Introduction to Support Vector Machines

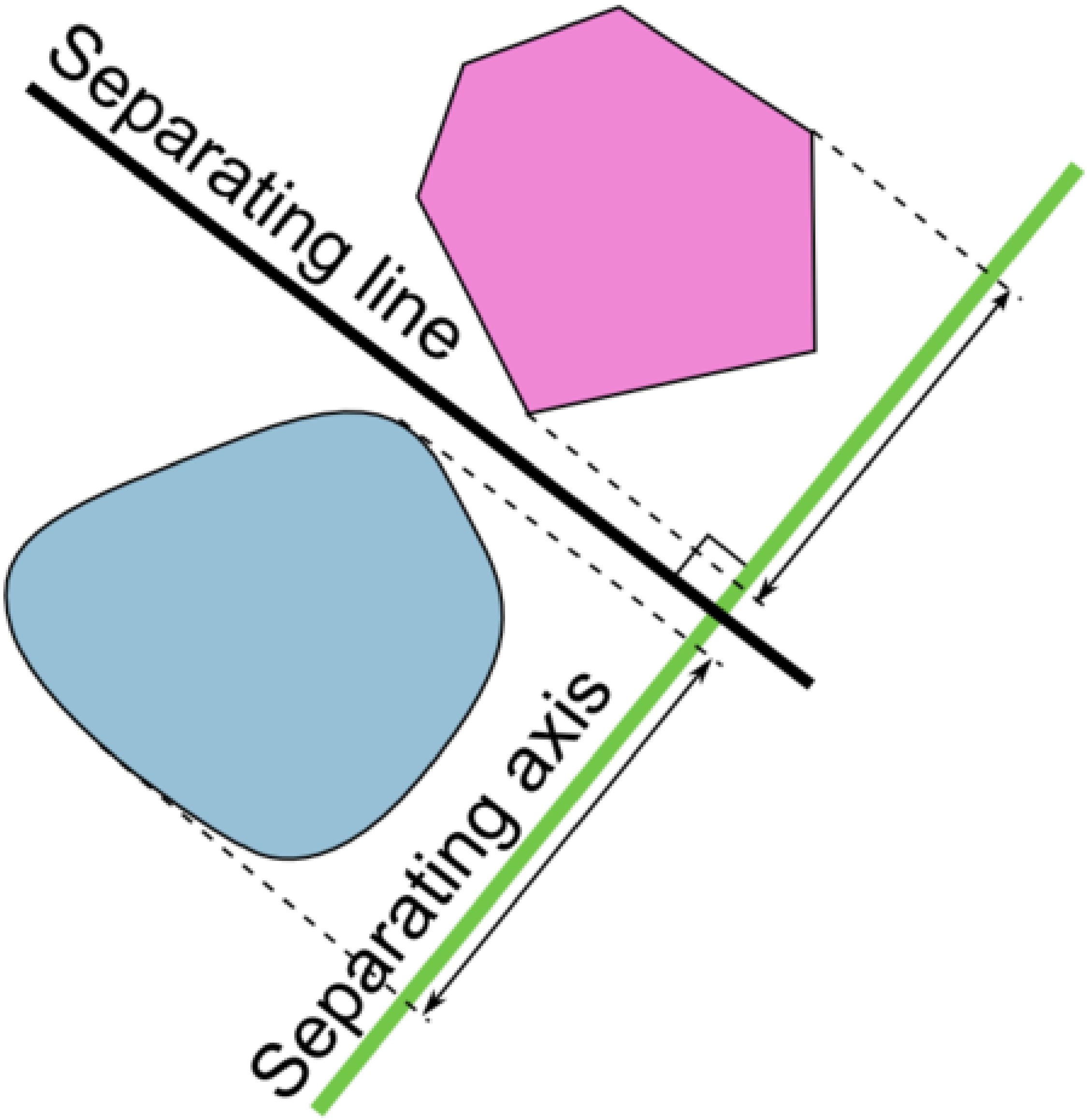
It is supervised machine learning algorithm which can be used for both classification and regression(SVR) i.e. **Support Vector Machine or simply SVM.**

Introduction

Support Vector Machines(SVM) are among one of the most popular machine learning algorithms.
They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high-performing algorithm with a little tuning.

