Machine Learning Kernel Functions – 3D Visualization

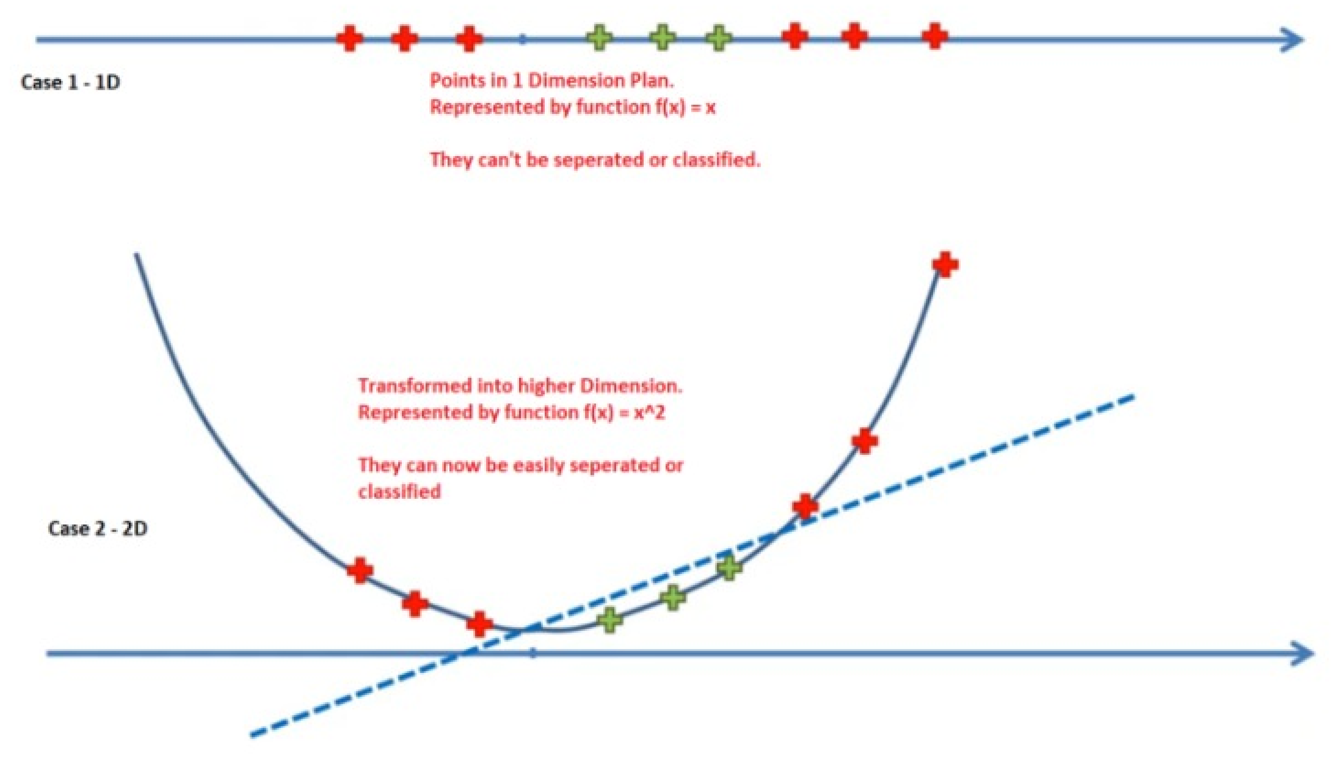
Due to the increased popularity of the Support Vector Machines, Kernel methods have received major attention in recent years.

Kernel functions are often utilized in many applications as they dispense a simple bridge from linearity to non-linearity for algorithms that can be expressed in terms of dot products. In this article, we will list some of the kernel functions with their properties.

What are Kernel Methods?

Kernel methods is a category of algorithms for pattern analysis or recognition, whose eminent element is the support vector machine (SVM). The accustomed task of pattern analysis is to find & study generic types of relations such as clusters, rankings, principal components, correlations, classifications in generic types of data such as sequences, text docs, cluster of points, vectors, images, graphs, etc.

