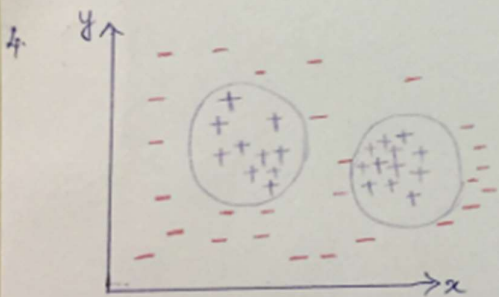
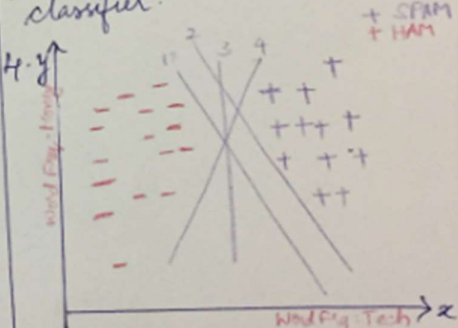


Subjective Questions – Assignment: Support Vector Machines

Question 1

How is Soft Margin Classifier different from Maximum Margin Classifier?

Answer 1: Below listed points explain the difference between the soft margin classifier and the maximum margin classifier.

SOFT MARGIN CLASSIFIER	MAXIMUM MARGIN CLASSIFIER
1. The support vector classifier or the soft margin classifier essentially allows certain points to be <u>deliberately misclassified</u> .	1. There could be multiple hyperplanes possible which <u>perfectly separate</u> the <u>two classes</u> .
2. Soft margin classifier allows some observations to fall on wrong side. The points which are close to the hyperplane are only considered for constructing hyperplane. Called as <u>support vectors</u> .	2. The best line is the one which maintains the <u>largest possible equal distance</u> from the nearest points of both the classes so for the separator to be optimal.
3. Support vector classifier works well when the data is partially intermingled.	3. The margin or the distance of the nearest point to the separator should be maximum. This is called maximal margin classifier.
	
5. Soft margin classifier, classifies most of the points <u>correctly</u> in the <u>unseen data</u> and is also more robust.	5. Maximal margin classifier will perform perfectly on training data set. But on the <u>unseen data</u> , it may perform <u>poorly</u> .
6. To select the <u>best fit</u> support vector classifier the <u>slack variables</u> are used to control misclassifications.	

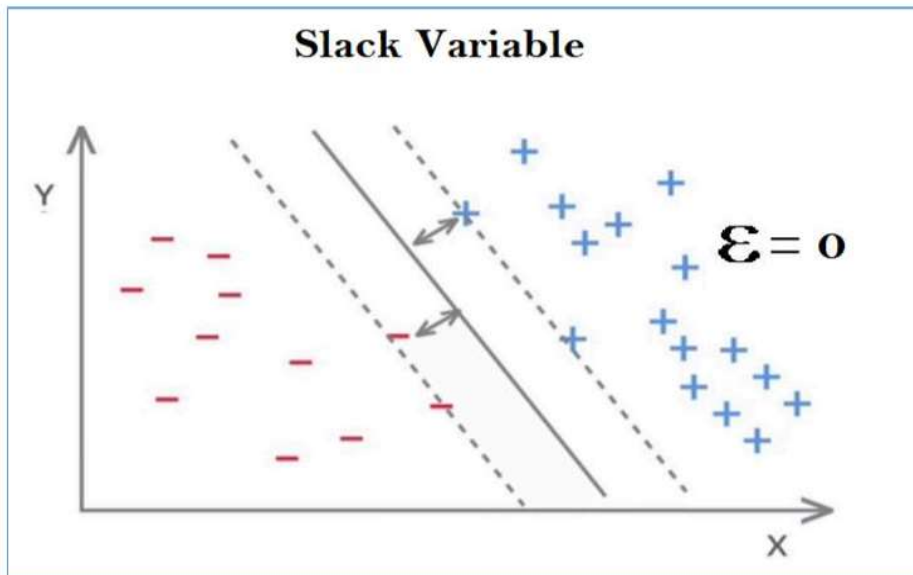
Subjective Questions – Assignment: Support Vector Machines

Question 2 : What does the slack variable Epsilon (ϵ) represent?

