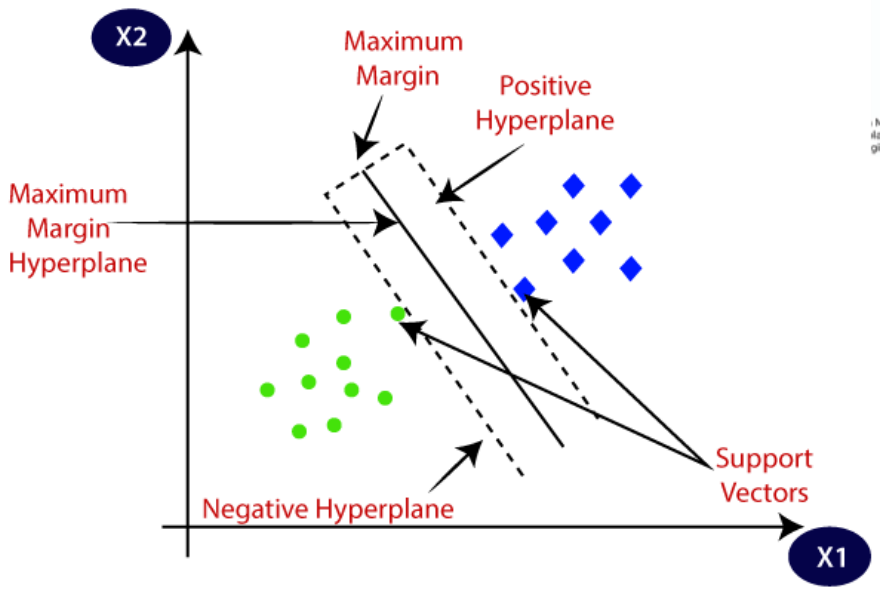


SVM

- In SVM we consider parallel hyperplanes

Positive hyperplane: $\bar{\mathbf{a}}^T\bar{\mathbf{x}} + b = 1$

Negative hyperplane: $\bar{\mathbf{a}}^T\bar{\mathbf{x}} + b = -1$



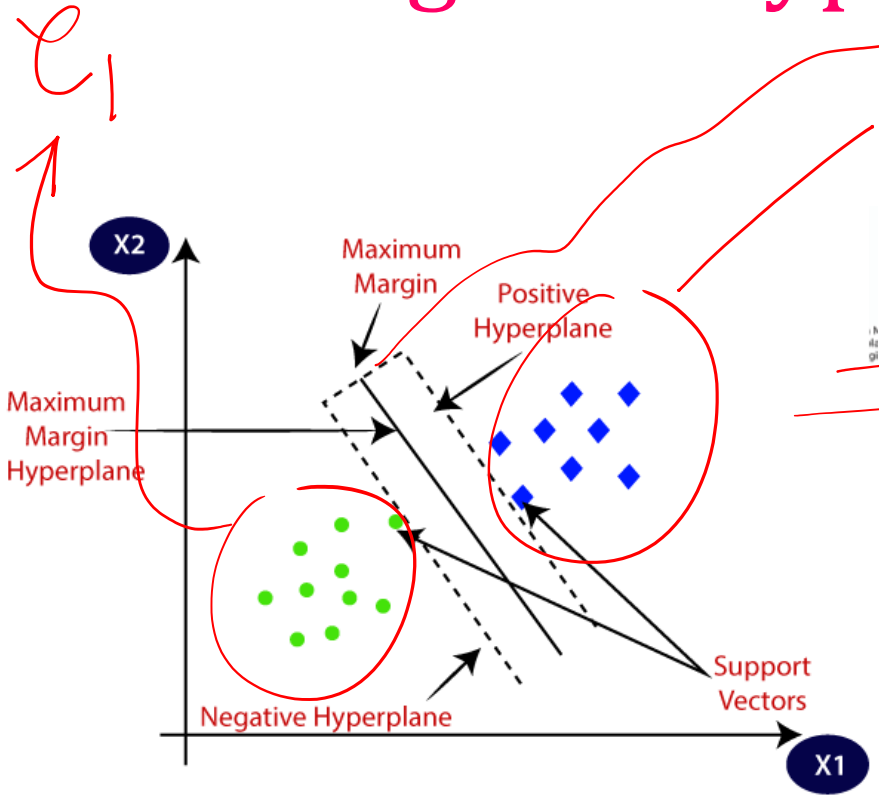
SVM

- The classes are given as

Positive hyperplane: $\bar{a}^T\bar{x} + b \geq 1$: Halfspace

Negative hyperplane: $\bar{a}^T\bar{x} + b \leq -1$: Halfspace

Margin



c_1

c_0

H_1

H_2

Positive Fence

Wall

Slab.