The major characteristic of Kernel Methods, however, is their definite approach.
Kernel methods map the data into higher dimensional spaces with an aim that – in this higher-dimensional space the data could become more easily separated or better structured. This is pictured as follows:

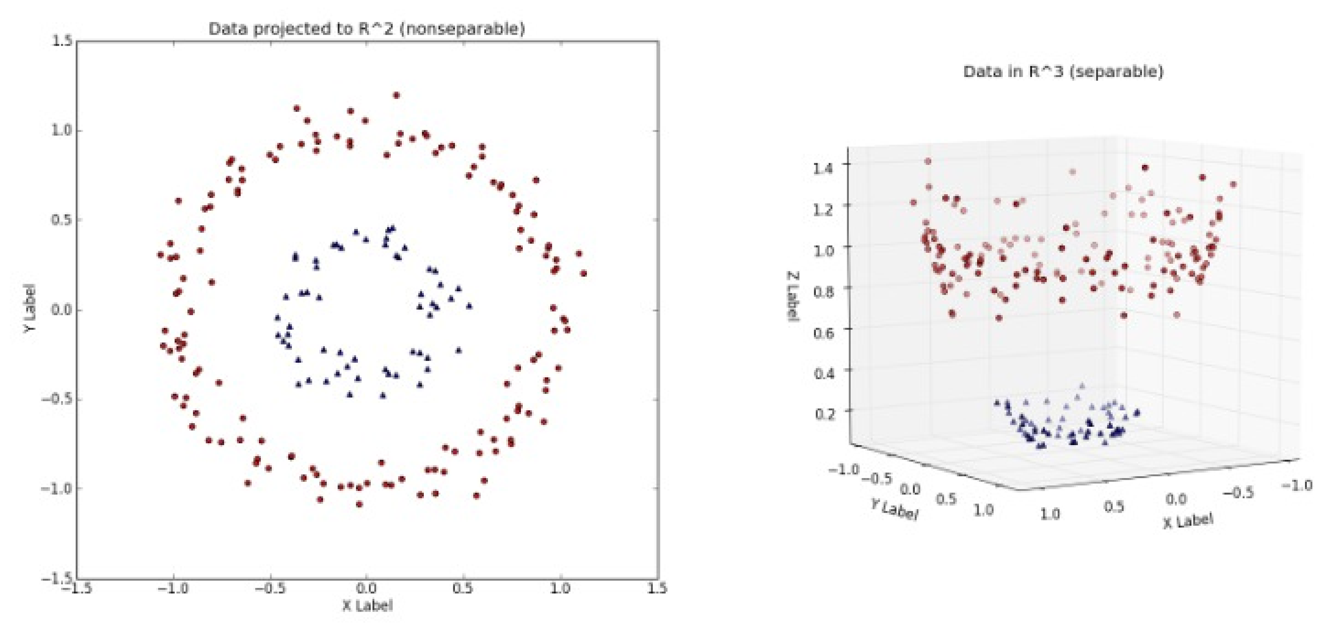


There aren't any constraints on the form of this mapping, which might lead to infinite-dimensional spaces. This mapping function, however, hardly be obliged to be computed because of a tool called the ***kernel trick***.

What is the Kernel Trick?

The Kernel trick is a very fascinating & powerful tool.
Its powerful because it bestow a bridge from linearity to non-linearity to any algorithm that can expressed entirely on terms of dot products between two vectors. It arrives from the fact that, if we initially map our input data into a higher-dimensional space, a linear algorithm operating in this space will act non-linearly in the original input space. And, we don't exactly require the exact data points, but only their inner products to reckon our decision boundary.

What it implies is that if we want to transform our existing data into a higher dimensional data, which in many cases help us classify better, we need not compute the explicit transformation of our data, we just require the inner product of our data in that higher dimensional space.



It's a lot easier to get the inner product in a higher dimensional space than the actual points in a higher dimensional space.