The objective of **SVM is to find a hyperplane in an N-dimensional space (N-Number of features) that distinctly classifies the data points.**
Support Vector Machine is a generalization of maximal margin classifier. This classifier is simple, but it cannot be applied to the majority of the datasets since the classes must be separated by a boundary which is linear.

In the context of support-vector machines, the *optimally separating hyperplane* or *maximum-margin hyperplane* is a hyperplane which separates two convex hulls of points and is equidistant from the two.

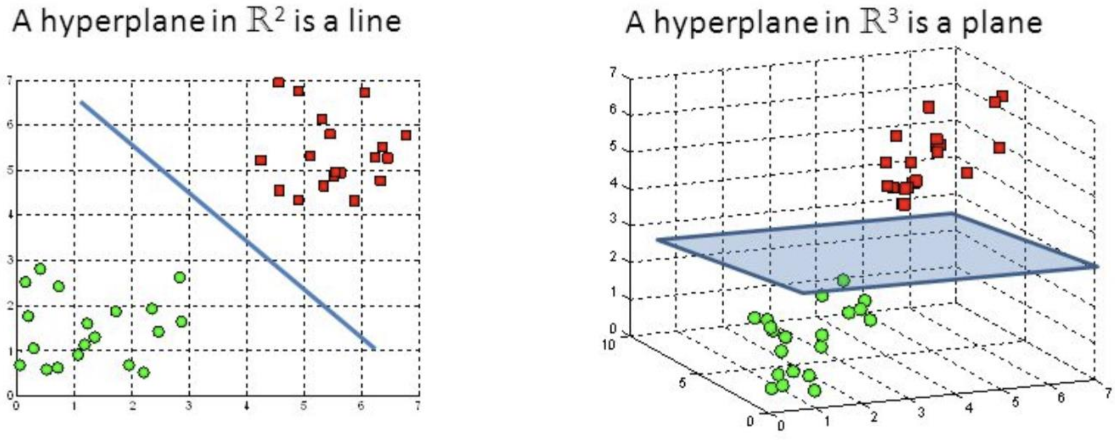


Maximum Margin Classifier

Ok. **What is a hyperplane?**

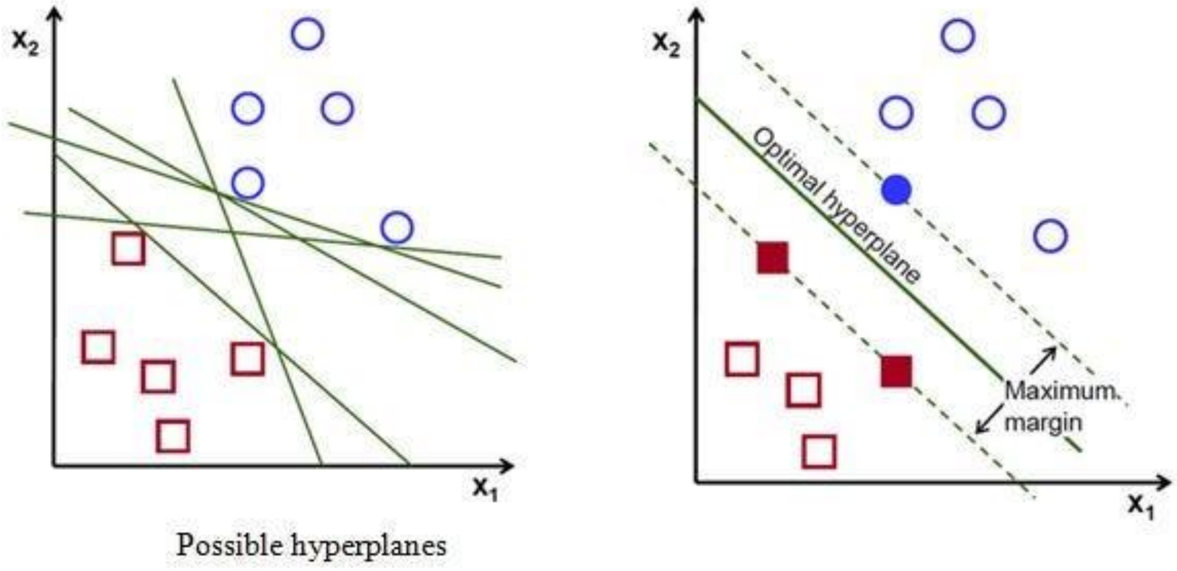
In an N-dimensional space, a hyperplane is a flat affine subspace of dimension N-1. Visually, in a 2D space, a hyperplane will be a line and in 3D space, it will be a flat plane.