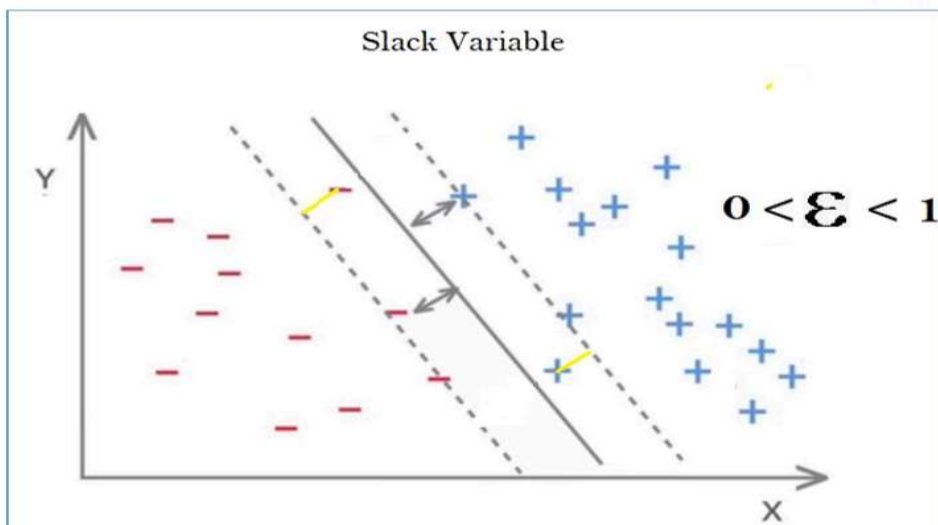
Answer 2: To select the best-fit Support Vector Classifier, the notion of slack variables (epsilons(ϵ)) can help in comparing the classifiers. There is also a concept of the slack variable(ϵ).

A slack variable is used to control misclassifications. It tells you where an observation is located relative to the margin and hyperplane.

There are three different conditions applied if any new data point comes into play. Suppose you draw a Support Vector Classifier in such a way that it doesn't allow any misclassification, i.e. $\text{Epsilon}(\epsilon) = 0$, then each observation is on the correct side of the margin as shown in figure below.

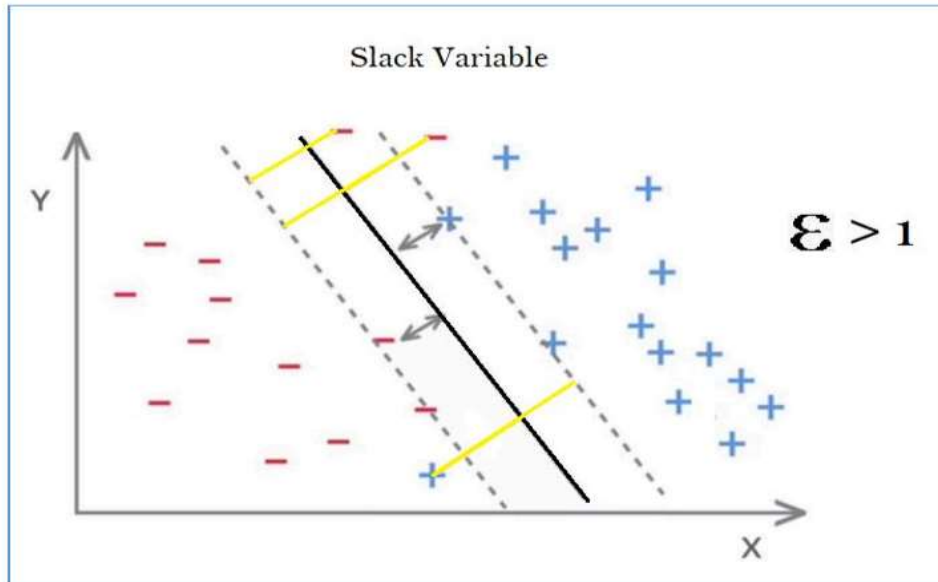


But if you draw a Support Vector Classifier in such a way that it only violates the margin, i.e. $0 < \text{Epsilon}(\epsilon) < 1$, the observations classify correctly as shown in figure below.



Subjective Questions – Assignment: Support Vector Machines

But if the data points violate the hyperplane, i.e. $\epsilon > 1$, then the observation is on the wrong side of the hyperplane, as shown in figure below.