Kernel Functions

Below is a list of some kernel functions:

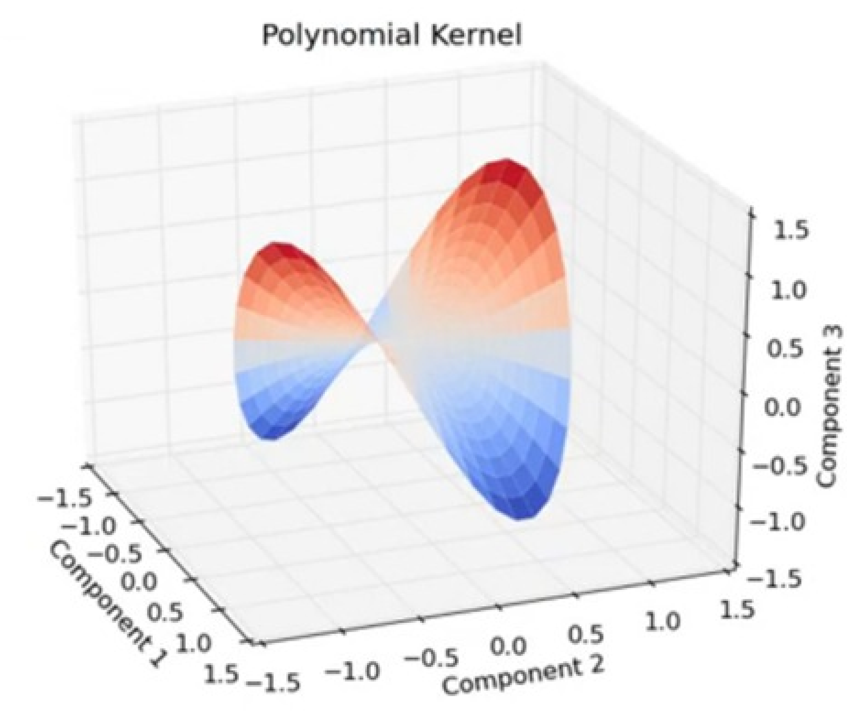
Polynomial Kernel

The Polynomial kernel is a non-stationary kernel.

$$k(x, y) = (\alpha x^T y + c)^d$$

Adjustable parameters are :

- constant term **c**
- slope **alpha**
- polynomial degree **d**.



Gaussian Kernel

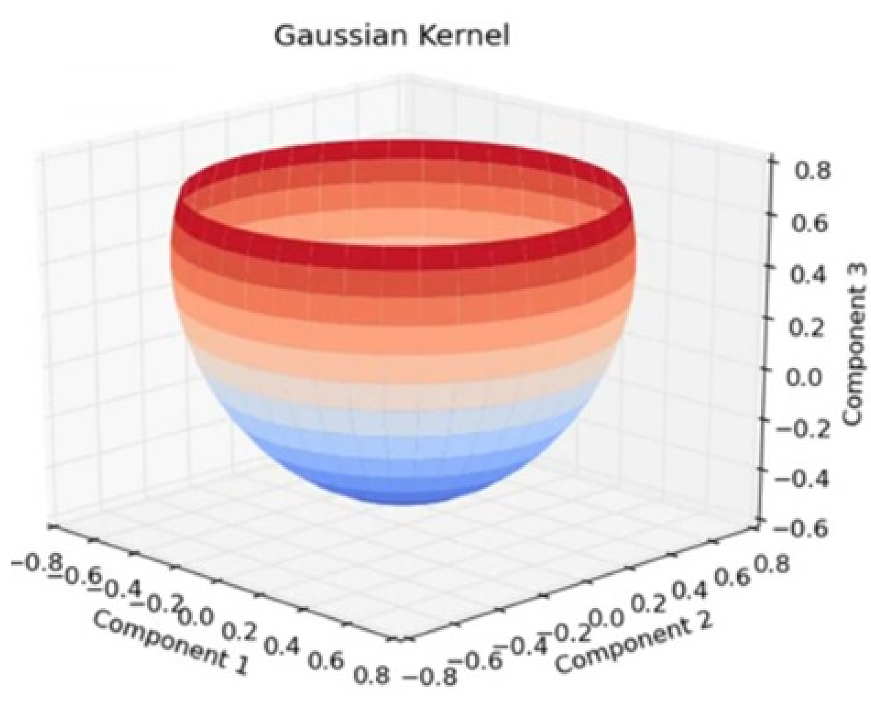
The Gaussian Kernel is an illustration of radial basis function Kernel.

