"Support Vector Machines" (SVM)

Support vector machines often referred to as SVM's are very popular machine learning techniques.

They can be used for both "Classification as well as Regression purposes"

{ (SVM) → Popular ML → Classification
└→ Regression }

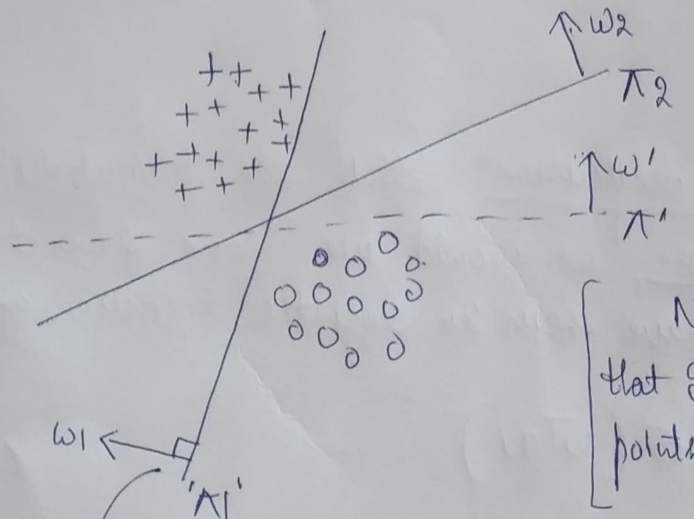
It become extremely popular in the late 1990's.

They are very very elegant & simple algorithms'

Before we go into mathematical details, let's understand the geometric intuition behind "SVM's"

Imagine, we have a dataset comprising of [+ive] & [-ive] points.

{ +ive points are represented using "+"
& -ive points are represented using "o" }



[Let's assume for simplicity that the data is linearly separable]

[Now I have multiple hyperplanes that I can draw to separate these points from one another.

I can draw this hyperplane, let me call the hyperplane as 'π_1'

Correspondingly "w_1" is a vector normal or perpendicular to it.

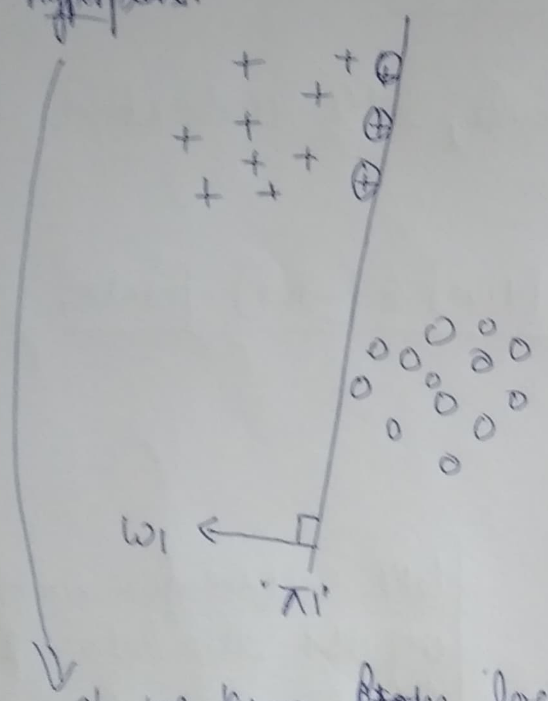
This hyperplane separate these two classes of points.

Similarly if you look at it, there is another hyperplane that separates these two clusters. let's call this hyperplane as "π_2" & "w_2" is a normal to it.

Like these two there are many many hyperplanes possible which separates these two classes (+ve points from -ve points)

When such a thing happens, which of these hyperplanes would you prefer. let's look at it.

Let's take 'π_1' as separating hyperplane. Now if you look closely there are lots of points which are lying very close to the separating hyperplane.

