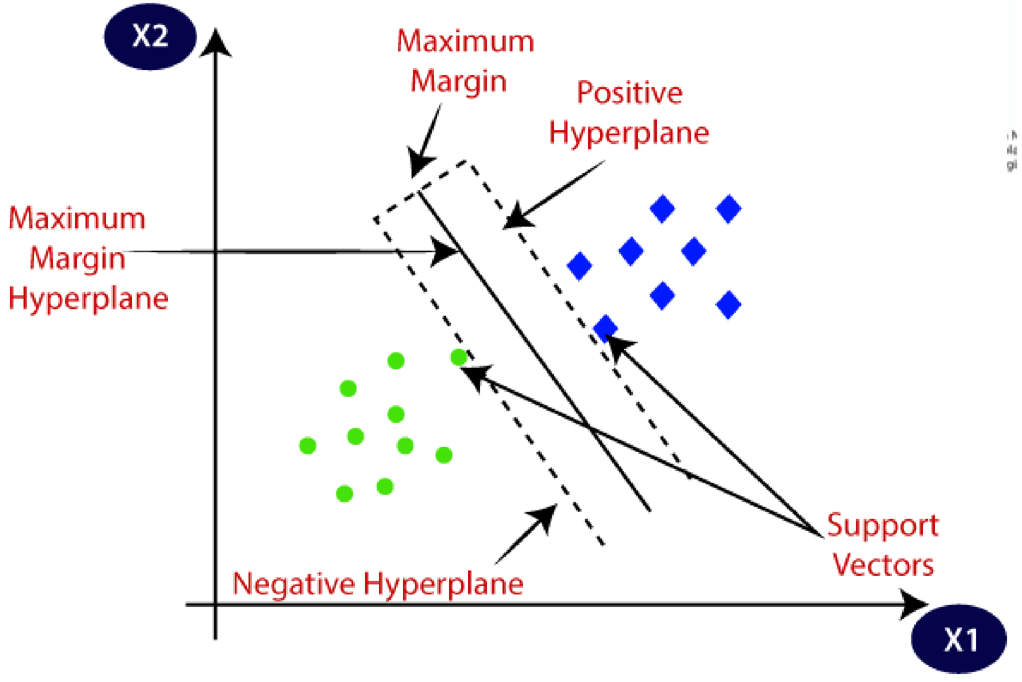
Support Vector Machine (SVM)

Conducted by:
Dr. Amran Hossain
Professor
CSE,DUET

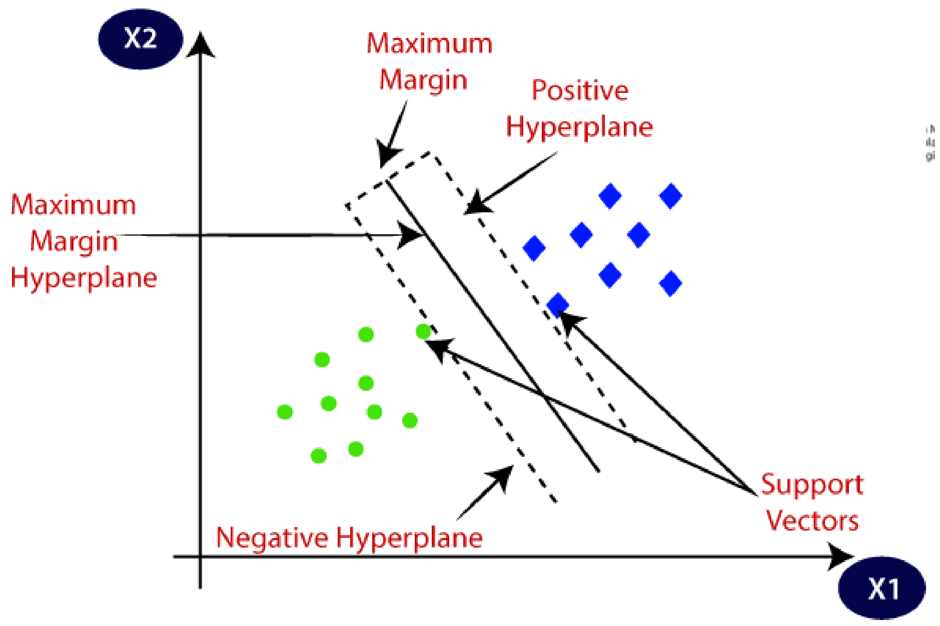
Support Vector Machine(SVM) Algorithm

- Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression.
- Though we say regression problems as well it's best suited for classification.
- The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space.
- The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible.
- The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line.
- If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.



