The kernel trick

We have seen the dual form of ~~the~~ SVM:

$$\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{x_i^T x_j}$$

$$s.t. \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0$$

$\boxed{x_i^T x_j}$ → is the function of x_i and x_j

We can use any similarity function of x_i and x_j

→ This is also **Kernel function**

→ It is often replaced by $K(x_i, x_j)$

Then we can write the query function as:

$$f(x_q) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x_q) + b$$

So, the replacement of $x_i^T x_j$ with $K(x_i, x_j)$ is called as **The kernel trick**

The most important idea in svm is the kernel trick.

If you don't apply the kernel trick, then it is called as Linear SVM.

→ There are many types of kernel functions available, and we are going to see few of them.

→ This is about the kernel trick, but the main reason why we do this?

We will see in a second.

Ok. so far what we have done is

Linear svm: $x_i^T x_j$

Kernel svm: $K(x_i, x_j)$

In linear svm, we are trying to ~~form~~ find margin maximizing hyper-plane, in space of x_i's.

In logistic Regression, we are trying to find hyperplane that minimizes the logistic loss, in the space of x_i's.

Most of the time, when you apply Logistic-Reg. and SVM the result looks very-very similar. (not always)

So, margin maximizing is a nice idea, but not the world changing idea.

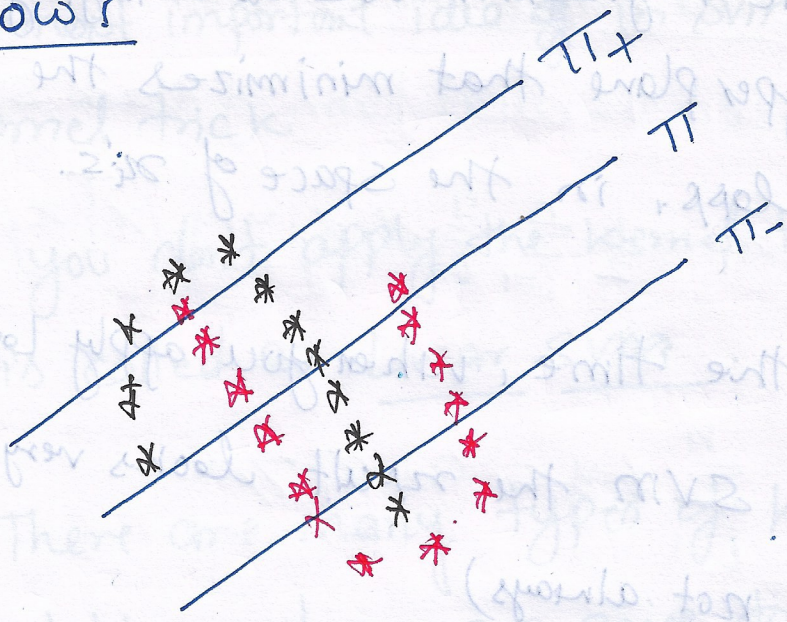
The world changing or the key idea of SVM is the kernel trick or kernalization.

Actually kernalization makes SVM to handle non-linearly seprable datasets. (how?) (we will see.)

You can think of kernalization kindof feature transformed Logistic Reg. (but they are different)

There is also kernel-logistic Regression in Research

How?



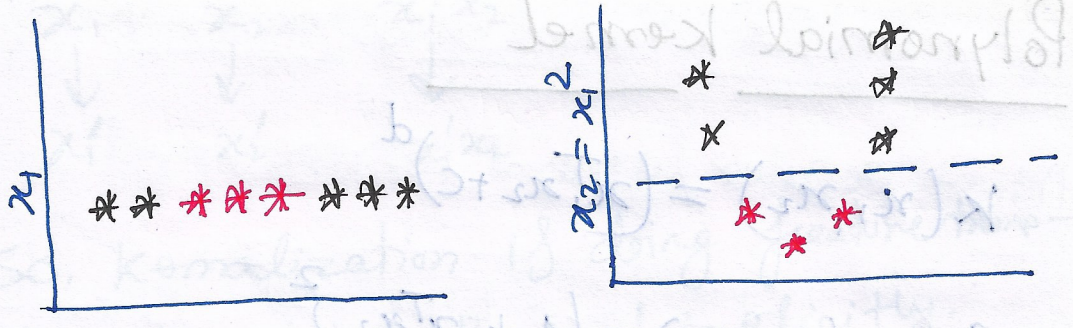
Most of the time, the real-life data pets are not nice, linearly separable.

Say I'm having moon-shaped data.

How to get the hyper plane.?

Mapping to higher dimensions

→ The most fundamental principal of kernel trick is that. "In a high dimensional space, there is a bigger chance the data will become linearly separable".



From the above pair plot we can see that, as the dimension increases, we can separate the data.

When we've only one feature, x_1 we can not separate the data with single line.

Adding another feature $x_2 = x_1^2$, makes it easy to separate the two classes.

So, Kernalization is basically a smart way of adding more features to the data in the hope of making linearly & separable.

Some popular kernels are
- Polynomial ker.
- Gaussian Radial Basis Function RBF kernel

Polynomial kernel

$$K(x_1, x_2) = (x_1^T x_2 + c)^d$$

e.g. $K(x_1, x_2) = (1 + x_1^T x_2)^2$

↳ Quadratic kernel

$c = 1, d = 2$

$\therefore K(x_1, x_2) = (1 + x_1^T x_2)^2$ $\quad x_1 = \langle x_{11}, x_{12}$

$x_2 = \langle x_{21}, x_2$

$= (1 + x_{11}x_{21} + x_{12}x_{22})^2$

$\Rightarrow K(x_1, x_2) = 1 + x_{11}^2 x_{21}^2 + x_{12}^2 x_{22}^2 + 2x_{11}x_{21}$

$+ 2x_{11}x_{22} + 2x_{11}x_{21}x_{12}x_{22}$

$= [1, x_{11}^2, x_{12}^2, \sqrt{2}x_{11}, \sqrt{2}x_{12},$

$\sqrt{2}x_{11}x_{12}] : x_1'$

$[1, x_{21}^2, x_{22}^2, \sqrt{2}x_{21} + \sqrt{2}x_{22}, \sqrt{2}x_{21}x_{22}] :$

$\therefore K(x_1, x_2) = (x_1')^T (x_2')$

Basically, what we have done is:

x_1 → x_1' ; x_2 → x_2' ; $x_1^T x_2$ → $x_1' x_2'$

So, kernalization is doing feature transformation internally and implicitly.

Kernalization $d \xrightarrow[\text{internally implicitly}]{FT} d'$ $\quad d' > d$

But the challange here is select the right kernel.