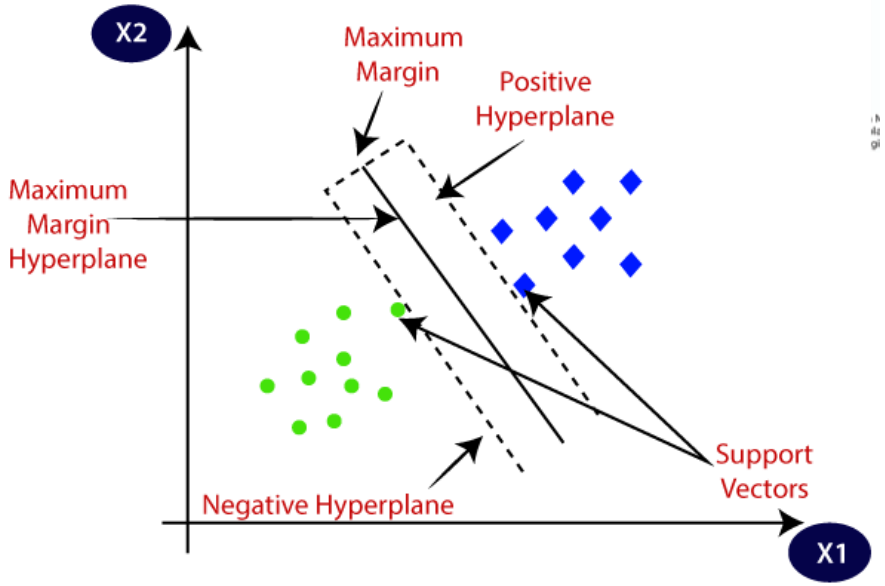
Negative Fence

SVM

- The classes are given as

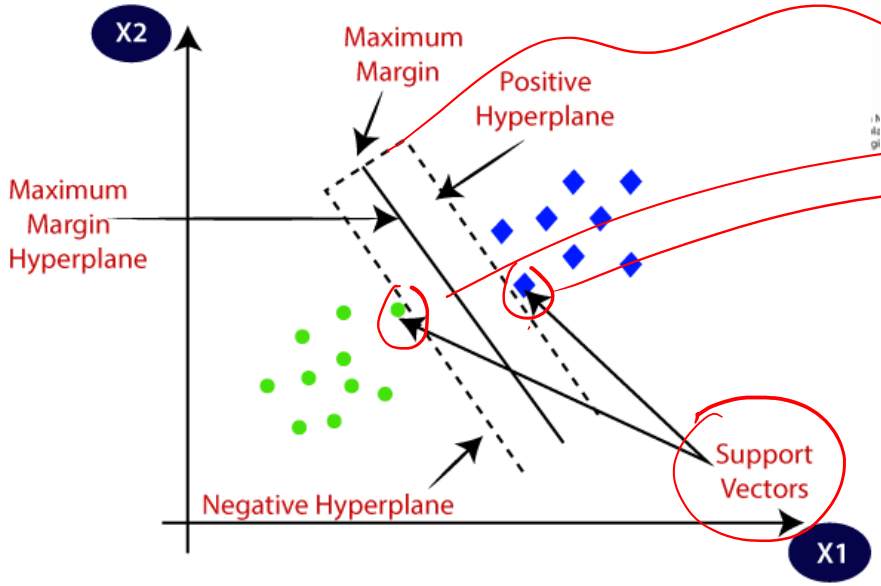
  Positive hyperplane: $\bar{\mathbf{a}}^T\bar{\mathbf{x}} + b \geq 1$

  Negative hyperplane: $\bar{\mathbf{a}}^T\bar{\mathbf{x}} + b \leq -1$



Modified optimization problem

- The width of the slab is termed the **margin**
- Best classifier **maximizes the margin** between the two classes



MaxMargin ⇒ Lower classification error

Support vectors.

Margin

- How to determine the **margin** between two hyperplanes?

$$\bar{\mathbf{a}}^T\bar{\mathbf{x}} = c_1$$
$$\bar{\mathbf{a}}^T\bar{\mathbf{x}} = c_2$$

Parallel hyperplanes-

Margin

- How to determine the margin between two hyperplanes?