Support Vector Machine Algorithm

- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.
- **Support Vectors**
  - Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.
- **Hyperplane**
  - A hyperplane is a decision plane which separates between a set of objects having different class memberships.
- **Margin**
  - A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

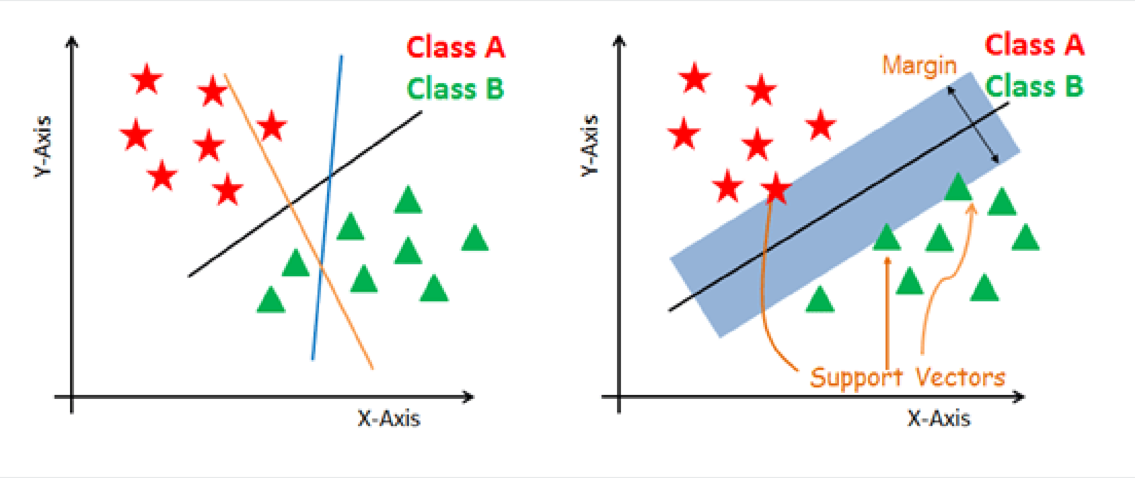


Types of SVM

- **SVM can be of two types:**
  - **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
  - **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

How does SVM work?

- The main objective is to segregate the given dataset in the best possible way. The distance between the either nearest points is known as the margin.
- The objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset. SVM searches for the maximum marginal hyperplane in the following steps:
    1. Generate hyperplanes which segregates the classes in the best way. Left-hand side figure showing three hyperplanes black, blue and orange. Here, the blue and orange have higher classification error, but the black is separating the two classes correctly.
    2. Select the right hyperplane with the maximum segregation from the either nearest data points as shown in the right-hand side figure.



Linear SVM

- The working of the SVM algorithm can be understood by using an example.
- Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the Right Figure image-1.
- So, as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the Right image 2:
- Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors.
- The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.