So, you can see that:

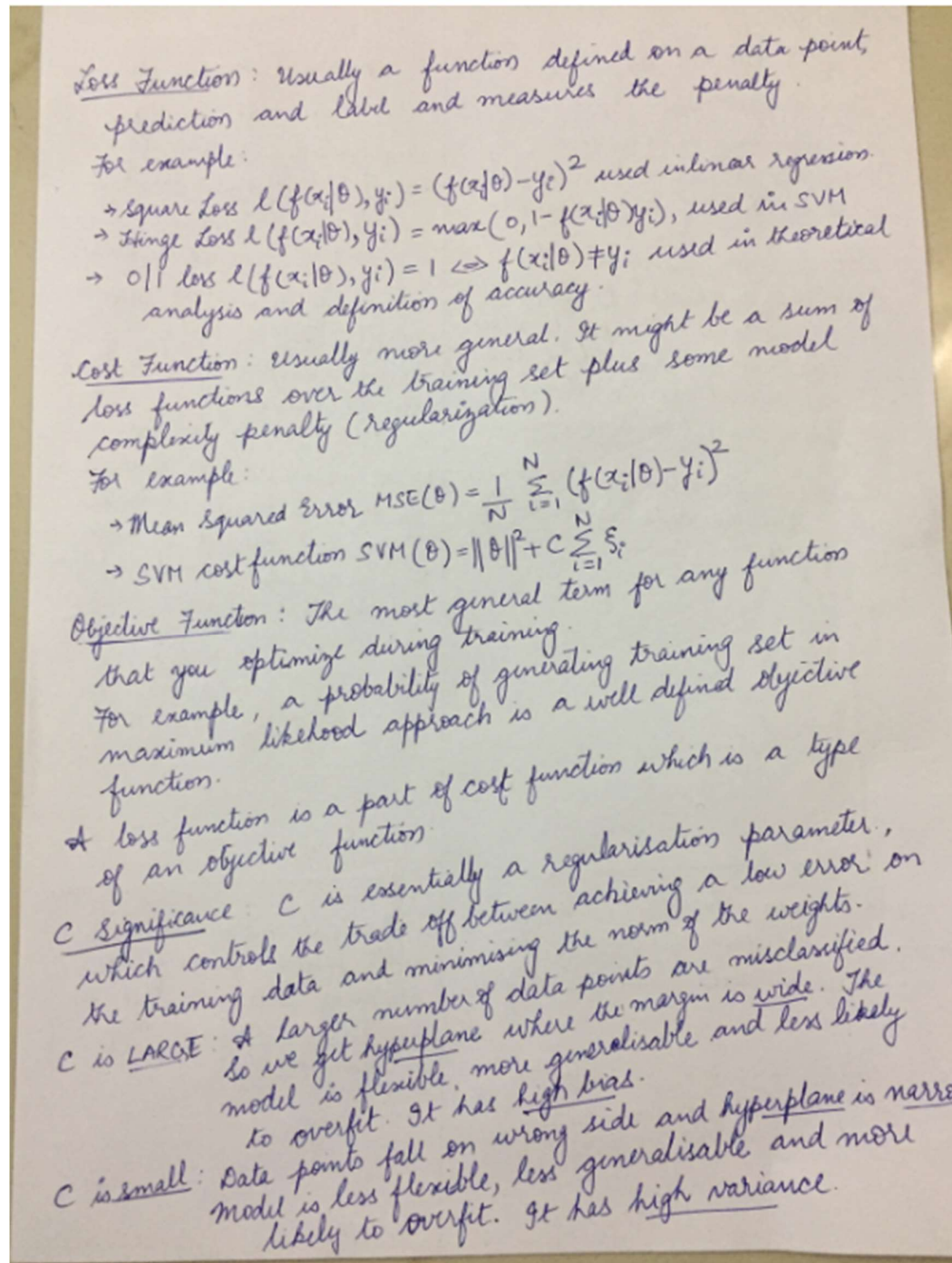
- Each data point has a slack value associated to it, according to where the point is located.
- The value of slack lies between 0 and +infinity.

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).