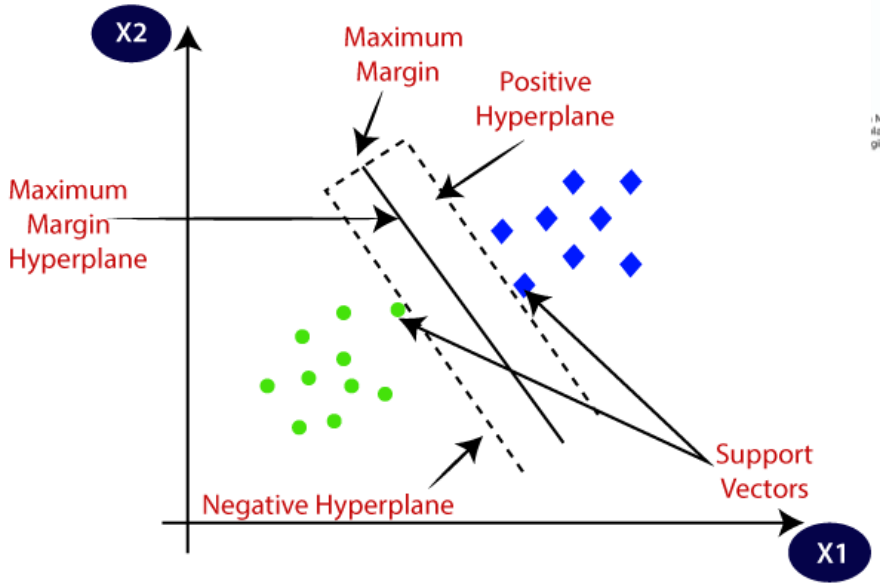
$$\bar{\mathbf{a}}^T\bar{\mathbf{x}} = c_1$$
$$\bar{\mathbf{a}}^T\bar{\mathbf{x}} = c_2$$

Margin

- Distance between the two hyperplanes is

Parallel.

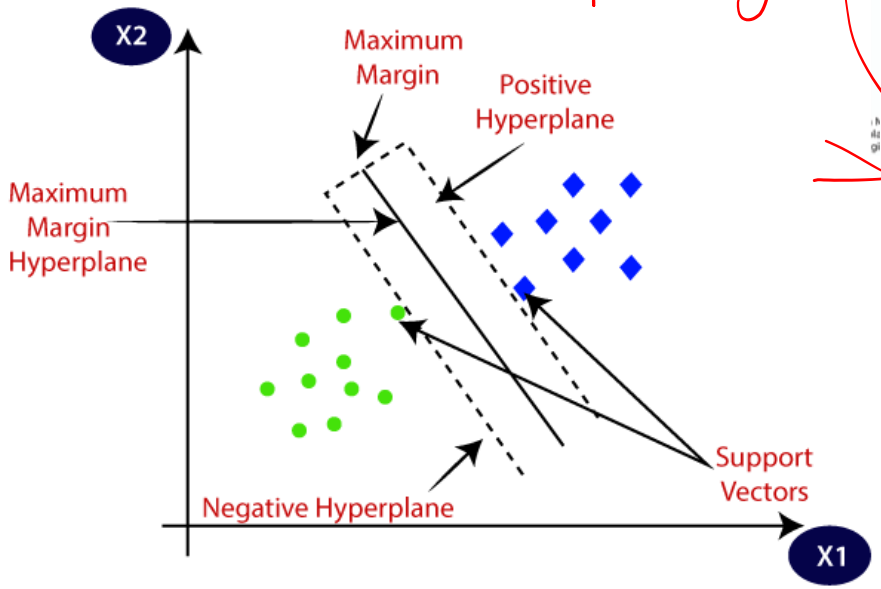
Margin $\frac{|c_1 - c_2|}{\|\bar{a}\|}$

