Support Vector Machines

SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees.

It is known for its kernel trick to handle nonlinear input spaces.

It is used in a variety of applications such as

- face detection,
- intrusion detection,
- classification of emails,
- news articles and web pages,
- classification of genes, and
- handwriting recognition.

The classifier separates data points using a hyperplane with the largest amount of margin.

That's why an SVM classifier is also known as a discriminative classifier.

SVM finds an optimal hyperplane which helps in classifying new data points.

Support Vector Machines

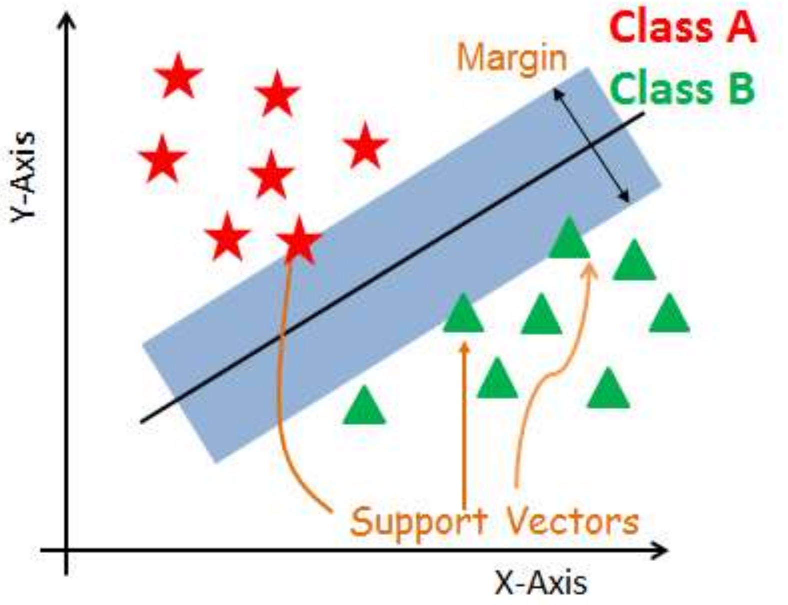
Generally, SVM is a classification approach, it but can be employed in both types of classification and regression problems.

It can easily handle multiple continuous and categorical variables.

SVM constructs a hyperplane in multidimensional space to separate different classes.

SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error.

The core idea of SVM is to find a maximum marginal hyperplane (MMH) that best divides the dataset into classes.



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.

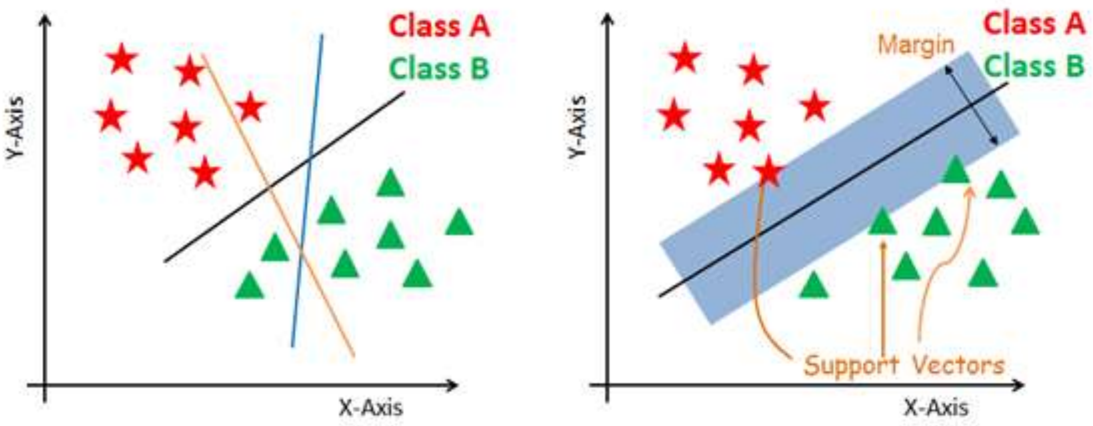
The main objective is to segregate the given dataset in the best possible way.
The distance between the either nearest points is known as the margin.

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.

