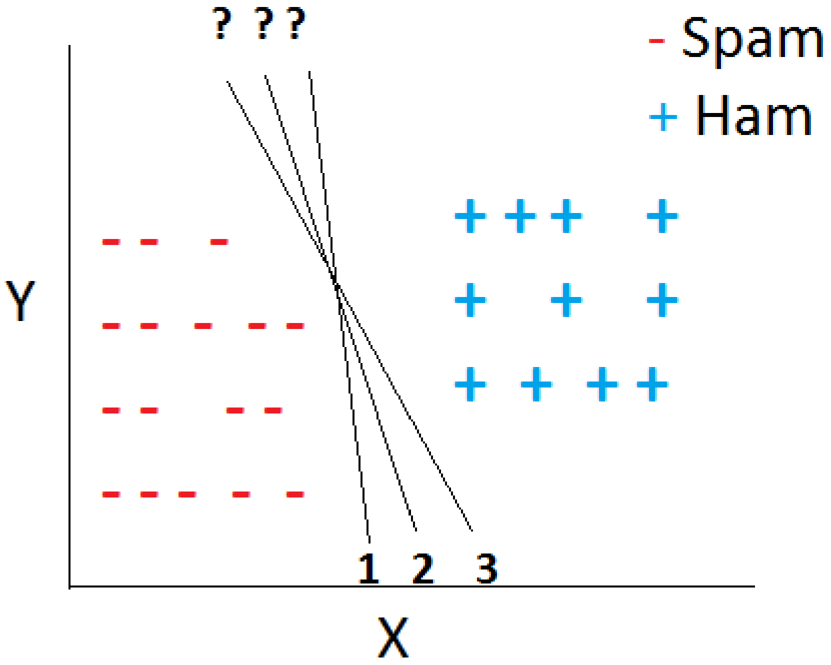
How is Soft Margin Classifier different from Maximum Margin Classifier?

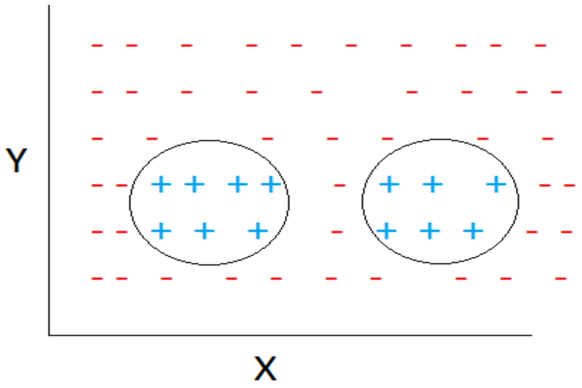
Answer 1-

Maximal margin classifier separates two classes with multiple possible Hyperplanes. Among these Hyperplanes the best line is that which maintains maximum distance between the nearest point of both the classes. We can see it clearly in below figure-



Soft margin classifier also called Support vector classifier. It works well with intermingled data. It allows certain points to be misclassified that leads to classify most of the unseen data points, so it called more robust in this term.

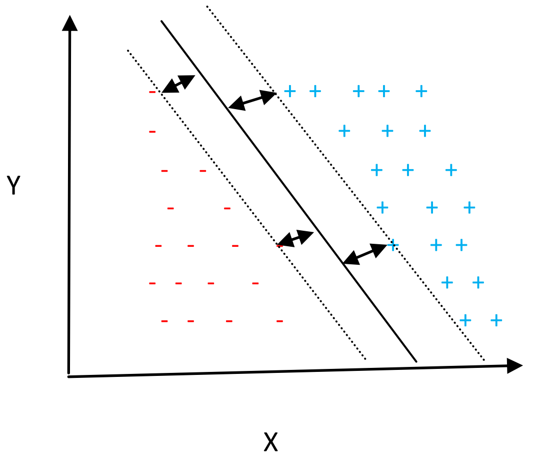
Soft margin classifier allows some observations to fall on the wrong side, so only nearest point of Hyperplane being considered for constructing the Hyperplane.



What does the slack variable Epsilon (ε) represent?

Answer-2

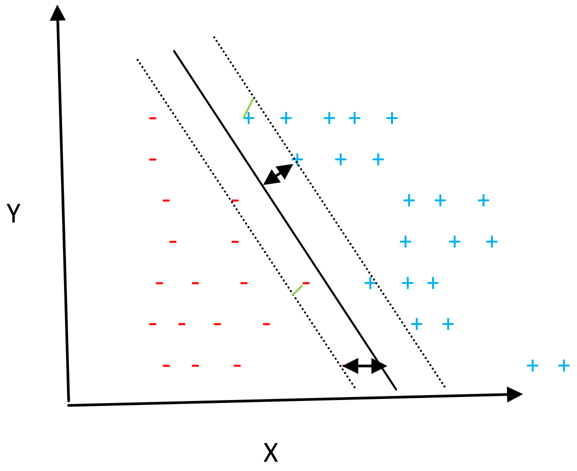
Slack variable represents about relativity of an observation to the margin and Hyperplane. It is used to control misclassifications. Slack variable is represented by Epsilon(ϵ).if Epsilon value is equal to zero, that means support vector classifier doesn't allow any misclassification and each observation is on correct side of the margin. You can see it in below figure-