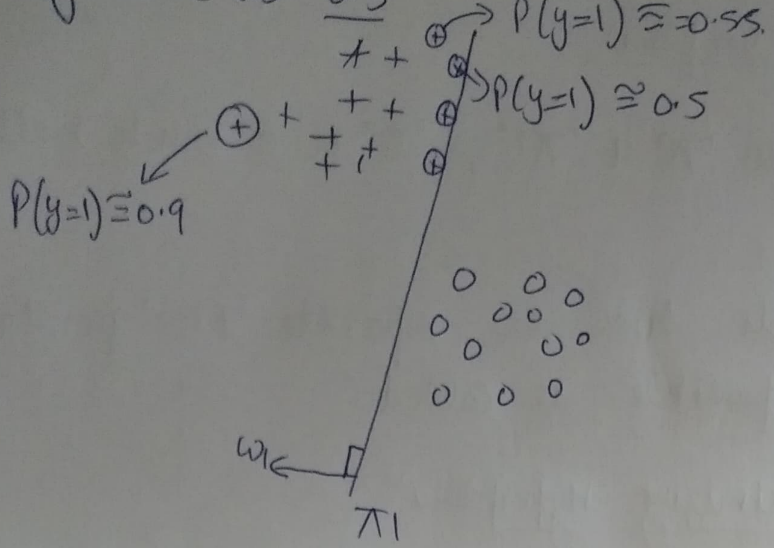


& we know from logistic regression that the probability of a point belonging to '+ve class', we would use some function like sigmoid function as we have seen in logistic regression of the distance

$$P(y_i = 1) = \sigma(w^T x_i)$$

So, if a point is very close for eg. the encircled point you see above. Since they are very very close to hyperplane, if the hyperplane changes slightly, they could get easily misclassified.

So, if a point is lying very close to the hyperplane, then the probability of the point belonging to 'class 1' let's say will be very close to '0.5' (3)



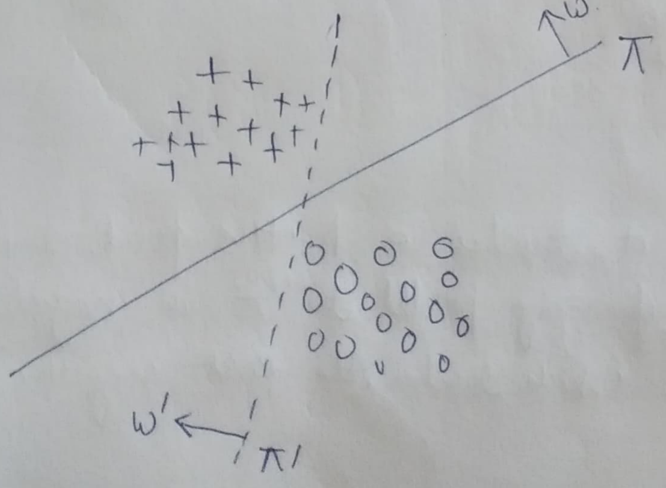
So, what we want is a hyperplane that separates the positive points (+ve) & -ve points as far away as possible)

↓ This is the key geometric idea of SVM's.

So, what SVM's is trying to do, is it is trying to find a hyperplane that separates +ve & -ve points as widely as possible.

Let's understand what does it mean actually from a geometric perspective :-

Let's have a bunch of +ve (+) & -ve points (o) as shown.



If you choose a hyperplane "π" with its normal "w". You will notice that +ve points & -ve points are as far away as possible to this hyperplane as compared to "π'" with normal "w'"

So, we see that $\boxed{\text{"}\pi\text{"}}$ is better than $\boxed{\text{"}\pi'\text{"}}$.

becoz in case of "π'" there are some '+ve' & '-ve' points which are very very close to the 'hyperplane'.