In simple terms, hyperplane is a decision boundary that helps classifying data points.



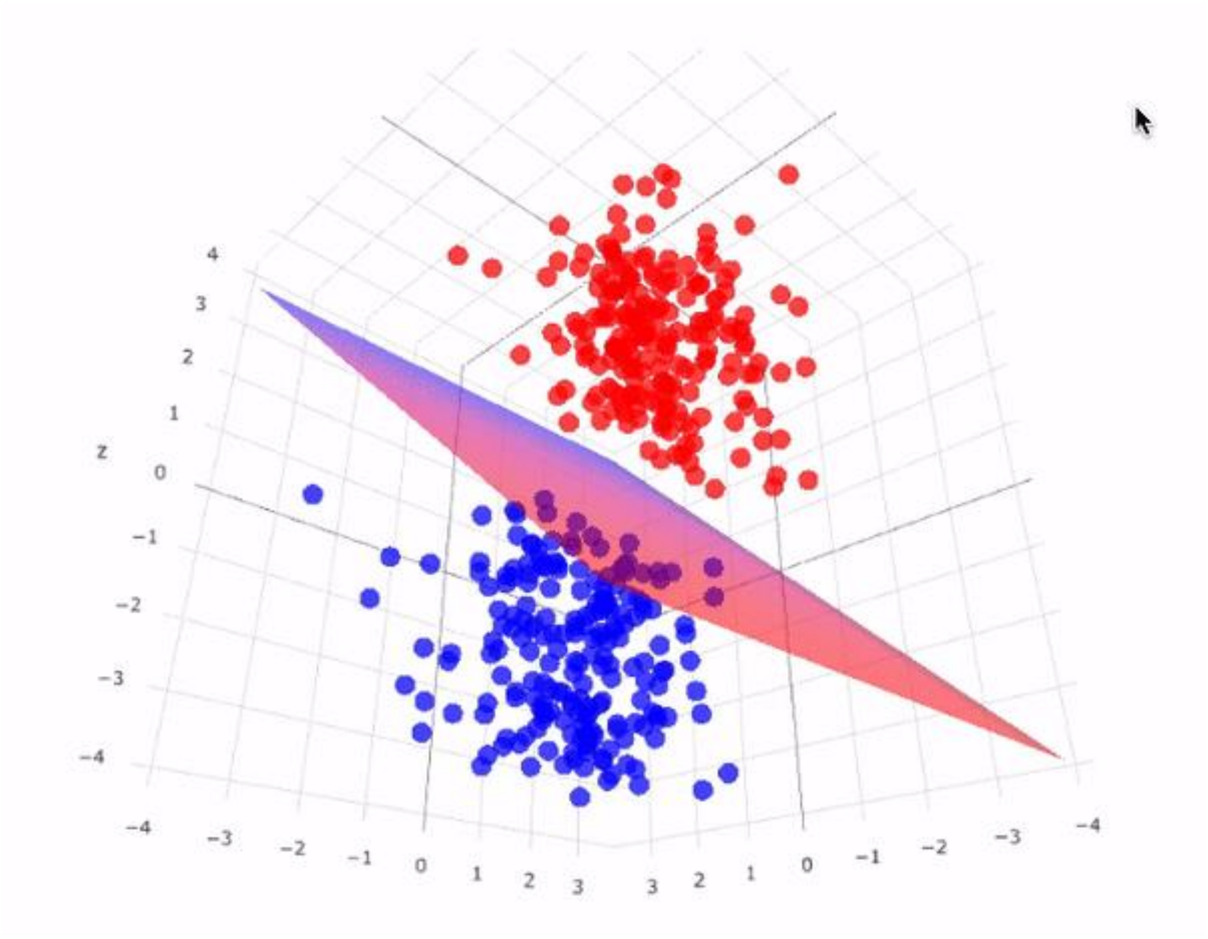
Hyperplane in 2D and 3D space

Now, to separate two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin i.e. the maximum distance between data points of both classes and below figure clearly explains this fact.