ϵ = 0

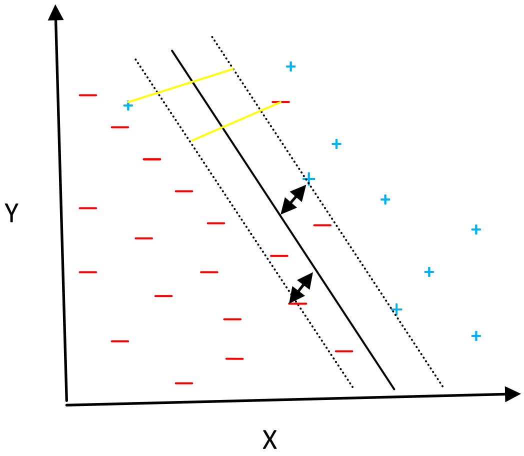
Slack variable

But if you draw a Support Vector Classifier in such a way that it only violates the margin, i.e. 0< Epsilon(ϵ) < 1, the observations classify correctly as shown in figure below.



0<ϵ <1

But if the data points violate the hyperplane, i.e. Epsilon(ϵ) > 1, then the observation is on the wrong side of the hyperplane, as shown in figure below.



(ϵ) > 1

Lower values of slack are better than higher values (slack = 0 implies a correct classification, but slack > 1 implies an incorrect classification, whereas slack within 0 and 1 classifies correctly but violates the margin.

Q3-How do you measure the cost function in SVM? What does the value of C signify?

Answer-3

Cost of misclassification is greater than or equal to the summation of all the epsilons of each data point, and is denoted by cost or 'C'.

$$\Sigma\varepsilon_i <= C$$

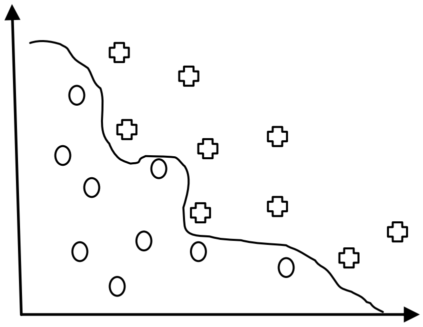
We can measure the summation of all the epsilons(ϵ) of both the hyperplanes and choose the best one that gives you the least sum of epsilons(ϵ). The summation of all the epsilons of each data point is denoted by cost or 'C', i.e.
When **C is large**, the **slack variables** can be large, i.e. you allow a larger number of data points to be misclassified or to violate the margin. So you get a hyperplane where the margin is wide and misclassifications are allowed. In this case, the model is flexible, more generalisable, and less likely to overfit. In other words, it has a high bias.
On the other hand, when **C is small**, you force the individual slack variables to be small, i.e. you do not allow many data points to fall on the wrong side of the margin or the hyperplane. So, the margin is narrow and there are few **misclassifications**. In this case, the model is less flexible, less generalisable, and more likely to overfit. In other words, it has a **high variance**.

Q-4 Given the above dataset where red and blue points represent the two classes, how will you use SVM to classify the data?

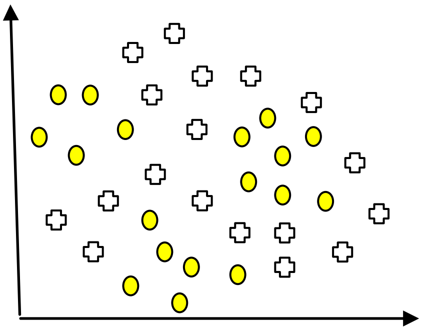
Answer-4

We can clearly see that given figure is Non-Linear Data. We can solve this by transforming Non-Linear data to linear data. SVM uses 2 kernels for it- polynomial kernel and RBF kernel.

1-Polynomial kernel- It is capable of creating nonlinear, polynomial decision boundaries.



2- The RBF(Radial Basis Functional) kernel- intermingled complex data is being handled by RBF kernel.



So this complex non-linear dataset can be classified using RBF kernel. It is even capable of creating elliptical (i.e. enclosed) decision boundaries.

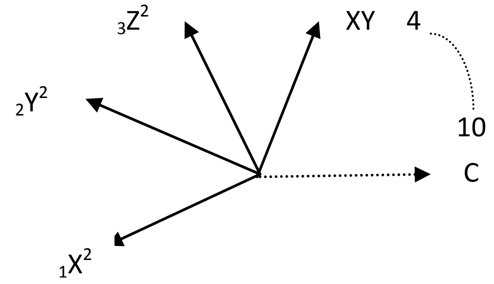
Q-5 What do you mean by feature transformation?

Answer-5

The process of transforming the original attributes into a new feature space is called 'feature transformation'. However as the number of attributes increases, there is an exponential increase in the number of dimensions in the transformed feature space.

Suppose you have three variables in your data set, then considering only a polynomial transformation with degree 2, you end up making **10 features** in the new feature space, as shown in the figure below.

$$a_1X^2 + a_2Y^2 + a_3Z^2 + a_4XY + a_5YZ + a_6ZX + a_7X + a_8Y + a_9Z + C = 0$$



$_3Z^2$
XY
4
$_2Y^2$
10
C
$_1X^2$