Week 3 : Session 1

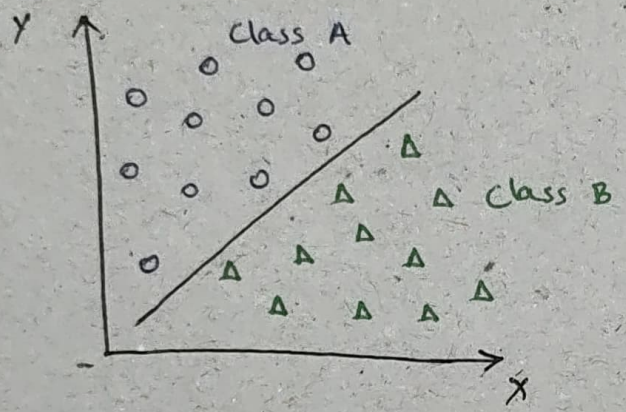
4. Support Vector Machines (SVM)

The Support Vector Machines are used for Classification, which is an important tool in ML.

The classification problem determines, to which class an observation belongs to.

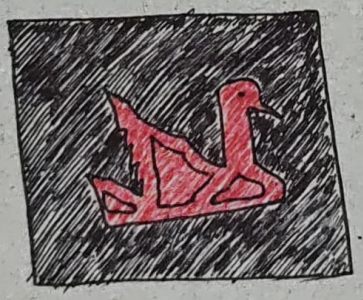
Binary classification :

Typically we have Binary classification, which means that there are 2 classes.



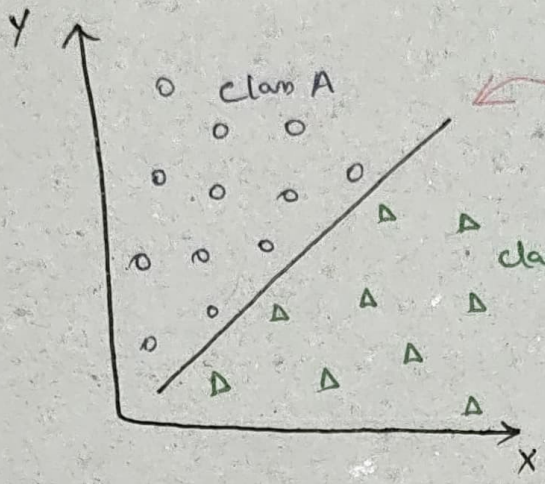
Applications : Image Segmentation

(i) Classify pixels as belonging to Background and foreground.



① Linear Classifier :

Linear classifier corresponds to a hyperplane in N dimensions. We prefer Linear classifier, as it is easy to determine and analyse.



In 2D, Linear classifier corresponds to a Straight Line

In 3D → Plane

In ND → Hyperplane

The General Structure of Linear classifier is

$$C_0 : \bar{a}^T \bar{x} \geq b$$
$$C_1 : \bar{a}^T \bar{x} < b$$

According to the theory of Linear classification,

$$a_1 x_1 + a_2 x_2 + \ldots + a_N x_N = b$$

is basically called as Hyperplane. The Hyperplane divides the N dimensional space into two parts, each of those is known as Half space.

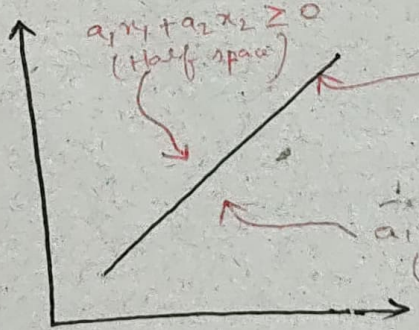
One of the Half space is

$$a_1 x_1 + a_2 x_2 + \ldots + a_N x_N \geq b$$

And, another Half space is

$$a_1 x_1 + a_2 x_2 + \ldots + a_N x_N < b$$

For instance, consider a 2D space as below,



Thus, $C_0 : \bar{a}^T \bar{x} \geq b$ is one half space (Foreground) and $C_1 : \bar{a}^T \bar{x} < b$ is another half space (Background).

Application : Disease Detection.

(ii) Based on the available data $\bar{x}$, which corresponds to either CT/MRI/Blood Test Report, we can characterize $\bar{x}$ as follows.

$\bar{x}$ < C_0 : Disease Absent
C_1 : Disease Present

How to determine the Linear classifier?

Consider the training set

$\underbrace{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_M \in C_0}$
M-points belong to class 0

$\underbrace{\bar{x}_{M+1}, \bar{x}_{M+2}, \ldots, \bar{x}_{2M} \in C_1}$
M-points belong to class 1

Using this Training data, we build the Linear classifier.