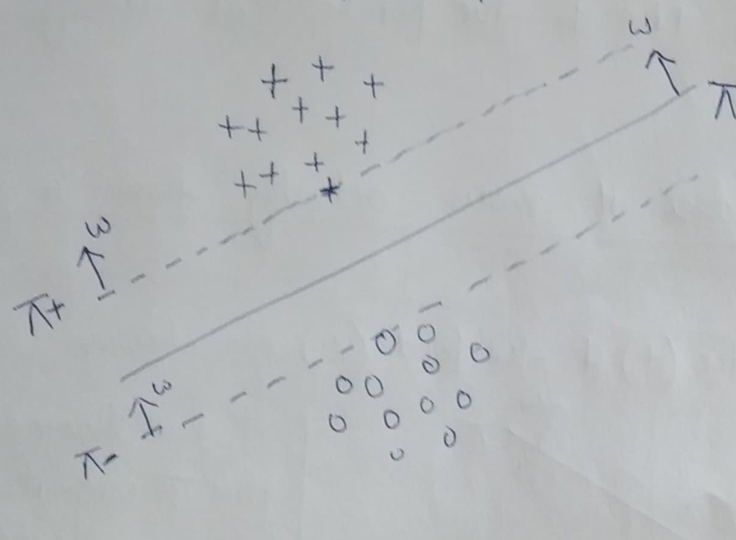
∴ If we have to choose b/w "π" & "π'", "π" is a much better choice.

& Such a hyper plane which tries to separate '+ve' points from '-ve' points as much as possible is called

"Margin Maximizing Hyperplane"

[π: Margin maximizing hyperplane]

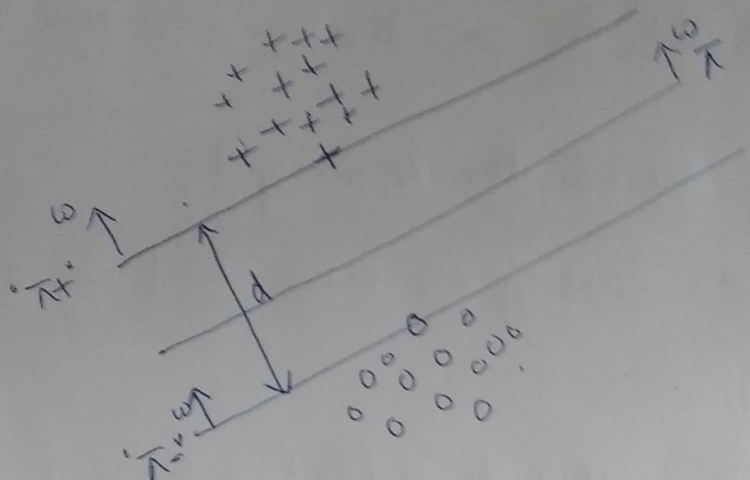
Let's understand Margin maximizing hyperplane, So, if we choose 'π' as a hyperplane which separates the '+ve' & '-ve' points & if we go parallel to the hyperplane which means if we keep on drawing the hyperplanes in +ve direction then at some point we will intersect the first positive point. Lets call this hyperplane as π^+.



π^+ is parallel to π & it touches a positive point & it is parallel to π. Similarly if we keep going parallel to 'π' in negative direction we will get a hyperplane which touches the first negative point. Lets call that as "π^-".

'π_+' is parallel to 'π' and 'π_-' is also parallel to 'π' except that it passes through the -ive point. ⑤

Now the gap between these two, let's denote it by d.



Now since ['π^+'] & ['π^-'] are parallel hyperplanes, they have a constant distance between them.

d = distance (π_+, π_-)

↓ This distance is also called as margin, because it is a margin or a gap b/w '+ive points' & '-ive points'.

So, we want to find a hyperplane 'π' such that, if we go parallel to 'π' & if we touch a positive point with a hyperplane 'π^+' & if we go parallel to 'π' & we have a hyperplane 'π^-' which touches the first '-ive' points, Then the margin that we get, which is basically the distance b/w 'π^+ & π^-', we want to maximize it.

∴ when this distance 'd' is maximized, then the '+ive points' & '-ive points' are quite far away from the actual separating hyperplanes, and they farther they are, the wider the gap, the better it is.