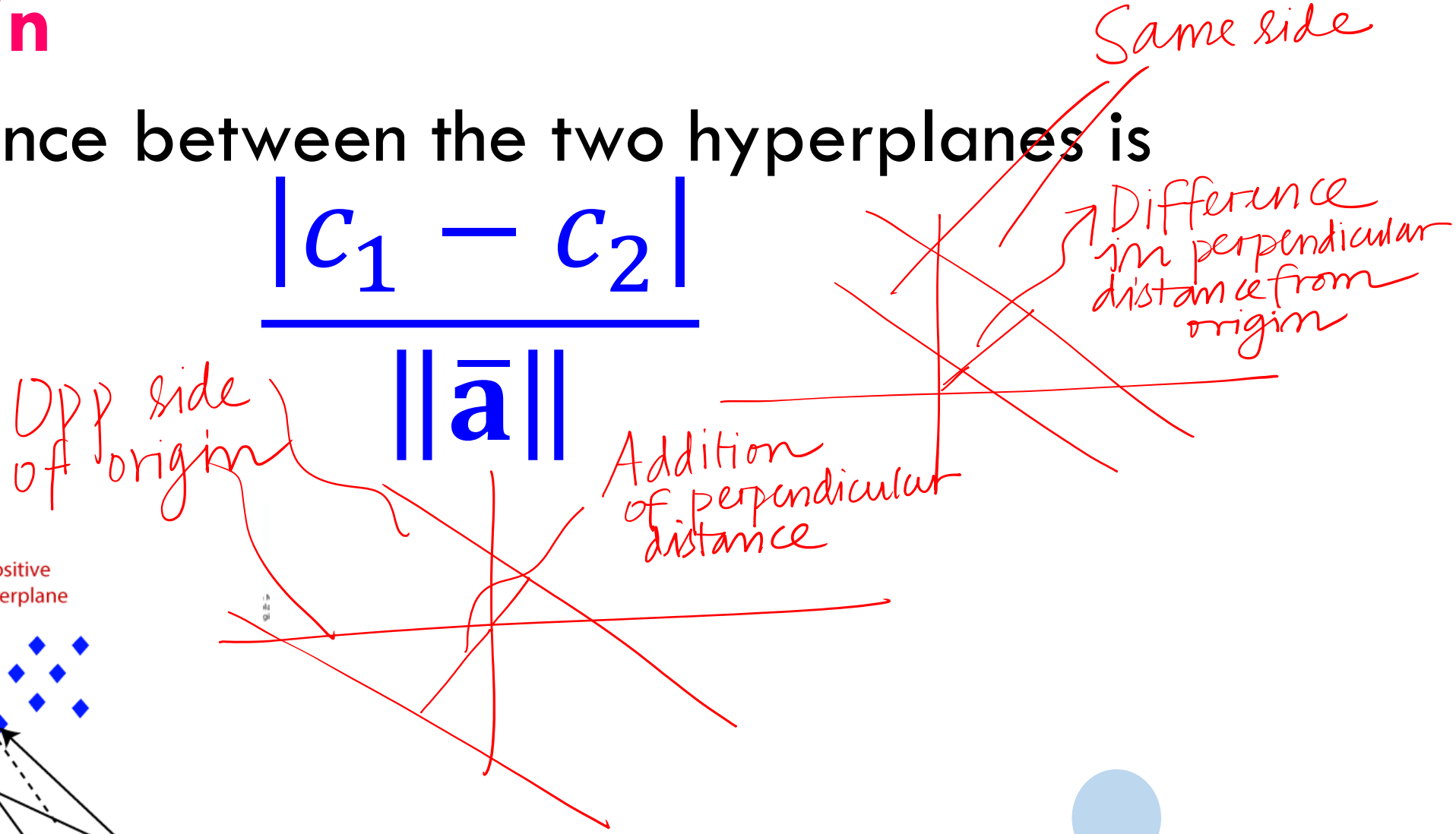


Margin

- Distance between the two hyperplanes is

$$\frac{|c_1 - c_2|}{\|\bar{a}\|}$$





SVM

+ve hyperplane

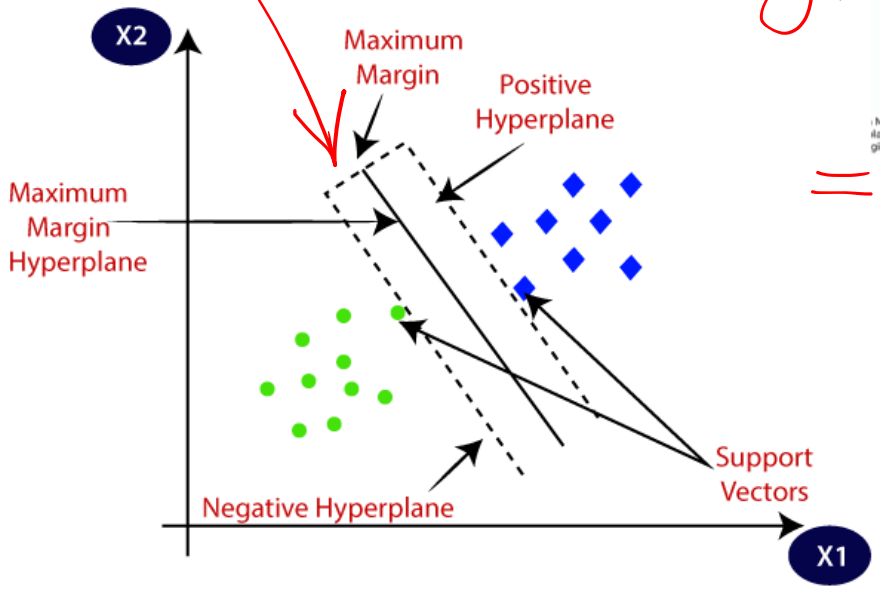
- In SVM we consider parallel hyperplanes

$$\bar{\mathbf{a}}^T\bar{\mathbf{x}} + b = 1 \Rightarrow \bar{\mathbf{a}}^T\bar{\mathbf{x}} = 1 - b \quad \text{— } c_1$$

$$\bar{\mathbf{a}}^T\bar{\mathbf{x}} + b = -1 \Rightarrow \bar{\mathbf{a}}^T\bar{\mathbf{x}} = -1 - b \quad \text{— } c_2$$