Possible Hyperplanes and Hyperplane with maximum margin

Hyperplane with maximum margin looks something like this in 3D space:

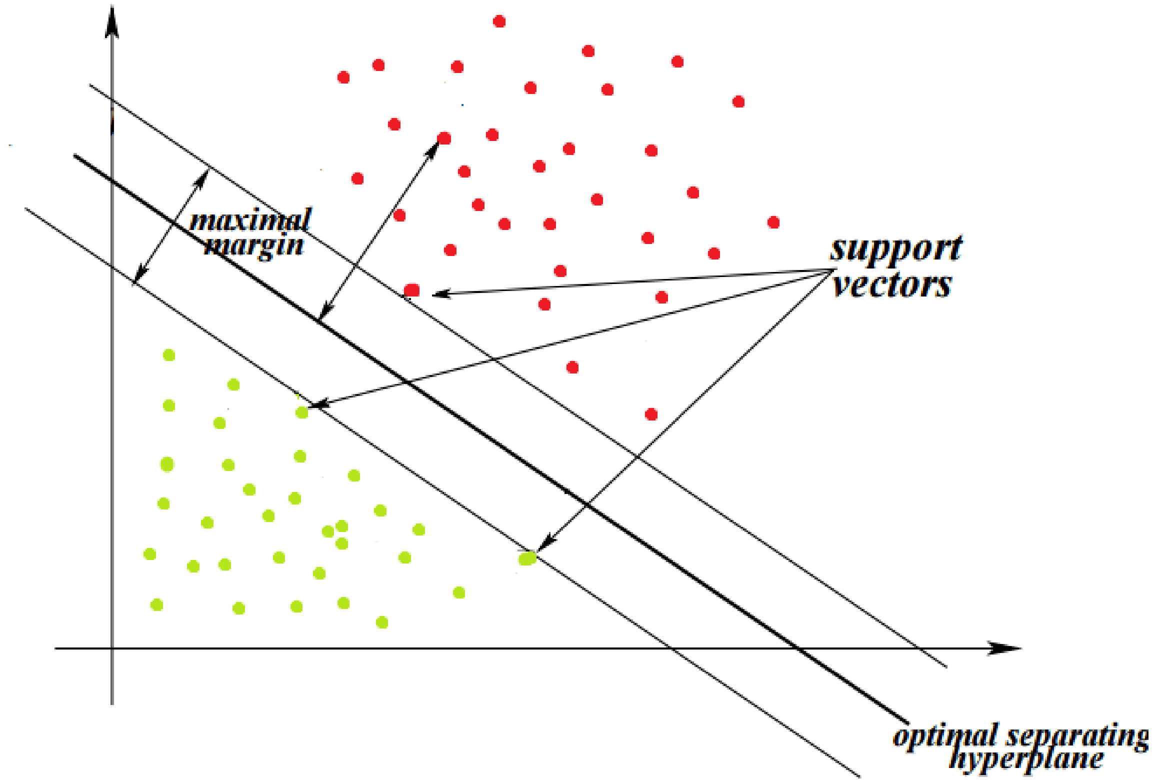


Visual representation of Hyperplane in 3D

Note:- The dimension of Hyperplane depends on the number of features.

Support Vectors

Support Vectors are the data points that are on or closest to the hyperplane and influence the position and orientation of the hyperplane. Using these support Vectors we maximize the margin of the classifier and deleting these support vectors will change the position of the hyperplane. These are actually the points that help us build SVM.