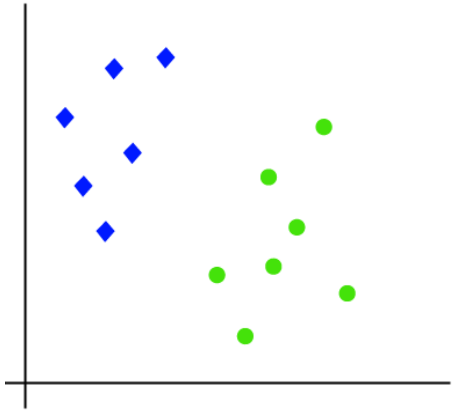


Fig-Image-1

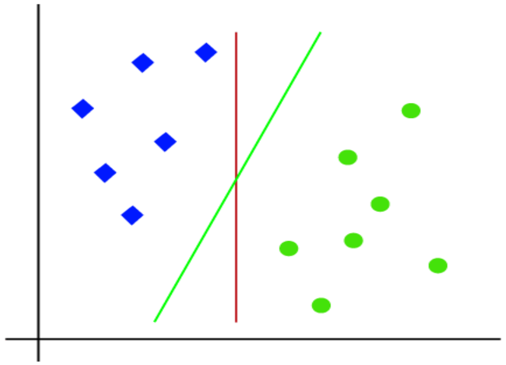


Fig-Image-2

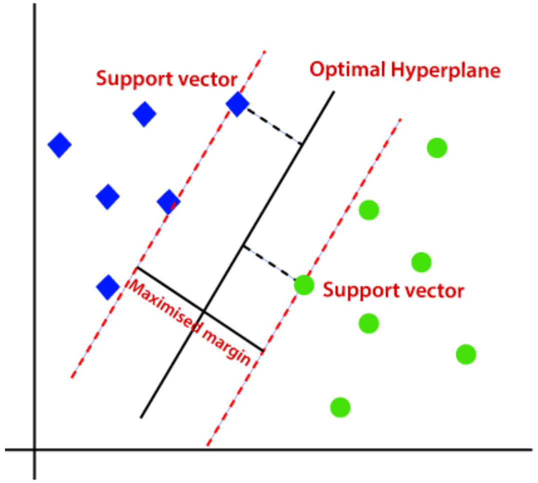


Fig-Image-3

Non-Linear SVM

- If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image 1:
- So, to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

$$z=x^2+y^2$$

By adding the third dimension, the sample space will become as below image 2:

So now, SVM will divide the datasets into classes in the following way. Consider the below image 3:

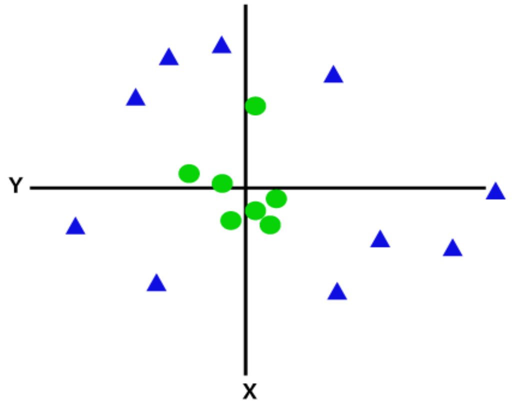


Image-1

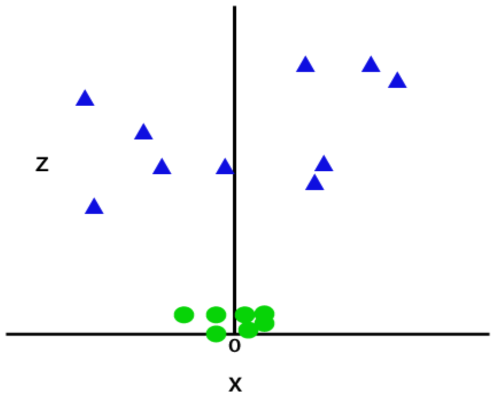


Image-2

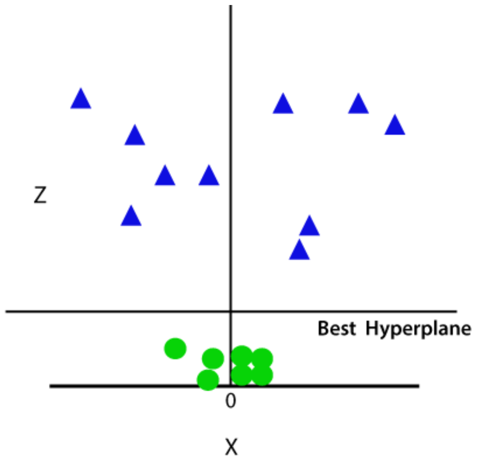
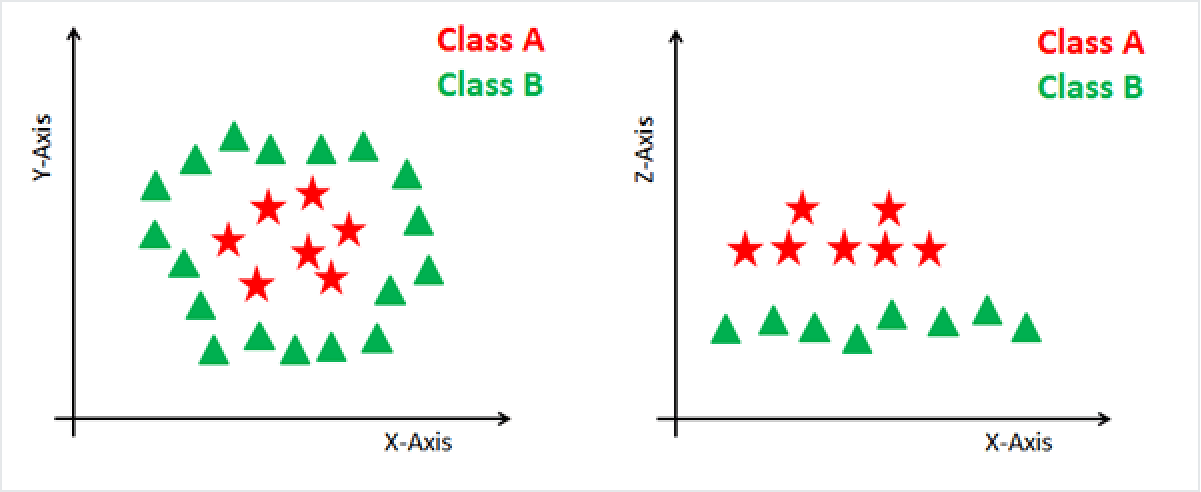