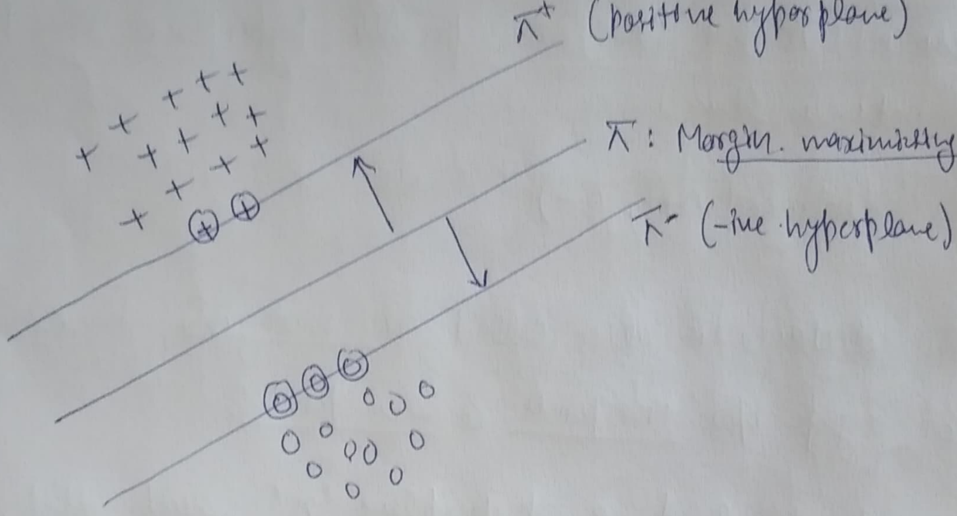
So, what SVM's try to do is, they attempt or try to find a hyperplane "π" that maximizes the [margin = dist(π^+, π^-)].

Note :-> If the margin is high, the chance that we will misclassify will decrease, which means that the generalization error will improve.

as, Margin ↑ ; generalization Accuracy ↑ ;

generalization accuracy means accuracy on unseen datapoints in future. It will also improve.

Another important idea in SVM's is "support vector".



"π^+" goes through ["2 positive points"] & "π^-" goes through ["3 negative points".]

Note :-> The points through with "π^+" & "π^-" passes through are called support vectors.

Given a query point, we will use original "π", the actual separating hyperplane to to classify the query point into "+ive" or "-ive" but we maximize the margin b/w "π^+" & "π^-". So the points which are on either "π^+" or "π^-" are called as "support vectors".

Alternative geometric intuition behind SVM's :->

It is as follows :->

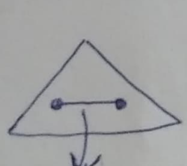
Suppose we have a bunch of "+ve" & "-ve" points. we can find the margin maximizing hyperplane by drawing something known as a "convex hull"

↳ Let's understand a "convex hull". Suppose we have a bunch of points. A convex hull is defined as a polygon that

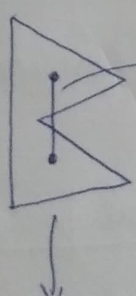
we can draw, such that, it is a smallest polygon, such that all the points are either inside the polygon or on the polygon. & the polygon has to be convex polygon.

Note :- A shape is said to convex. If the line joining any two points inside that shape is also present inside the shape/region.

Eg :-



The line joining these two points is lying inside the triangle hence it a convex polygon.



→ This shape is not convex, because the line joining these points, some part of it is lying outside the region.