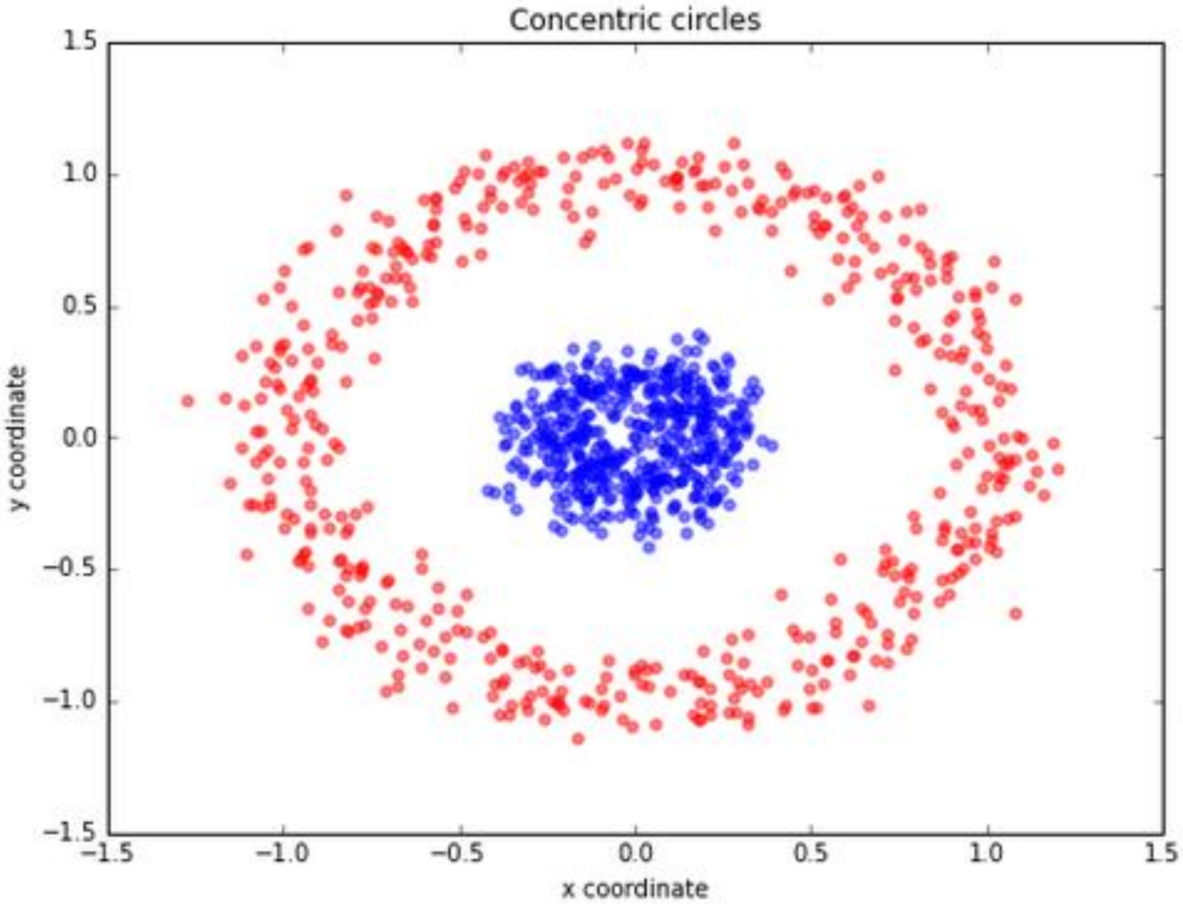


Support Vectors

Support Vectors are equidistant from the hyperplane. They are called support vectors because if their position shifts, the hyperplane shifts as well. This means that the **hyperplane depends only on the support vectors** and not on any other observations.
SVM that we have discussed until now can only classify the data which is linearly separable.

What if the data is non-linearly separated?

For example: look at the below image where the data is non-linearly separated, of course, we cannot draw a straight line to classify the data points.



Non-linearly separated data

Here comes the concept of Kernel in SVM to classify non-linearly separated data. **A kernel is a function which maps a lower-dimensional data into higher dimensional data.**

There are two ways by which kernel SVM will classify non-linear data.

1. Soft margin
2. Kernel tricks

Soft Margin

It allows SVM to make a certain number of mistakes and keep the margin as wide as possible so that other points can still be classified correctly.

