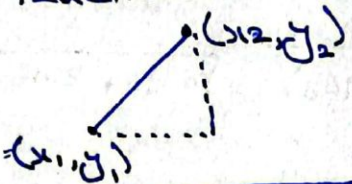
Lecture 4 (1)

Classification

K-Nearest Neighbour:

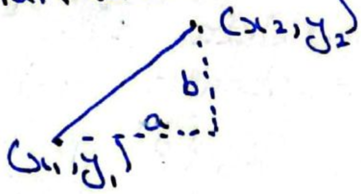
→ Instance based method. Consider all instances corresponding to points in the n-dimentional space R^n.
→ Can be used for both classification & Regression.
→ In classification, we have fixed number of categories.
→ Voting by neighbours, calculating distance.

(1) Euclidean Distance

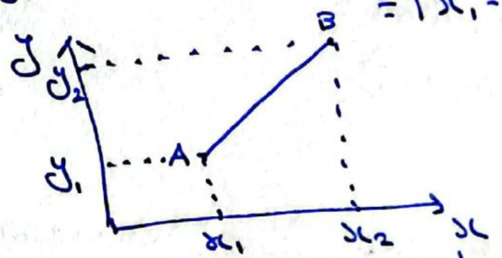


$$ED = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

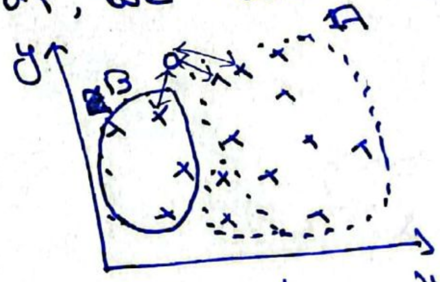
(2) Manhattan Distance



$$MD = a + b$$
$$= |x_1 - x_2| + |y_1 - y_2|$$



→ In classification, we have fixed number of categories.

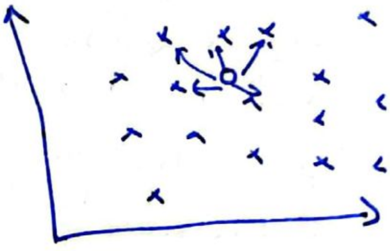


→ Either 1 or 2 category

A → 2
B → 1
So will be in A.

→ If we take Regression (House price prediction) problem.

Hous Size	Rooms	Price
5M	3	1.6M
5M	4	1.9M
5M	5	2.1M
5M	4	2M
10M:	5	2.5M

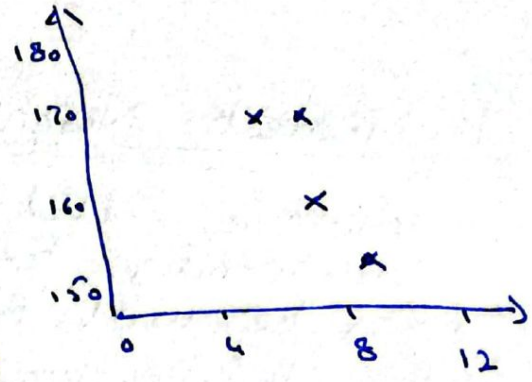


Price of test data = $\frac{1.6 + 1.9 + 2.1 + 2 + 2.2}{5}$

= ___

Numerical Example:

Rating	Duration	Genre
8.0 (Mission Imp)	160	Action
6.2 (Gadar)	170	Action
7.2 (Rockey)	168	Comedy
8.2 (OMG)	155	Comedy



→ Classify "Barbie" movie with rating of 7.4 & 114 minute duration. Predict genre

Step 1

Calculate ~~Bre~~ the ED betw the new and ~~neighbours~~ each of old movie.

E.Dist to (8.0, 160) = $\sqrt{(7.4-8.0)^2 + (114-160)^2} = \sqrt{0.36 + 2116} = 46$

ED to (6.2, 170) = " $= 56.01$

" " (7.2, 168) = " $= 54$

" " (8.2, 155) = " $\boxed{= 41}$

Step 2

Select K nearest Neighbour

for $K = 1$
So "Comedy".