Subjective Questions – Assignment: Support Vector Machines

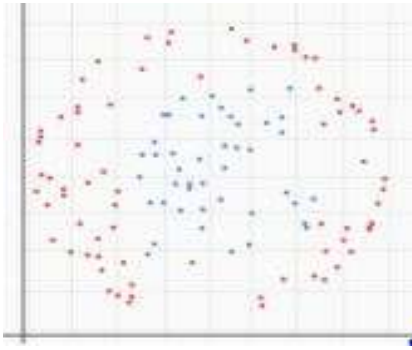
Question 3 : How do you measure the cost function in SVM? What does the value of C signify?

Answer 3: The cost function is explained below in detail along with the value of significance of C in SVM.



Subjective Questions – Assignment: Support Vector Machines

Question 4



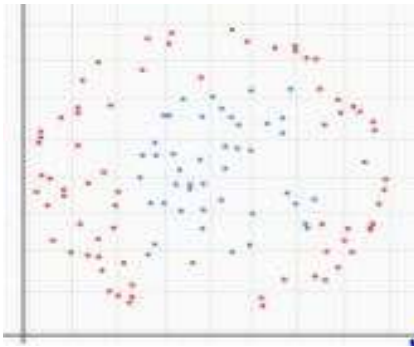
Given the above dataset where red and blue points represent the two classes, how will you use SVM to classify the data?

Answer 4: Kernels are one of the most interesting inventions in machine learning, partly because they were born through the creative imagination of mathematicians, and partly because of their utility in dealing with non-linear datasets. The above dataset with red and blue points can be classified by tweaking the linear SVM model using Kernel. They enable the linear SVM model to separate nonlinearly separable data points.

Mapping Nonlinear Data to Linear Data:

You can transform nonlinear boundaries to linear boundaries by applying certain functions to the original attributes. The original space (X, Y) is called the attribute space, and the transformed space (X', Y') is called the feature space.

Let's say you want to classify emails into 'spam' or 'ham' on the basis of two attributes — 'word_freq_office' (X) and 'word_freq_lottery' (Y). The following plot shows the data set below, which is clearly nonlinear



To convert this data set into a linearly separable one, a simple transformation into a new feature space (X', Y') can be made. You may almost never need to manually transform data sets. Just assume that some appropriate transformation from (X, Y) to (X', Y') can make the data linearly separable.

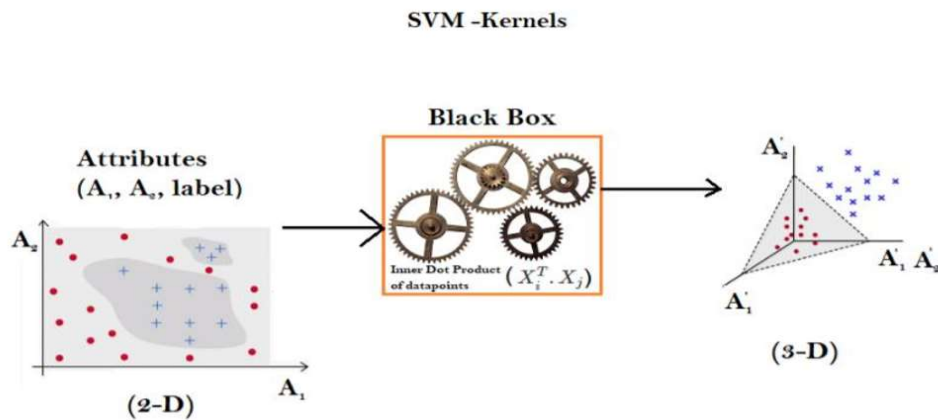
In the original attribute space, notice that the observations are distributed in a circular fashion. This gives you a hint that the transformation should convert the circular distribution to a linear distribution.

$$\text{word_freq_office}(X') = (\text{word_freq_office}(X) - a)^2$$

Subjective Questions – Assignment: Support Vector Machines

$$\text{word_freq_office}'(Y') = (\text{word_freq_office}(Y) - b)^2$$

So the above data set would be transformed into linear dataset using Kernels.



The three most popular types of kernel functions are:

- The linear kernel: This is the same as the support vector classifier, or the hyperplane, without any transformation at all
- The polynomial kernel: It is capable of creating nonlinear, polynomial decision boundaries
- The radial basis function (RBF) kernel: This is the most complex one, which is capable of transforming highly nonlinear feature spaces to linear ones. It is even capable of creating elliptical (i.e. enclosed) decision boundaries

Question 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 four variables in your data set, then considering only a polynomial transformation with degree 2, you end up making 15 features in the new feature space, as shown in the figure below.