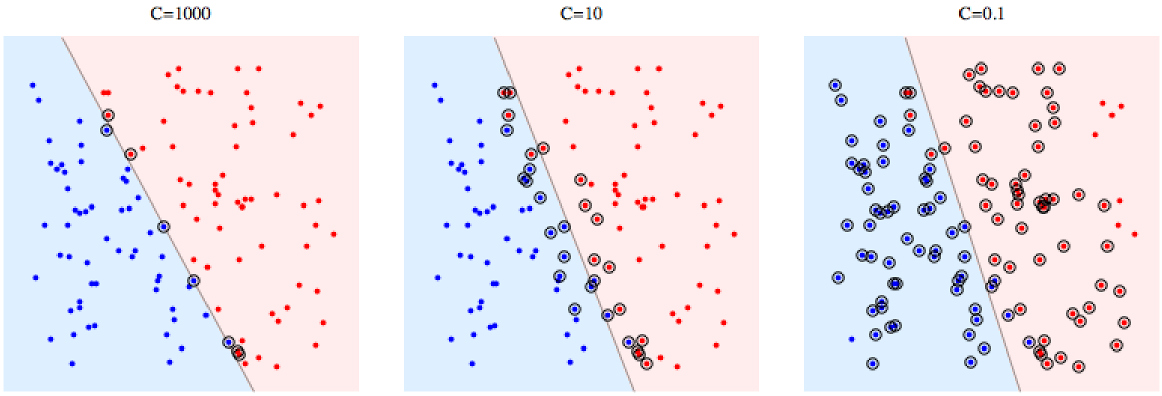
“In other words, SVM tolerates a few dots to get misclassified and tries to balance the tradeoff between finding the line that maximizes the margin and minimizes misclassification.”

There are two types of misclassifications can happen:

1. The data point is on the wrong side of the decision boundary but on the correct side
2. The data point is on the wrong side of the decision boundary and on the wrong side of the margin

Degree of tolerance

How much tolerance we want to set when finding the decision boundary is an important hyper-parameter for the SVM (both linear and nonlinear solutions). In Sklearn, it is represented as the penalty term — ‘C’.



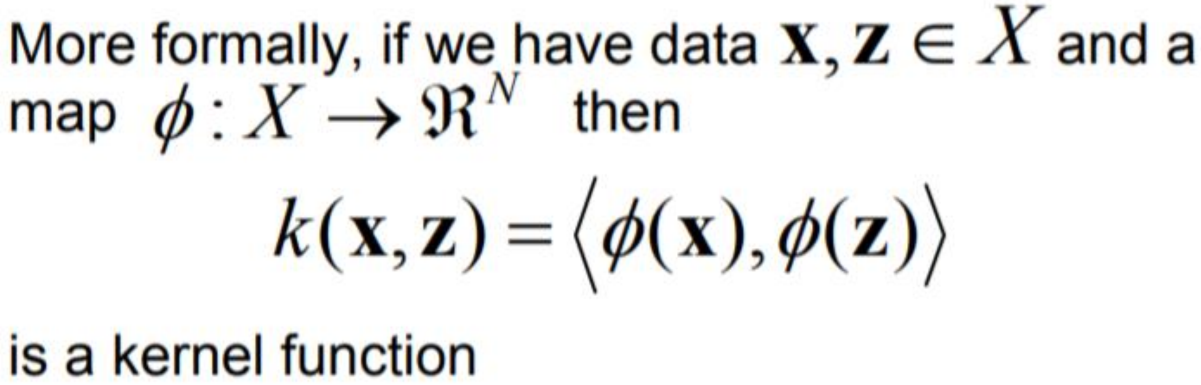
The bigger the C, the more penalty SVM gets when it makes misclassification. Therefore, the narrower the margin is and fewer support vectors the decision boundary will depend on.

Kernel Trick

The idea is mapping the non-linear separable data from a lower dimension into a higher dimensional space where we can find a hyperplane that can separate the data points.

So it is all about finding the mapping function that transforms the 2D input space into a 3D output space and to reduce the complexity of finding the mapping function SVM uses Kernel Functions.