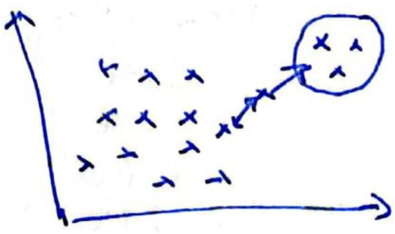
for $K = 3$
Action, Comedy, Comedy
So "Comedy".

Step 3

Majority Voting
So "Comedy".

Limitations:

1) Not feasible for large data sets, becz of distance calculations involved.
2) Very sensitive to outliers.
3) Sensitive to missing values.



Support Vector Machine (SVM):

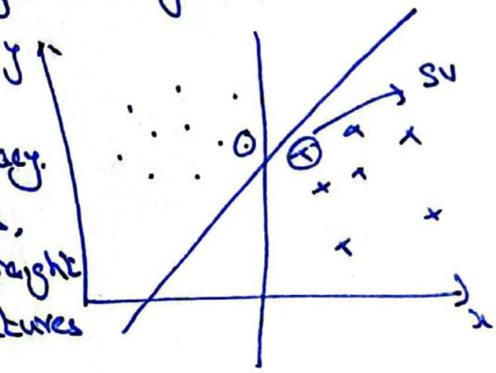
→ Most popular classification algo, however can be used for both Classification & Regression.

→ SVM goal is to draw best line/decision boundry, so it can segregate n-dimensional space into classes for putting data into appropriate categories.

→ Understand

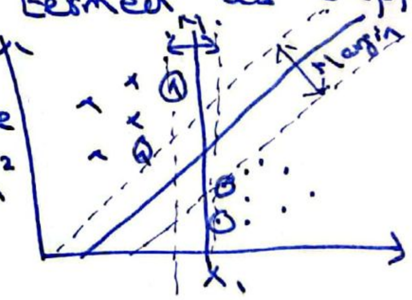
1) Support vectors
2) Hyperplane
3) Marginal Distance
4) Linear separable (SVM)
5) Non-linear separable (SVM)

Objective is to maximize Margin/Marginal distance.



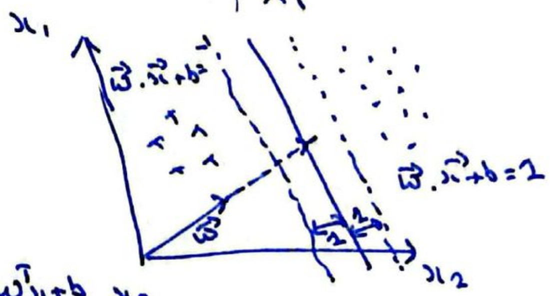
→ The line/boundry is known as hyper plane.

1) This is chosen based on accuracy.
2) Hyperplane depends on features, 2 features (x, y) will have a straight line for hyperplane while 3 features will have a 2D plane
3) Data points/vectors closet to hyperplane & which affect hyperplane position are termed as support vectors.



M = Margin or Marginal Distance
= -ve Margin + +ve Margin

Mathematical Derivation

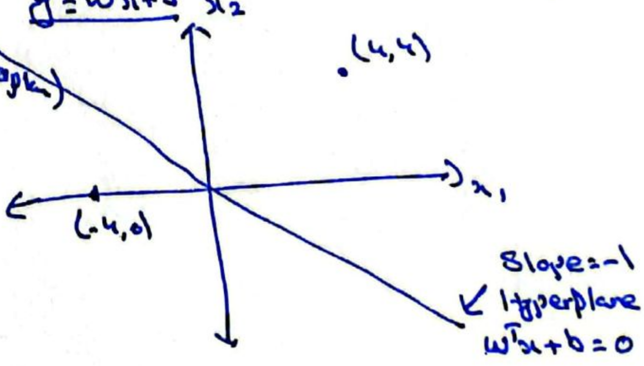


$m = 1 \quad ; \quad b = 0$

$y = W^T x + b$

$= \begin{bmatrix} -1 \\ 0 \end{bmatrix} [-4 \quad 4]$

$= 4$



Note that below the slope (hyperplane) value will always be positive

Similarly

$$y = w^T x + b$$
$$= \begin{bmatrix} -1 \\ 0 \end{bmatrix} [4 \quad 4]$$
$$= -4$$