The classifier can be trained as

$C_0 : \bar{a}^T \bar{x}_i + b \geq 0 \quad ; \quad i = 1, 2, \ldots, M$

$C_1 : \bar{a}^T \bar{x}_i + b < 0 \quad ; \quad i = M+1, M+2, \ldots, 2M$

Now, we need to determine $\bar{a}$ and b, that characterize the linear classifier. This is because, recall the linear classifier corresponds to the Hyperplane $\bar{a}^T \bar{x} + b = 0$. One side of the Hyperplane is half space that corresponds to class 0; other side of the Hyperplane is half space that corresponds to class 1.

So, $\bar{a}^T \bar{x}_i + b \geq 0$ and $\bar{a}^T \bar{x}_i + b < 0$ is our problem to train the linear classifier based on the set of two M points.

(ii) M points $\in C_0$ and M points $\in C_1$

What is the problem with this formulation ?

$$\bar{a}^T \bar{x}_i + b \geq 0 \quad ; \; 1 \leq i \leq M$$

$$\bar{a}^T \bar{x}_i + b \leq 0 \quad ; \; M+1 \leq i \leq 2M$$

The above problem has a trivial solution !

(i) $\bar{a} = 0$ and $b = 0$

Therefore, if we look at a simple hyperplane and dividing it into two half spaces, that problem has a trivial solution.

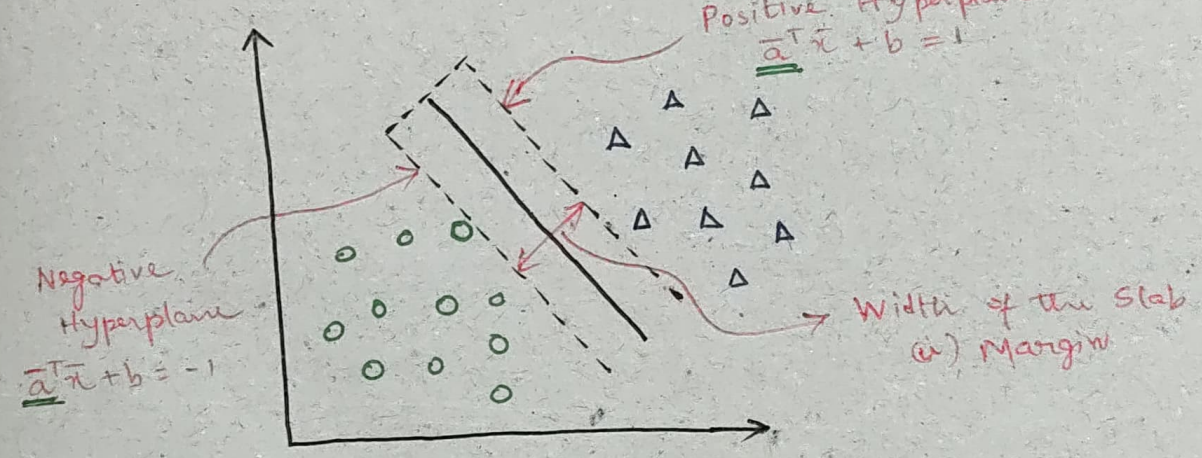
In order to avoid the trivial solution, the problem has to be modified. (i) avoid $\bar{a} = 0$ and $b = 0$.

The modified optimization problem is

$$C_0 : \bar{a}^T \bar{x}_i + b \geq 1 \quad ; \; i = 1, 2, \ldots, M$$

$$C_1 : \bar{a}^T \bar{x}_i + b \leq -1 \quad ; \; i = M+1, M+2, \ldots, 2M$$

where, $\bar{a}^T \bar{x}_i + b = 1$ and $\bar{a}^T \bar{x}_i + b = -1$ are two parallel hyperplanes.



The modified optimization problem determines two parallel hyperplanes, which separates both classes by a slab.

Recall, a line is represented as "$y = mx + c$" where 'm' is slope of the line. For any two lines, if the 'm' is same, the line would be parallel. In our case, it is $\bar{a}^T$ which is same in both hyperplanes.

The width of the slab is termed as the MARGIN, and the **best classifier** maximizes the margin between the two classes, thus the error due to misclassification is small.

- How to determine the margin between two hyperplanes?