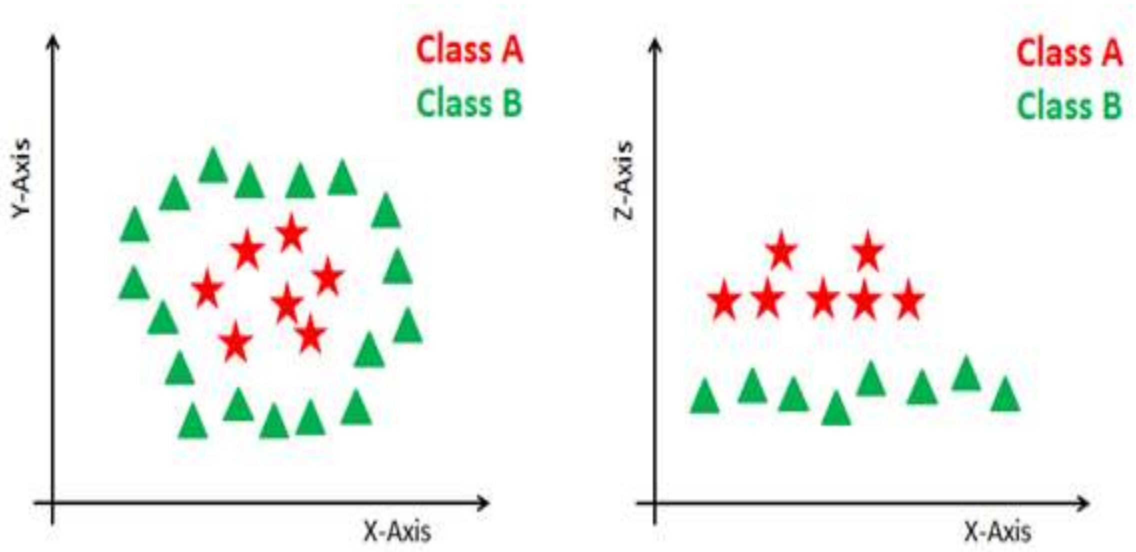


Dealing with non-linear and inseparable planes

Some problems can’t be solved using linear hyperplane, In such situation,

SVM uses a kernel trick to transform the input space to a higher dimensional space as shown on the right.

The data points are plotted on the x-axis and z-axis (Z is the squared sum of both x and y: z=x^2=y^2).



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 non separable 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.

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)

Polynomial Kernel

A polynomial kernel is a more generalized form of the linear kernel.

The polynomial kernel can distinguish curved or nonlinear input space.

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

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.

Radial Basis Function 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 a good default value. The value of gamma needs to be manually specified in the learning algorithm.

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.

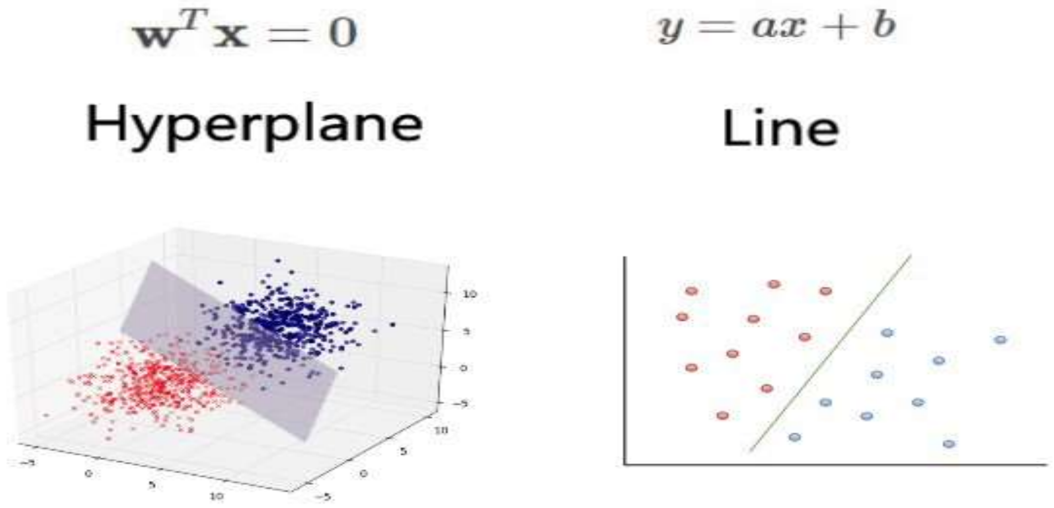
SVM works well with a clear margin of separation and with high dimensional space.