Kernel Functions are generalized functions that take 2 vectors(of any dimension) as input and output a score(dot product) that denotes how similar the input vectors are. If the dot product is small, vectors are different and if the dot product is large, vectors are more similar.



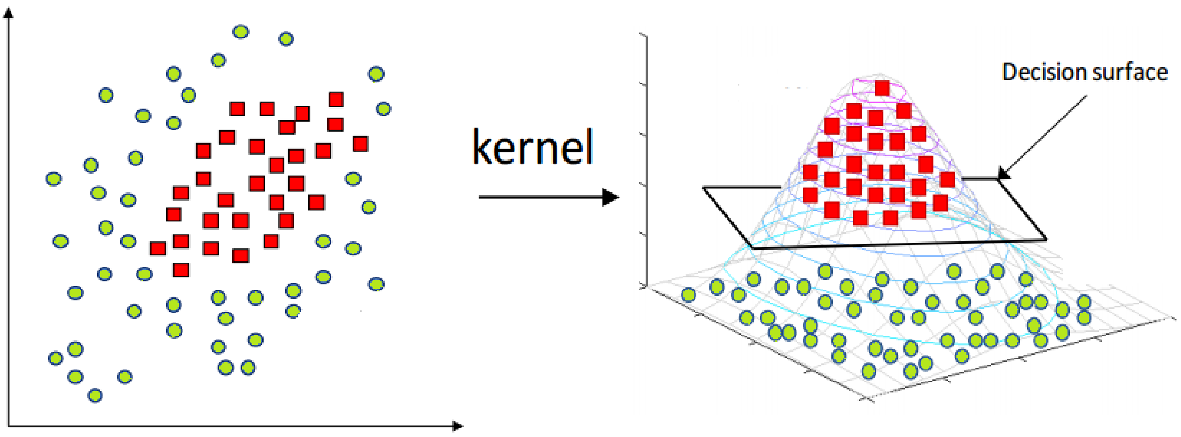
More formally, if we have data $\mathbf{x}, \mathbf{z} \in X$ and a map $\phi : X \rightarrow \Re^N$ then

