

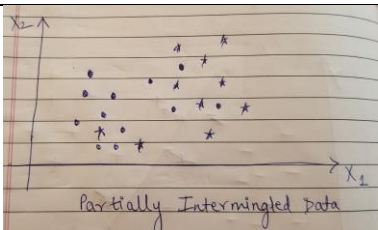

SVM Assignment – Part II

Submission By: Barkha Garg

Question 1

How is Soft Margin Classifier different from Maximum Margin Classifier?

Answer:

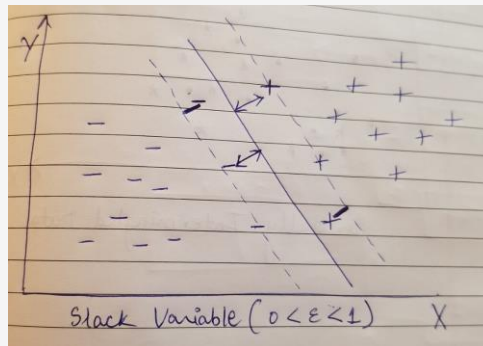
S.No.	Soft Margin Classifier	Maximum Margin Classifier
1	This model can work on data that is not completely Linearly separable i.e, the data points are intermingled in the two categories	This model can only work on data that is not completely Linearly separable. The Hyperplane should be able to divide the data perfectly with each class lying on one side of the Hyperplane
2	This Classifier does allow some data points to be classified incorrectly and hence supports classification for intermingled data	This classifier does not allow any data point to be classified incorrectly.
3	It uses the concept of slack variable(ϵ). A value >1 signifies an incorrect classification, while a value between 0 and 1 signifies a correct classification. A lower value of slack variable is desired.	Since it does not allow any misclassification, no concept of slack variable
4	This is practically more useful as most of the data sets in real life are not perfectly linearly separable.	This is practically less popular and has very limited applicability as hardly a few real life datasets are perfectly linearly separable
5		

Question 2

What does the slack variable Epsilon (ϵ) represent?

Answer:

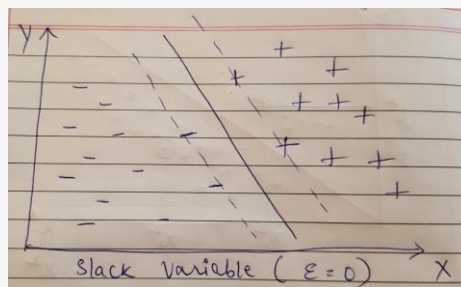
The slack variable bears relevance in case of Soft Margin Classifier (Support Vector Classifier) where the data points are partially intermingled among the two categories to be classified and hence the data is not perfectly separable.



The slack variable (denoted by ϵ), measures whether a data point is classified correctly or not. A value like $0 < \epsilon < 1$ signifies a correct classification of the data point while $\epsilon > 1$ indicates a misclassification of the data point. The slack variable gives the position of a data point relative to the margin and the Hyperplane.

The higher value of ϵ would mean a lot of incorrect classifications and hence a lower value is desired.

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).



Question 3

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

Answer:

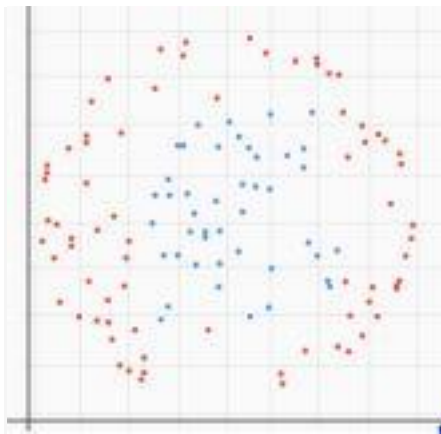
The cost function in case of SVM is the summation of all epsilons (slack variable, ϵ) for individual data points. When comparing the 2 models, the model with lower value of this summation is preferred. This summation is denoted by C, the cost function

$$C = \sum \epsilon_i$$

When **C is large**, the slack variables can be large, i.e. we allow a larger number of data points to be misclassified or to violate the margin. So our hyperplane has a wide margin and misclassifications are allowed. In this case, the model is **flexible, more generalisable, and less likely to overfit**. So, it has a **high bias**.

In contrast, when the value of C is small, the individual slack variables are forced to be small, i.e. not many data points are allowed 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**. So, it has a **high variance**.

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:

Clearly, the data set given above is not linearly separable. This is a non-linear data. We will have to do some transformation to the actual features in a way that the transformed features are in a space that is separable using some linear/non-linear function. We have to introduce some form of non-linearity to the code separate the data given into 2 classes.

So first we have to do feature transformation. However manually doing a feature transformation is exhaustive and computationally expensive as the number of dimensions may explode exponentially with number of features, Hence we can use Kernels which will do this transformation for us implicitly.

We can try out Linear, Polynomial, or RBF (Radial Bias Function) kernels to see which technique produces the best model.

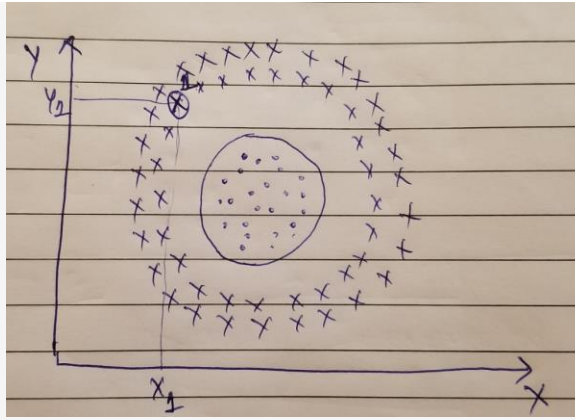
In a non-linear kernel, such as the RBF kernel, we need to choose two tuning parameters: **gamma** and 'C'. The hyperparameter **gamma controls the amount of non-linearity** in the model - as gamma increases, the model becomes more non-linear, and thus model complexity increases. And C is cost functions that controls the number of miss-classifications acceptable.

Question 5

What do you mean by feature transformation?

Answer:

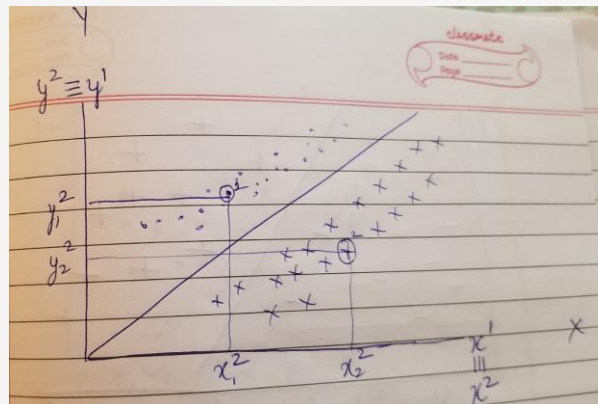
Formally, Feature transformation is the process of transforming the original attributes in a new feature space. These new features may not have the exact same interpretation in original space but may have more discriminatory/explanatory power in the transformed space. This technique is used when the problem we are trying to solve cannot be solved with the current feature space the data is in. For example



As we see in the above picture, the data appears to be distributed in random pattern (circular precisely). It is not possible to predict anything out of it using any linear transformation. To address this problem, we can perhaps transform the features to another space (maybe a higher dimensional) where the features can be represented as some function that can be solved using a Linear/Non-Linear function. For above example it can be

$$\frac{x^2}{a} + \frac{y^2}{b} = c$$

Which will make the data look as below in the new space:



However, with feature transformation the number of dimensions may explode exponentially. This downside can be handled using Kernels.