-ve hyperplane

$\frac{2}{\|\bar{a}\|}$



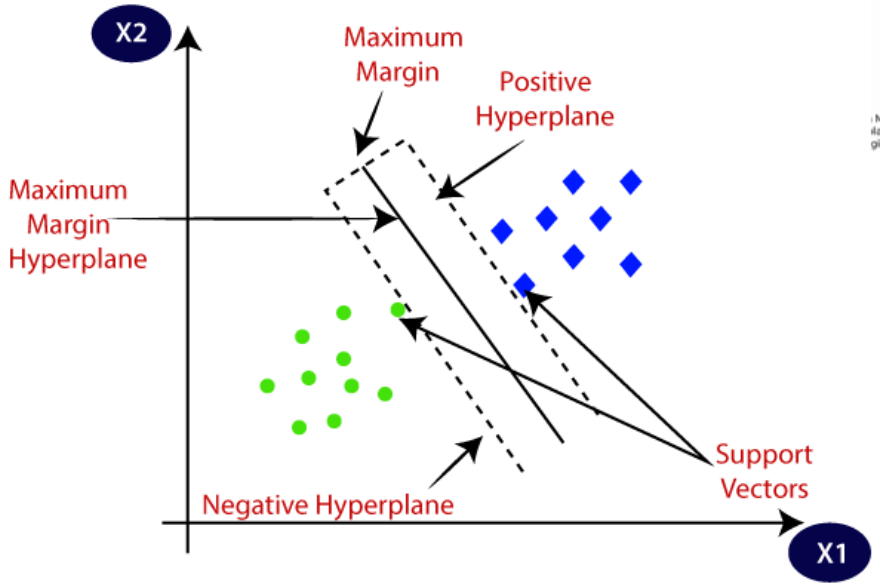
$$\frac{|c_1 - c_2|}{\|\bar{a}\|} = \frac{|(1-b) - (-1-b)|}{\|\bar{a}\|} = \frac{2}{\|\bar{a}\|}$$

Margin

- Distance between the hyperplanes or the **margin** is

$$\frac{2}{\|\bar{a}\|}$$

Distance between positive & negative hyperplanes..



Linear classifier

- How to determine the linear classifier
- Consider the **training set**

$$\bar{\mathbf{x}}(1), \bar{\mathbf{x}}(2), \ldots, \bar{\mathbf{x}}(M) \in C_0$$
$$\bar{\mathbf{x}}(M+1), \ldots, \bar{\mathbf{x}}(2M) \in C_1$$

2M points

M

$\widetilde{M}$

Maximum margin classifier

- The problem to determine classifier with **maximum margin** is

$$\max \frac{2}{\|\bar{a}\|} \equiv \min \|\bar{a}\|$$

$$\text{s.t.} \quad \bar{a}^T \bar{x}(i) + b \geq 1 \quad : \quad i = 1, 2, \ldots, M$$
$$\bar{a}^T \bar{x}(i) + b \leq -1 \quad : \quad i = M+1, \ldots, 2M$$

Constraints

Maximum margin classifier

- The problem to determine classifier with **maximum margin** is

convex

objective

$$\max \frac{2}{\|\bar{\mathbf{a}}\|_2} \equiv \min \|\bar{\mathbf{a}}\|_2$$

constrained optimization problem

constraints

$$\mathcal{C}_0: \bar{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b \geq 1, 1 \leq i \leq M$$

$$\mathcal{C}_1: \bar{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b \leq -1, M+1 \leq i \leq 2M$$

$2M$ constraints

$\Rightarrow 2M$ Lagrange multipliers λ_i

$\lambda_1, \lambda_2, \ldots, \lambda_{2M}$

Support vector machine

- The above problem is **convex** and can be readily solved
- This classifier is termed a **Support Vector Machine (SVM)**

Dual SVM and Kernel SVM

SVM Response

- Let the response be defined as

$\bar{x}(i)$

Response

$$\begin{cases} y(i) = 1, & i = 1, 2, \ldots, M \\ y(i) = -1, & i = M+1, \ldots, 2M \end{cases}$$