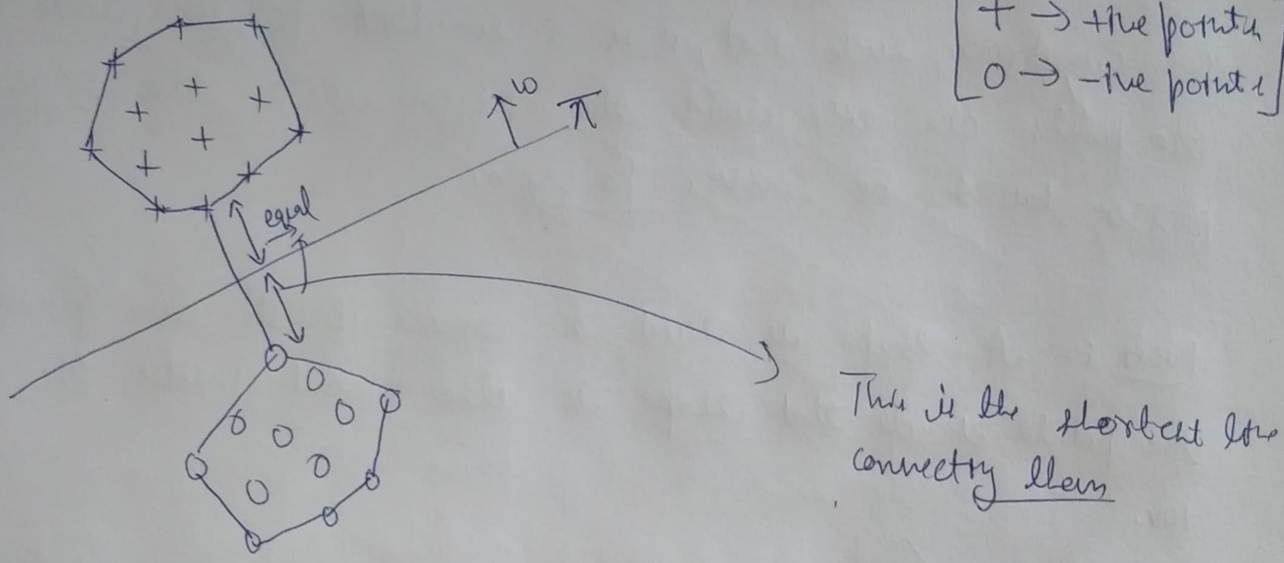
↓ This is called non-convex polygon.

"Convex-polygon" :-> It is a polygon, such that if we want to go from one point to another point, the line connecting both of them is lying inside the polygon.

"Convex-hull" :-> Given a bunch of points, if we can build a smallest convex polygon, that has every point either inside the polygon or on the polygon.

Let's understand the alternative geometric intuition behind behind these SVM's.

Suppose we are given a bunch of positive & negative points.