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right)$$

Alternatively, it could also be implemented using

Invalid Equation

The adjustable parameter **sigma** plays an important role in the performance of the kernel, and it should be efficiently tuned to the problem in hand. If overestimated, the exponential will act almost linearly and the higher-dimensional projection will begin to lose its non-linear power. On other hand, if underestimated, the function will have a deficit regularization & the decision boundary will be highly sensitive to noise in training data.

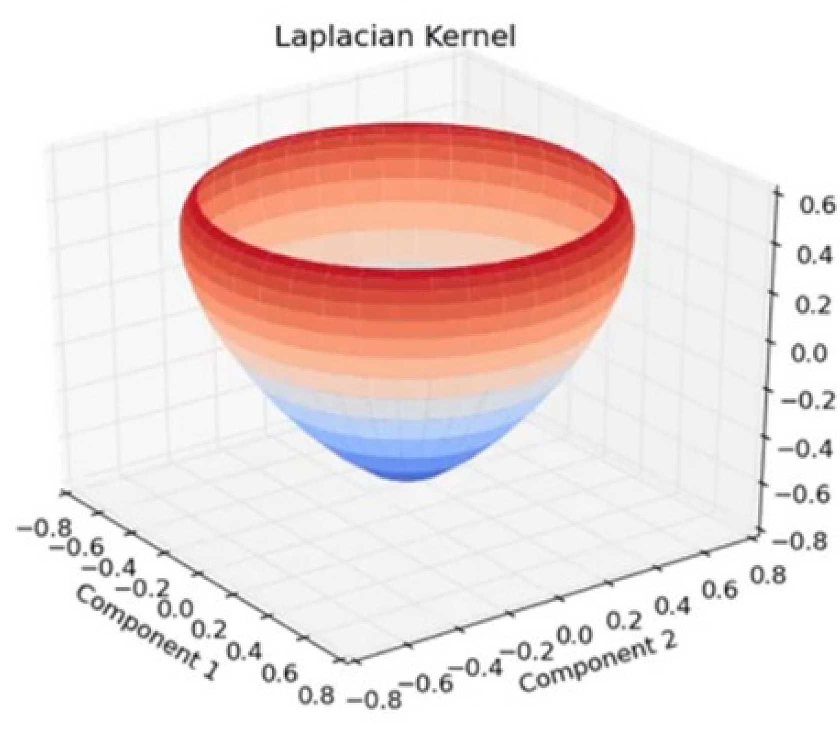


Laplacian Kernel

The Laplace Kernel is completely similar to the exponential kernel, except for being less sensitive for changes in the sigma parameter. Being equivalent, it's also a radial basis function kernel.

$$k(x, y) = \exp\left(-\frac{||x - y||}{\sigma}\right)$$

Its important to list that the observations made about the sigma parameter for the Gaussian kernel also apply to the Exponential and Laplacian