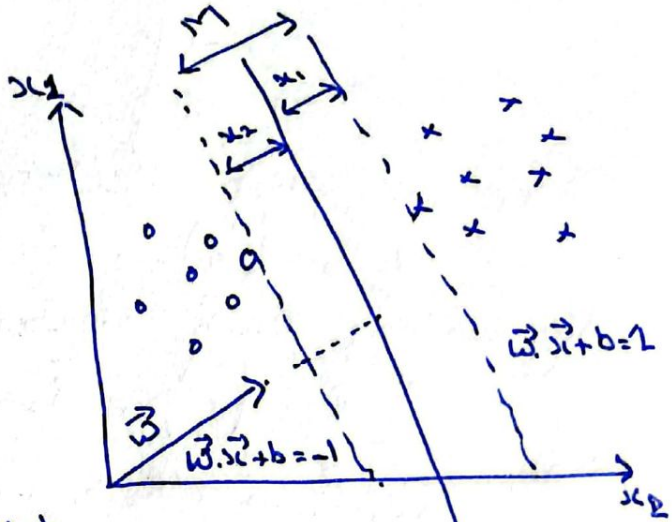
Any point above slope(hyperplane), we will get negative values.

Margin Distance = $x_2 - x_1$

$$= w^T x_1 + b = -1$$
$$w^T x_2 + b = 1$$
$$\overline{w^T(x_2 - x_1) = 2}$$

$$\frac{w^T}{\|w\|}(x_2 - x_1) = 2$$

$$(x_2 - x_1) = \frac{2}{\|w\|}$$



such that

$$y_i \begin{cases} +1 & w^T x + b \geq 1 \\ -1 & w^T x + b \geq -1 \end{cases}$$

+ve   -ve

$\vec{w}.\vec{x}+b=0$

For checking of misclassification

$$y_i \times (w^T x_i + b_i) \geq 1$$

Error Calculation

$$(\vec{w}, \vec{b}) = \min \frac{\|w\|}{2} + C_i \sum_{i=1}^{n} \xi_i$$

→ Aim is to create a generalized model, so some errors are okay to avoid overfitting.

→ C_i is known as regularization & we get it by hypertuning.

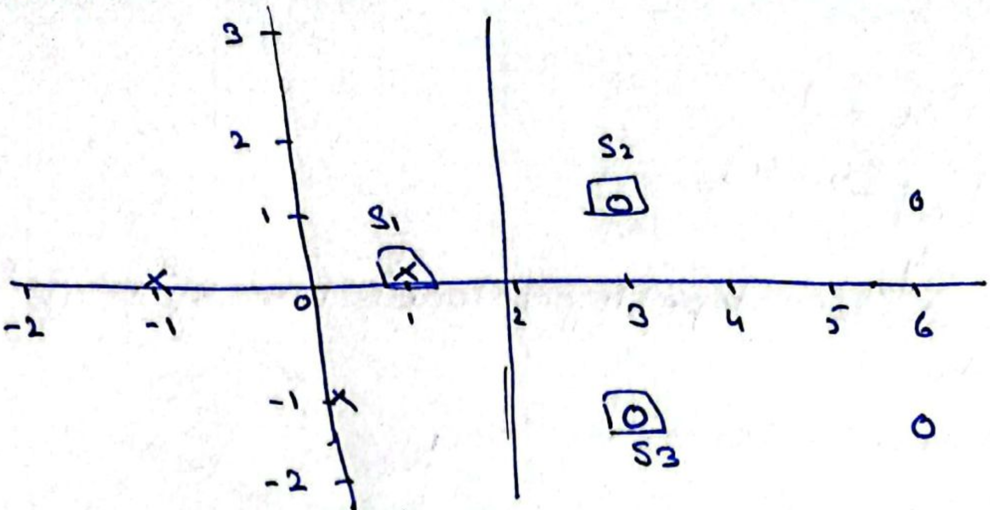
Numerical Example:

Positively labelled Data points

$$\left\{ \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} 6 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 \\ -1 \end{pmatrix} \right\}$$

Negatively labelled data points

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}$$



O → 1
x → -1
□ → SV

Three support vectors

$$\left\{ s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad s_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \quad s_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \right\}$$

Augmenting each vector with 1 as bias input

$$\left\{ \tilde{s}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \tilde{s}_2 = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}, \quad \tilde{s}_3 = \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \right\}$$

So

$$\alpha_1 \tilde{s}_1 \cdot \tilde{s}_1 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_1 + \alpha_3 \tilde{s}_3 \cdot \tilde{s}_1 = -1$$
$$\alpha_1 \tilde{s}_1 \cdot \tilde{s}_2 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_2 + \alpha_3 \tilde{s}_3 \cdot \tilde{s}_2 = +1$$
$$\alpha_1 \tilde{s}_1 \cdot \tilde{s}_3 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_3 + \alpha_3 \tilde{s}_3 \cdot \tilde{s}_3 = +1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} = 1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} = 1$$

$$\alpha_1 (1+0+1) + \alpha_2 (3+0+1) + \alpha_3 (3+0+1) = -1$$
$$\alpha_1 (3+0+1) + \alpha_2 (9+1+1) + \alpha_3 (9-1+1) = 1$$
$$\alpha_1 (3+0+1) + \alpha_2 (9-1+1) + \alpha_3 (9+1+1) = 1$$

$$2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1$$
$$4\alpha_1 + 11\alpha_2 + 9\alpha_3 = 1$$
$$4\alpha_1 + 9\alpha_2 + 11\alpha_3 = 1$$

$$\alpha_1 = -3.5 \; ; \; \alpha_2 = 0.75 \; ; \; \alpha_3 = 0.75$$

Calculating weight factor

$$\vec{w} = \sum_i \alpha_i \tilde{s}_i$$
$$= -3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix}1\\0\\-2\end{pmatrix}$$

Equating last entry to bias

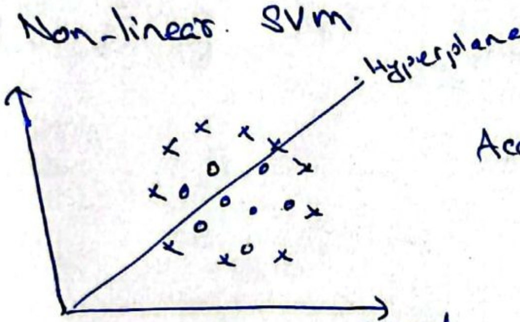
Hyperplane Equation

$$y = wx + b$$

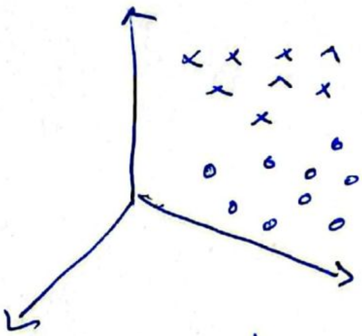
$$w = \begin{pmatrix}1\\0\end{pmatrix} \quad \& \quad b = -2$$

→ SVM Types:

1) Linear SVM
2) Non-linear SVM



Accuracy ≤ 50%



→ Convert from 2D (low dimension) into high dimension using SVM kernels. 2D → 3D, 4D

e.g.

$$Z = x^2 + y^2$$

→ There are three types (1) Polynomial kernels, (2) RBF kernels; & (3) Sigmoid kernel.

For example

$y = f(x)$ → Transform to $(x)^2$