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$$

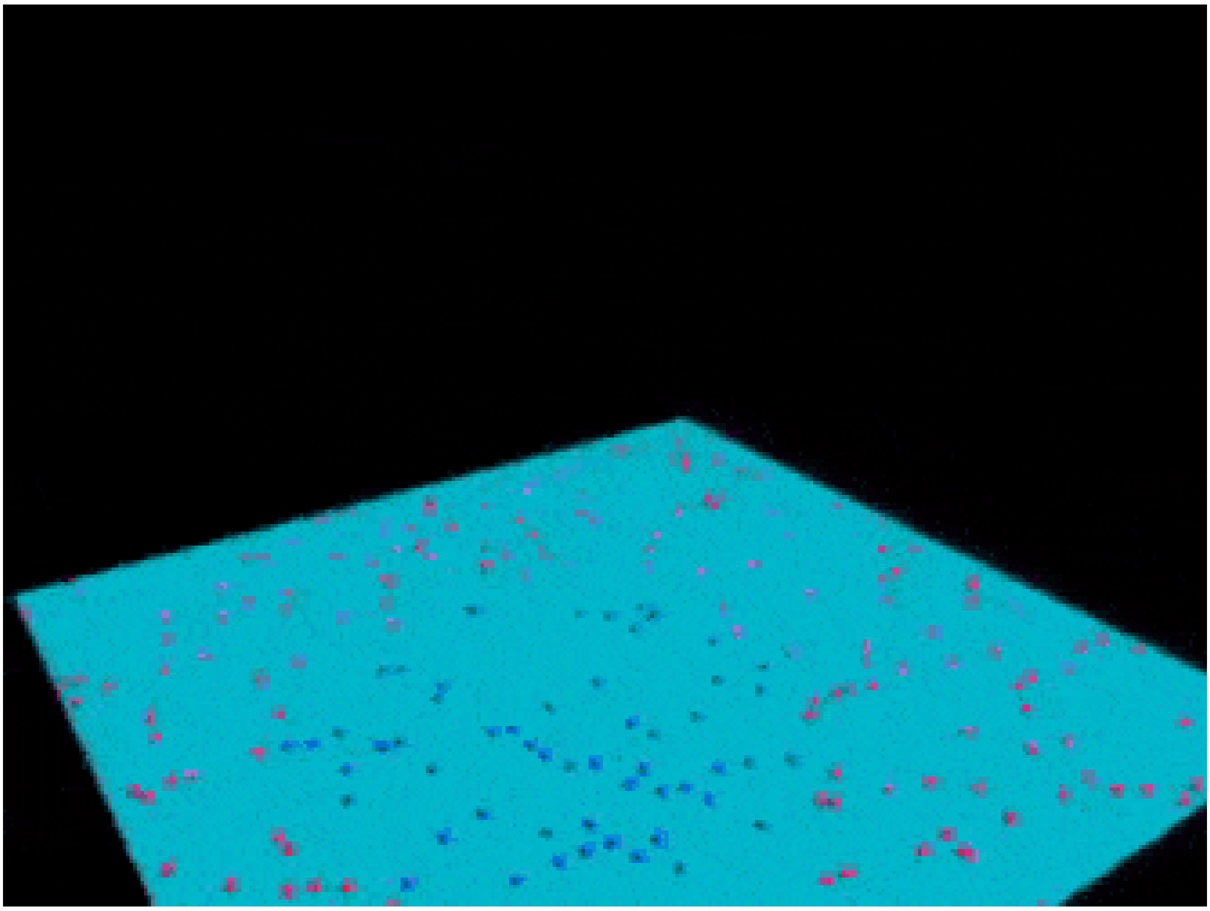
is a kernel function

Mathematical representation of Kernel Function

Pictorial representation of Kernel Trick :



Visual representation of Kernel Trick :



Kernel Trick in action, by udiprod in Youtube

Types of Kernel Functions:

1. Linear
2. Polynomial
3. Radial Basis Function(rbf)
4. Sigmoid

Let's talk about the most used kernel function i.e. **Radial Basis Function(rbf)**.

Think of rbf as a transformer/processor to generate new features of higher dimension by measuring the distance between all other data points to a specific dot.

The most popular rbf kernel is Gaussian Radial Basis function. Mathematically:

$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2)$$

where gamma($\boldsymbol{\gamma}$) controls the influence of new features on the decision boundary. Higher the value of $\boldsymbol{\gamma}$, more influence of features on the decision boundary.

Similar to Regularization parameter/penalty term(C) in the soft margin, Gamma($\boldsymbol{\gamma}$) is a hyperparameter that can be tuned when we use kernel trick.

Popular SVM Kernel Functions

Linear Kernel

It is the most basic type of kernel, usually one dimensional in nature. It proves to be the best function when there are lots of features. The linear kernel is mostly preferred for **text-classification problems** as most of these kinds of classification problems can be linearly separated.

Linear kernel functions are **faster** than other functions.

Linear Kernel Formula

$$F(x, xj) = sum(x.xj)$$

Here, **x, xj** represents the data you’re trying to classify.

Polynomial Kernel

It is a more generalized representation of the linear kernel. It **is not** as preferred as other kernel functions as it is **less efficient** and accurate.

Polynomial Kernel Formula

$$F(x, xj) = (x.xj+1)^d$$

Here ‘.’ shows the **dot product** of both the values, and **d** denotes the degree.

F(x, xj) representing the **decision boundary** to separate the given classes.

Gaussian Radial Basis Function (RBF)

It is one of the most preferred and used kernel functions in svm. It is usually chosen for non-linear data. It helps to make proper separation when there is no prior knowledge of data.

Gaussian Radial Basis Formula

$$F(x, xj) = exp(-gamma * ||x - xj||^2)$$

The value of gamma varies from **0 to 1**. You have to manually provide the value of gamma in the code. The most preferred value for **gamma is 0.1**.

Sigmoid Kernel

It is mostly preferred for **neural networks**. This kernel function is similar to a two-layer perceptron model of the neural network, which works as an **activation function** for neurons.

Sigmoid Kenel Function

$$F(x, xj) = \tanh(\alpha xay + c)$$