SVM Response

- Let the response be defined as

$$\mathcal{C}_0 : y(i) = 1, 1 \leq i \leq M$$

$$\mathcal{C}_1 : y(i) = -1, M + 1 \leq i \leq 2M$$

Lagrangian

Convex optimization

- We use the **Lagrangian** to simplify the problem
- And formulate the DUAL optimization problem
- Details in Appendix

Constrained optimization problem

$f(x)$: objective function

$g_i(x) \leq 0$: constraints $i = 1, 2, \ldots$

Lagrangian

$f(x) + \sum_i \lambda_i g_i(x)$

Lagrange multiplier

convex $\Rightarrow$ convex optimization problem

Lagrangian

$$\bar{\mathbf{a}} = \sum_{i=1}^{2M} \lambda_i y(i) \bar{\mathbf{x}}(i)$$

Lagrange multipliers. weights.

Linear combination

- Thus, $\bar{\mathbf{a}}$ can be expressed as linear combination of $\bar{\mathbf{x}}_i$.

Lagrangian

$$\bar{\mathbf{a}} = \sum_{i=1}^{2M} \lambda_i y(i) \bar{\mathbf{x}}(i)$$

$\bar{x}(i)$ is included only if $\lambda_i \neq 0$

Support vectors.

- The points for which $\lambda_i \neq 0$ are termed the **support vectors**.

Dual SVM

- Therefore, the **dual problem** can be formulated as

$$\max \sum_{i=1}^{2M} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{2M} \lambda_i \lambda_j y(i) y(j) \bar{\mathbf{x}}^T(i) \bar{\mathbf{x}}(j)$$

$$\text{subject to } \lambda_i \geq 0$$

$$\sum_{i=1}^{2M} \lambda_i y(i) = 0$$

Dual SVM

- How to calculate b?
- For any point for which $\lambda_i \neq 0$

$$y(i)(\vec{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b) - 1 = 0$$

Any point $\lambda_i \neq 0$.

Solve this equation
This determines b.

Dual SVM

- Note that the quantity $\bar{\mathbf{x}}^T(i)\bar{\mathbf{x}}(j)$ denotes the inner product.
- This can be represented as $\langle \bar{\mathbf{x}}(i), \bar{\mathbf{x}}(j) \rangle$

Dot product Between vectors.

inner product

Dual SVM

- Using this notation, the **dual SVM** problem can be defined as

$$\max \sum_{i=1}^{2M} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{2M} \lambda_i \lambda_j y(i) y(j) \langle \bar{\mathbf{x}}(i), \bar{\mathbf{x}}(j) \rangle$$

$$\text{subject to } \lambda_i \geq 0$$

$$\sum_{i=1}^{2M} \lambda_i y(i) = 0$$

Dual objective

inner product

Dual constraints

Kernel

- One can now replace $\langle \bar{\mathbf{x}}(i), \bar{\mathbf{x}}(j) \rangle$ by a kernel

$$K\left(\bar{\mathbf{x}}(i), \bar{\mathbf{x}}(j)\right)$$

inner product in a high dimensional non-linear feature space

Kernel SVM

- Using this notation, the **kernel SVM** problem can be defined as

Dual objective

$$\max \sum_{i=1}^{2M} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{2M} \lambda_i \lambda_j y(i) y(j) K\left(\bar{\mathbf{x}}(i), \bar{\mathbf{x}}(j)\right)$$

Kernel.

$$\text{subject to } \lambda_i \geq 0$$

$$\sum_{i=1}^{2M} \lambda_i y(i) = 0$$

Kernel

- Kernel can be used to model ***non-linear features***.

Ultra high dimensional nonlinear features.

Gaussian Kernel

- An interesting kernel is the **Gaussian kernel** defined as,

$$K(\bar{x}(i), \bar{x}(j)) = \exp\left(-\frac{\|\bar{x}(i) - \bar{x}(j)\|^2}{2\sigma^2}\right)$$

∞ dimensional non-linear feature space

Gaussian Kernel

- An interesting kernel is the **Gaussian kernel** defined as,

$$K\left(\bar{\mathbf{x}}(i), \bar{\mathbf{x}}(j)\right) = \exp\left(-\frac{\|\bar{\mathbf{x}}(i) - \bar{\mathbf{x}}(j)\|^2}{2\sigma^2}\right)$$

Gaussian Kernel

- Example- **Handwritten digit recognition**, from 16×16 images
- **Gaussian kernel** SVMs yield very good performance!

Very powerful Technique.

```python
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC

```

Scikit Learn

Datasets For ML

model selection Sublibrary

Test Train Split To split training dataset into test & Train components

SVC Support Vector classifier Algorithm

ScaleData

To characterize accuracy of SVM

```
8  bcancer = datasets.load_breast_cancer()
9
```

Breast cancer Dataset.