Disadvantages

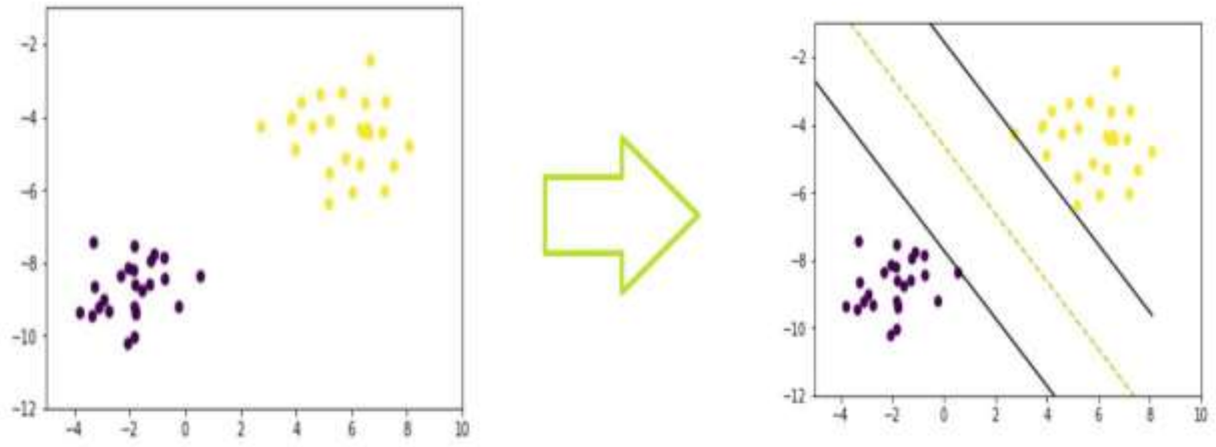
SVM is not suitable for large datasets because of its high training time and it also takes more time in training compared to Naïve Bayes.

It works poorly with overlapping classes and is also sensitive to the type of kernel used.

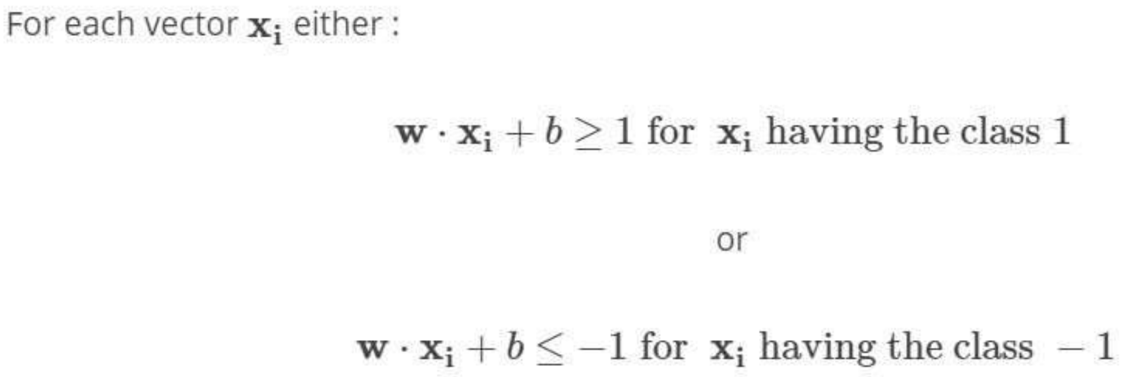
SVM hyperplane



The line equation and hyperplane equation — same, it’s a different way to express the same thing.



w.x+b=0 which is same as w.x =0 (which has more dimensions)



For each vector $\mathbf{x_i}$ either :

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 \text{ for } \mathbf{x_i} \text{ having the class } 1$$

or

$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 \text{ for } \mathbf{x_i} \text{ having the class } -1$$

if w.x+b=0 then we get the decision boundary

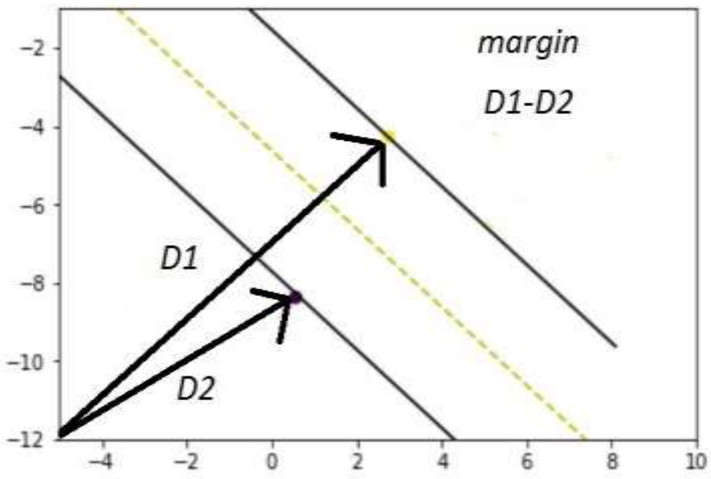
→The yellow dashed line

if w.x+b=1 then we get (+) class hyperplane

for all positive(x) points satisfy this rule (w.x+b ≥1)

if w.x+b=-1 then we get (-) class hyperplane

for all negative(x) points satisfy this rule (w.x+b≤-1)



$D1 = w^T x+b=1$ $w^T x+b-1=0$

$D2= w^T x+b=-1$ $w^T x+b+1=0$

$w^T x+b -1 \ - \ w^T x+b+1$

⇩ *Solve algebraically*

$$\frac{2}{|w|}$$

so to increse the margin, minimize the |w| $\frac{1}{2} |w|^2$