Parallel Hyperplanes : $\bar{a}^T \bar{x} = c_1$

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

The distance b/w two Hyperplanes, (i.e) Margin $= \dfrac{|c_1 - c_2|}{\|\bar{a}\|}$

where, $\|\bar{a}\| = \sqrt{a_1^2 + a_2^2 + \dots + a_N^2}$

Example: What is the distance b/w the hyperplanes

$x_1 + 2x_2 + 3x_3 + \dots + Nx_N = 1$

$x_1 + 2x_2 + 3x_3 + \dots + Nx_N = -1$

Given: $\bar{a}^T = [1 \ 2 \ \dots \ N]$

$\Rightarrow \bar{a} = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ N \end{bmatrix}$

$c_1 = 1, \ c_2 = -1$

$$d = \frac{|c_1 - c_2|}{\|\bar{a}\|} = \frac{|1 - (-1)|}{\sqrt{1^2 + 2^2 + \dots + N^2}} = \frac{2}{\sqrt{\frac{(N)(N+1)(2N+1)}{6}}}$$

Now, consider the two Hyperplanes

$$C_0 : \bar{a}^T \bar{x} + b = 1$$

$$\Rightarrow \boxed{\bar{a}^T \bar{x} = 1 - b}$$

$$C_1 : \bar{a}^T \bar{x} + b = -1$$

$$\Rightarrow \boxed{\bar{a}^T \bar{x} = -1 - b}$$

The distance b/w the Hyperplanes, (ie) Margin $= \dfrac{|c_1 - c_2|}{\|\bar{a}\|} = \dfrac{|1 - b - (-1 - b)|}{\|\bar{a}\|}$

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

Now, we have to maximize this Margin, to get the optimal classifier.

Therefore, the problem to determine classifier with maximum margin is

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

subject to the constraints

$$\begin{cases} \bar{a}^T \bar{x}_i + b \geq 1 \; ; \; i = 1, 2, \ldots, M \\ \bar{a}^T \bar{x}_i + b \leq -1 \; ; \; i = M+1, \ldots, 2M \end{cases}$$

2M Training Set

Convex objective function

Affine constraints.

Since this is a convex optimization problem, it can be solved very easily and efficiently. This classifier is termed as Support Vector Machine (SVM)

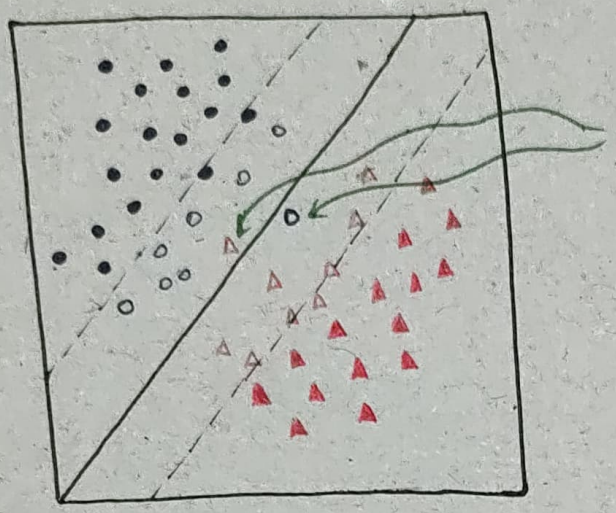
Using the convexity, it is very efficient to determine. This is the reason why the SVM is very popular.

Similar to the Least Squares (LS) / Online Learning / Logistic Regression where we have the LMS algorithm, the SVM also can be determined very efficiently, thanks to the fact that there is a Convex optimization problem.

Week 3 : Session 2

(2) Approximate classifier

Sometimes, the points are not linearly separable.



For instance, in the above example, one can never draw a line that completely separates both the circles and triangles on two different sides. Thus, there will be classification error. (ie) Some points in class 0 will be classified as class 1, and other points in class 1 will be classified as class 0. So, it is not possible to linearly separate.

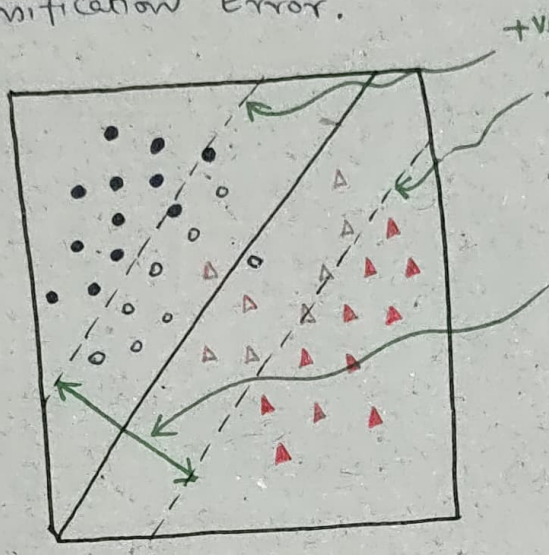
Of course, we can draw something nonlinear, but determining the non-linear classifier is not easy to do.