Breast cancer dataset

- Imported from Scikit-learn
- Contains 569 samples
- 30 features like mean radius, mean texture, mean perimeter etc
- 212 samples are labeled malignant and 357 are benign.

```
10  X = bcancer.data
11  Y = bcancer.target
```

569 Feature Vectors.
569 rows
Each of N=30 Features.

Type of mass.
Benign 0
Malignant 1.

```
13  scaler = StandardScaler();
14  X = scaler.fit_transform(X)
```

Scale data to make mean 0 unit variance.

To bring all Features onto same scale.

} Preprocessing step.

```
16
17  Xtrain, Xtest, Ytrain, Ytest \
18  = train_test_split(X, Y, test_size = 0.20, random_state = 0)
19
20
```

Test-Train Split of Training Data.

20% For Testing

80% For training

```
20
21  # Linear SVM
22
23  svmc = SVC(kernel = 'linear', random_state = 0)
24  svmc.fit(Xtrain, Ytrain)
25  Ypred = svmc.predict(Xtest)
26  svmcscore = accuracy_score(Ypred,Ytest)
27  print('Accuracy score of Linear SVM Classifier is',100*svmcscore,'%\n')
28
```

Linear SVM

Linear SVC

Fit linear SVC to training data

Predict For Test data

Compare actual & Prediction

True diagnosis

Prediction by SVC

Prints / accuracy of prediction

```
29  # Kernel SVM RBF - Gaussian Kernal
30
31  ksvmc = SVC(kernel = 'rbf', random_state = 0)
32  #ksvmc = SVC(kernel = 'poly', random_state = 0)
33  #ksvmc = SVC(kernel = 'sigmoid', random_state = 0)
34  ksvmc.fit(Xtrain, Ytrain)
35  Ypred = ksvmc.predict(Xtest)
36  svmcscore = accuracy_score(Ypred,Ytest)
37  print('Accuracy score of Kernel SVM Classifier with RBF is',100*svmcscore,'%\n')
38
```

Kernel SVM

RBF — Radial Basis Function — Gaussian kernel.

Polynomial kernel.

Sigmoid kernel.

```
29  # Kernel SVM RBF - Gaussian Kernal
30
31  ksvmc = SVC(kernel = 'rbf', random_state = 0)
32  #ksvmc = SVC(kernel = 'poly', random_state = 0)
33  #ksvmc = SVC(kernel = 'sigmoid', random_state = 0)
34  ksvmc.fit(Xtrain, Ytrain)
35  Ypred = ksvmc.predict(Xtest)
36  svmcscore = accuracy_score(Ypred,Ytest)
37  print('Accuracy score of Kernel SVM Classifier with RBF is',100*svmcscore,'%\n')
38
```

Kernel SVM Fit to data

Predict type of ~~mass~~ using kernel SVM.

SVM score between 0 - 1. What Fraction of Predictions are accurate

Accuracy of Prediction for kernel SVM.

Appendix

SVM Classifier

- Recall, the problem to determine classifier with **maximum margin** is

$$\max \frac{2}{\|\bar{\mathbf{a}}\|_2} \equiv \min \|\bar{\mathbf{a}}\|_2$$

$$\mathcal{C}_0: \bar{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b \geq 1, 1 \leq i \leq M$$

$$\mathcal{C}_1: \bar{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b \leq -1, M+1 \leq i \leq 2M$$

SVM Response

- Let the response be defined as

$$\mathcal{C}_0 : y(i) = 1, 1 \leq i \leq M$$

$$\mathcal{C}_1 : y(i) = -1, M + 1 \leq i \leq 2M$$

SVM Constraints

- The constraints can be combined as

SVM Constraints

- The constraints can be combined as

$$y(i)(\bar{\mathbf{a}}^T\bar{\mathbf{x}}(i)+b)\geq 1, 1\leq i\leq 2M$$

$$\Rightarrow -y(i)(\bar{\mathbf{a}}^T\bar{\mathbf{x}}(i)+b)\leq -1$$

$$\Rightarrow -(y(i)(\bar{\mathbf{a}}^T\bar{\mathbf{x}}(i)+b)-1)\leq 0$$

SVM Classifier

- The SVM classifier problem can be recast as

$$\min \frac{1}{2} \|\bar{\mathbf{a}}\|^2$$

$$\text{subject to}$$

$$-(y(i)(\bar{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b) - 1) \leq 0, 1 \leq i \leq 2M$$