Image-3

Non-Linear SVM

- **Dealing with non-linear and inseparable planes**
- Some problems can't be solved using linear hyperplane, as shown in the figure below (left-hand side).
- In such situation, SVM uses a kernel trick to transform the input space to a higher dimensional space as shown on the right.



SVM Kernels

- The SVM algorithm is implemented in practice using a kernel. A kernel transforms an input data space into the required form. SVM uses a technique called the kernel trick.
- Here, the kernel takes a low-dimensional input space and transforms it into a higher dimensional space. In other words, you can say that it converts nonseparable problem to separable problems by adding more dimension to it.
- It is most useful in non-linear separation problem.
- Kernel trick helps you to build a more accurate classifier.

SVM Kernels

- **Linear Kernel** A linear kernel can be used as normal dot product any two given observations. The product between two vectors is the sum of the multiplication of each pair of input values.

```
K(x, xi) = sum(x * xi)
```

- This code defines a function called K that takes in two arguments **x** and **xi**.
- The function calculates the dot product of **x** and **xi** by multiplying each element of **x** with the corresponding element of **xi** and then summing up the products.
- The **sum()** function is used to perform the summation.
- Note that this code is written in Julia, a high-level dynamic programming language designed for numerical and scientific computing.

SVM Kernels

- **Polynomial Kernel** A polynomial kernel is a more generalized form of the linear kernel. The polynomial kernel can distinguish curved or nonlinear input space.

```
K(x,xi) = 1 + sum(x * xi)^d
```

- This code defines a function called **K** that takes two input vectors **x** and **xi**.
- The function calculates the dot product of **x** and **xi** using the * operator and raises it to the power of **d**.
- It then adds 1 to the result and returns it.
- The ^ operator is used for exponentiation.
- The value of **d** is not specified in this code snippet and would need to be defined elsewhere in the code.
- This code is written in Julia.

Where d is the degree of the polynomial. d=1 is similar to the linear transformation. The degree needs to be manually specified in the learning algorithm.

SVM Kernels

- **Radial Basis Function(RBF) Kernel** The Radial basis function kernel is a popular kernel function commonly used in support vector machine classification.
- RBF can map an input space in infinite dimensional space.

```
K(x,xi) = exp(-gamma * sum((x - xi^2))
```

Here gamma is a parameter, which ranges from 0 to 1.
A higher value of gamma will perfectly fit the training dataset, which causes over-fitting.
Gamma=0.1 is considered to be a good default value.
The value of gamma needs to be manually specified in the learning algorithm

Advantages and Disadvantages of SVM

- **Advantages**
  - SVM Classifiers offer good accuracy and perform faster prediction compared to Naïve Bayes algorithm. They also use less memory because they use a subset of training points in the decision phase.
  - SVM works well with a clear margin of separation and with high dimensional space.
- **Disadvantages**
  - SVM is not suitable for large datasets because of its high training time and it also takes more time in training compared to Naïve Bayes.
  - It works poorly with overlapping classes and is also sensitive to the type of kernel used.