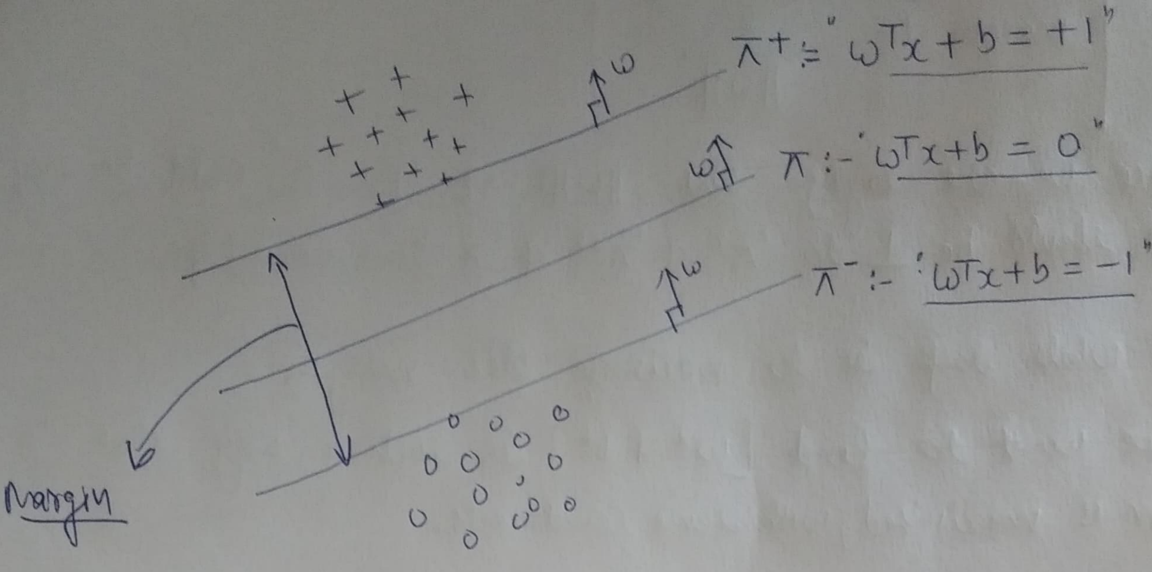


Step ① :-> Construct a convex hull for +ve points.
& Construct a convex hull for -ve points separately.

Step ② :-> Find the shortest line connecting these convex hulls.

Step ③ :-> Bisect the line, (Break it into two equal parts).
The plane that bisects the line is the margin maximizing hyper plane.

→ "Mathematical derivation of SVM"



In the formulation for SVM's the equations that we use to represent the planes are.

$$\begin{bmatrix} \pi: & w^Tx + b = 0 \\ \pi^+: & w^Tx + b = +1 \\ \pi^-: & w^Tx + b = -1 \end{bmatrix}$$

Now the question here is why ["+1"] & ["-1"] on the "R.H.S" of positive & negative hyperplanes.

Note :- Here we are not saying that "w" has to be unit vector we are saying that "w" could be any vector (it could have any length).

All we are saying here is, "w" is $\perp$ to "π" but it could have any length. So the "L2 Norm" or length of "w" need not be "1", it could be any value.

$\{ \; \|w\| \neq 1$ (any length) need not to be a unit vector $\}$

The whole margin could be derived very simply as the distance of a point lying on (π^+ or π^-) from the alternate plane, hyperplane

The margin can be derived as :-

$$\left[\text{Margin :-} \quad \frac{2}{\|w\|} \right]$$

& since we are saying that $\|w\|$ is not "1". It could be any vector.
but it should be $\perp$ to "π" & "π^+" & "π^-" equivalently.

Our whole task is to maximize this margin
So, we want to find $[w^* \, \& \, b^*]$ in such a way that the margin is maximized, with some constraints.

$$\left\{ w^*, b^* = \underset{w,b}{\operatorname{argmax}} \frac{2}{\|w\|} \right\} \longrightarrow \text{objective function of SVM.}$$

And the constraints are all the +ive points should be on one side & all the negative points should be on other side, but the objective is to maximize the margin.

Let's now look at the first approach to prove that "+1" & "-1" on R.H.S doesnot matter.

①. let π^+ : $w^T x + b = K$
& π^- : $w^T x + b = -K$

The only requirement is that "K" should be greater than "0" $(K > 0)$ only then π^+ & π^- are away from π

& why are we taking "+K" & "-K" here. &
Why cant we take "$+K_1$ & $-K_2$". The reason is we want both of these +ive & -ive hyperplanes to be equally far away from the margin hyperplane..

Note :-> "K" could be any number, as long as it is greater than "0"

P.T.O

Now when we have this, the margin will change to

margin :- $\frac{2k}{\|w\|}$

Here 'K' is a constant, It could have any value.

If this is the margin, then from optimization's point of view, we know that the 'w & b' which maximizes $\left[\frac{2}{\|w\|}\right]$ is the same as the 'w & b' which maximizes $\left[\frac{2k}{\|w\|}\right]$

$$\left\{ \underset{w,b}{\arg\max} \frac{2}{\|w\|} = \underset{w,b}{\arg\max} \frac{2k}{\|w\|} \right\}$$

Let $k = 4$. then we have

$$\underset{w,b}{\arg\max} \frac{2}{\|w\|} = \underset{w,b}{\arg\max} \frac{8}{\|w\|}$$

So, from optimization's point of view, if we take '1' or 'K' it really doesnot matter.

So, we only took '1' here as a way of convenience, Nothing will change.

—o—