So, in that sense, when the points are not linearly separable, we employ an approximate classifier (ie) by tolerating some classification error. But, on the whole, we want to minimize the classification error (eg, recall, any learning algorithm doesn't work with 100% accuracy. It might work with 100% accuracy on the training data, but in practice, when we try it on real time data, there will be some classification errors.

So, the classification error / Regression error would be a part of any learning algorithm, coz no learning algorithm is going to be 100% perfect.

So, here, in approximate classifier, we have to allow some classification error during the training phase itself unlike the linear classifier where the training set able to completely classify accurately and we were able to insert a maximum margin slab that perfectly separates the training data.

Thus, the approximate classifier minimizes the classification Error.



In Approximate classifier, there will be some points in the "No man's Land", which we'll call as "Slack".
(ii) The extent of the penetration / deviation into the "No man's land" is known as "slack".

So, essentially what we have to do, is to minimize the slack. (ii) minimize the points that lie on the "No man's land".

Mathematically, this can be represented as

$$C_0: \bar{a}^T \bar{x}_i + b \geq 1 - u_i \quad ; i = 1, 2, \ldots, M$$

$$C_1: \bar{a}^T \bar{x}_i + b \leq -1 + v_i \quad ; i = M+1, \ldots, 2M$$

where u_i, v_i are slack. And slack has to be non-negative. (ii) $u_i, v_i \geq 0$.

(ii) there are no slack for the points that are well within the +ve and -ve Hyperplane. Slack is only for those which cross the +ve Hyperplane for class 0 and -ve Hyperplane for class 1. So, the slack can only be +ve.

$u_i \rightarrow$ Slacks for class 0

$v_i \rightarrow$ Slacks for class 1

Meaning of the word "Slack" → Relaxation, Deviation

(ii) There can be some kind of classification error (ii) they can actually cross over the Hyperplanes. Since the points are not linearly separable, so we allow a certain slack.

Hence, the "SOFT" classifier problem is given as

Minimize Total Slack.

$$\text{(ii) } \min \left(\sum_i u_i + \sum_i v_i \right)$$

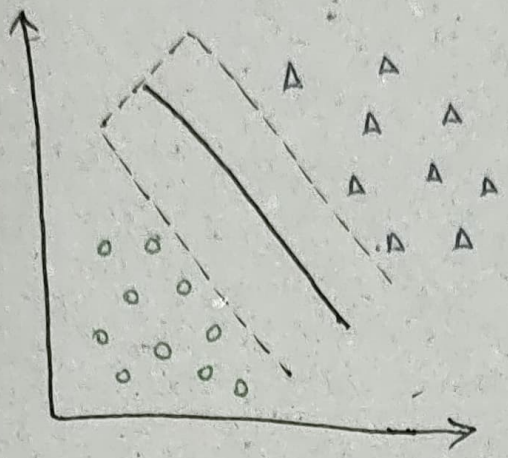
Linear objective function

subject to the constraints

$$\begin{cases} \bar{a}^T x_i + b \geq 1 - u_i \; ; \; i = 1, 2, \ldots, M \\ \bar{a}^T x_i + b \leq -1 + v_i \; ; \; i = M+1, \ldots, 2M \end{cases}$$

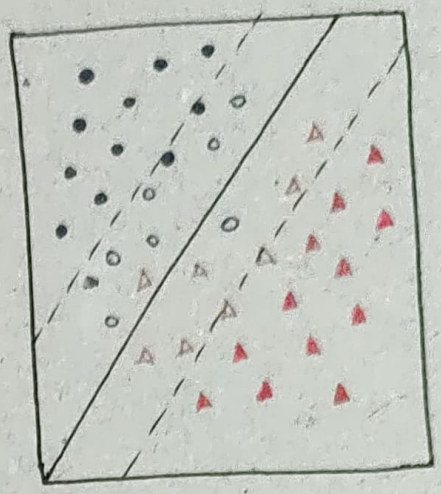
Affine constraints

Since, this is a Convex optimization problem, it can be solved very easily and efficiently.



LINEAR CLASSIFICATION

- AKA Hard classifier
- There is NO classification error (ii) No points in the Margin / slab region



APPROXIMATE CLASSIFICATION

- AKA Soft classifier
- Here, we are tolerating some classification error (ii) There are some points in the Margin

In approximate classification, the Hope is to minimize both the number of points in the Margin and the extent that they penetrate into the margin.

· So, the total slack is not only the total number of points that get into the margin, but also the extent the points get into the margin. Minimizing the number of points in the margin is going to be a very complicated problem, because minimization of number is a highly non-convex problem, which is very difficult to solve. So, interestingly, we have converted that into a problem that minimizes basically the total amount of trespassing. (ii) we are minimizing the extent to which the points are sort of wandering into the region, where they should n't.