Lagrangian

- The **Lagrangian** for this problem is

$$\frac{1}{2}\|\bar{\mathbf{a}}\|^2 - \sum_{i=1}^{2M} \lambda_i (y(i)(\bar{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b) - 1)$$

Lagrangian

- Setting gradient wrto $\bar{\mathbf{a}}$ to zero

$$\nabla_{\bar{\mathbf{a}}}\left(\frac{1}{2}\|\bar{\mathbf{a}}\|^2-\sum_{i=1}^{2M}\lambda_i(y(i)(\bar{\mathbf{a}}^T\bar{\mathbf{x}}(i)+b)-1)\right)=0$$

$$\Rightarrow \bar{\mathbf{a}}-\sum_{i=1}^{2M}\lambda_i y(i)\,\bar{\mathbf{x}}(i)=0\Rightarrow \bar{\mathbf{a}}=\sum_{i=1}^{2M}\lambda_i y(i)\bar{\mathbf{x}}(i)$$

Lagrangian

$$\bar{\mathbf{a}} = \sum_{i=1}^{2M} \lambda_i y(i) \bar{\mathbf{x}}(i)$$

- Thus, $\bar{\mathbf{a}}$ can be expressed as linear combination of $\bar{\mathbf{x}}(i)$.

Lagrangian

$$\bar{\mathbf{a}} = \sum_{i=1}^{2M} \lambda_i y(i) \bar{\mathbf{x}}(i)$$

- The points for which $\lambda_i \neq 0$ are termed the **support vectors**.

Lagrangian

- Setting gradient wrto b to zero

$$\nabla_b \left(\frac{1}{2} \|\bar{\mathbf{a}}\|^2 - \sum_{i=1}^{2M} \lambda_i (y(i)(\bar{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b) - 1) \right) = 0$$

$$\Rightarrow \sum_{i=1}^{2M} \lambda_i y(i) = 0$$

Lagrangian

- The expression for $\bar{\mathbf{a}}$ can be substituted in the Lagrangian
- The resulting expression can be simplified as shown next

Lagrangian

- Consider the **Lagrangian** given as

$$\frac{1}{2}\|\bar{\mathbf{a}}\|^2 - \sum_{i=1}^{2M} \lambda_i (y(i)(\bar{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b) - 1)$$

$$= \frac{1}{2}\bar{\mathbf{a}}^T \bar{\mathbf{a}} - \sum_{i=1}^{2M} \lambda_i (y(i)(\bar{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b) - 1)$$

Lagrangian

- Substitute $\bar{\mathbf{a}} = \sum_{i=1}^{2M} \lambda_i y(i) \bar{\mathbf{x}}(i)$

$$= \frac{1}{2}\left(\sum_{i=1}^{2M} \lambda_i y(i) \bar{\mathbf{x}}(i)\right)^T \left(\sum_{i=1}^{2M} \lambda_i y(i) \bar{\mathbf{x}}(i)\right)$$

$$- \sum_{i=1}^{2M} \lambda_i \left(y(i) \left(\left(\sum_{j=1}^{2M} \lambda_i y(i) \bar{\mathbf{x}}(i)\right)^T \bar{\mathbf{x}}(i) + b \right) - 1 \right)$$

Lagrangian

$$= \sum_{i=1}^{2M} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{2M} \lambda_i \lambda_j y(j) \bar{\mathbf{x}}^T(i) \bar{\mathbf{x}}(j) - b \sum_{i=1}^{2M} \lambda_i y(i)$$

$$= \sum_{i=1}^{2M} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{2M} \lambda_i \lambda_j y(i) y(j) \bar{\mathbf{x}}^T(i) \bar{\mathbf{x}}(j)$$

Dual SVM

- Therefore, the **dual problem** can be formulated as

$$\max \sum_{i=1}^{2M} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{2M} \lambda_i \lambda_j y(i) y(j) \bar{\mathbf{x}}^T(i) \bar{\mathbf{x}}(j)$$

$$\text{subject to } \lambda_i \geq 0$$

$$\sum_{i=1}^{2M} \lambda_i y(i) = 0$$

Dual SVM

- How to calculate b?
- For any point for which $\lambda_i \neq 0$

Dual SVM

- How to calculate b?
- For any point for which $\lambda_i \neq 0$

$$y(i)(\bar{\mathbf{a}}^T \bar{\mathbf{x}}(i) + b) - 1 = 0$$

Instructors may use this white area (14.5 cm / 25.4 cm) for the text. Three options provided below for the font size.

Font: Avenir (Book), Size: 32, Colour: Dark Grey

Font: Avenir (Book), Size: 28, Colour: Dark Grey

Font: Avenir (Book), Size: 24, Colour: Dark Grey

Do not use the space below.