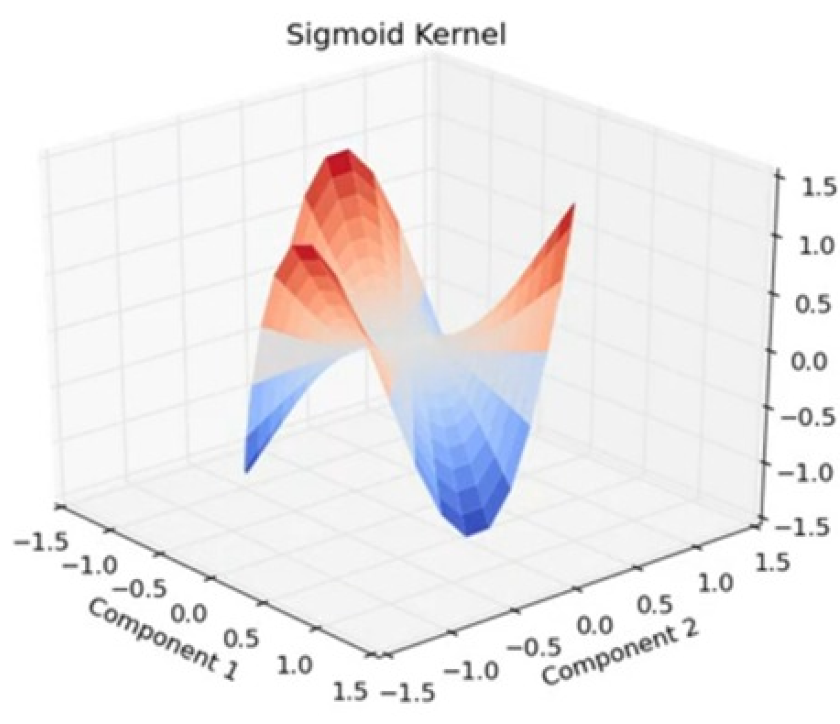
Sigmoid Kernel

The Sigmoid Kernel originates from the Neural Networks field, where the bipolar sigmoid function is often utilized as an activation function for artificial neurons.

$$k(x, y) = \tanh(\alpha x^T y + c)$$

It's interesting to note that a SVM model utilizing a sigmoid kernel function is equivalent to a two-layer, perceptron neural network. This kernel was quite favored for support vector machines because of its origin from neural network theory.

There are two adjustable parameters in the sigmoid kernel, the intercept constant **c** & the slope **alpha**. A frequent value for alpha is 1/N, where N is the data dimension.



Chi-Squared Kernel

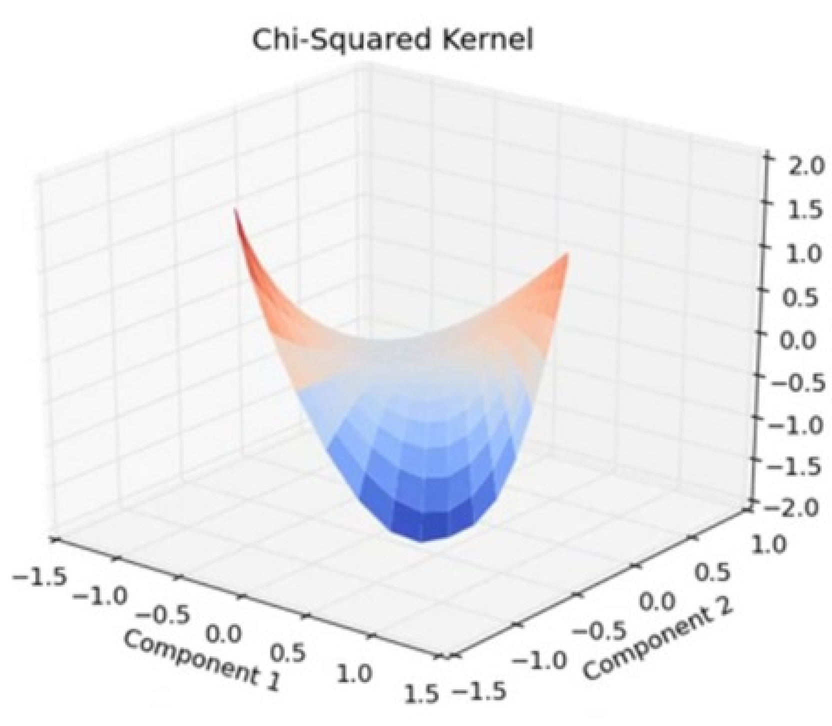
The Chi-Square kernel originates from the Chi-Square distribution:

$$k(x, y) = 1 - \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}$$

However, this version of the kernel is only conditionally positive-definite (CPD). A positive-definite version of this kernel is given as:

$$k(x, y) = \sum_{i=1}^{n} \frac{2x_i y_i}{(x_i + y_i)}$$

and is advisable to be used by methods other than support vector machines.



Periodic Kernel

The periodic kernel permits one to model functions that repeat themselves exactly. Its parameters are easily interpretable:

- The period pp commonly determines the distance between repetitions of the function.
- The length scale ℓℓ determines the length scale function in the similar fashion as in the SE kernel.

$$k_{\mathrm{Per}}(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{\ell^2}\right)$$