Robust Support Vector Machines for Classification

I. Vranckx, J. Schreurs, B. De Ketelaere, M. Hubert and J.Suykens

February 22, 2019

Overview

In this talk I will talk about joint work with: ...

Kernel feature space

As a warm-up, we'll introduce the kernel feature space by an example.

Linear Regression

Define a *linear* regression problem, given a dataset $\{x_i, y_i\}_{i=1}^{n} \in \mathbb{R}$ as follows:

$$y_i = w^T x_i + \epsilon_i \tag{1}$$

Here ϵ_i is assumed to be white noise. Least squares is used to obtain an estimate $\hat{w}$ of w.

If the system from which the data are collected is linear, the regression provides a good approximation of the system behaviour.

However, if the true system follows the non-linear process:

$$y = w_1 x^2 + w_2 \sqrt{2} x_i + w_3 + \epsilon_i \tag{2}$$

The regression is not correctly specified because it does not contain the non-linear effect x^2.

To obtain the correct specification, the original input x can be mapped to a higher dimensional space by means of the feature map $\varphi : \mathbb{R} \to \mathbb{R}^3$, in this example defined by:

$$\varphi(x) = [x^2, \sqrt{2}x, 1] \tag{3}$$

Solving the regression

$$y_i = w^T \varphi(x_i) + \epsilon_i \tag{4}$$

...yields an estimate of $w = [w_1, w_2, w_3]$. In this example, the feature map $\varphi(x)$ is assumed to be known, and thus the coordinates in the high-dimensional space can be computed directly to arrive at the correct regression specification.

However, note that a feature map $\varphi(x)$ does not have to be known explicitly: non-linear transformations are rather implicitly defined by kernel functions.

For example, the inner product of two vectors is defined as:

$$\varphi(x_1)^T \varphi(x_2) = [x_1^2, \sqrt{2}x_1, 1]^T . [x_2^2, \sqrt{2}x_2, 1] \tag{5}$$

$$\tag{6}$$

Which is equivalent to following kernel function:

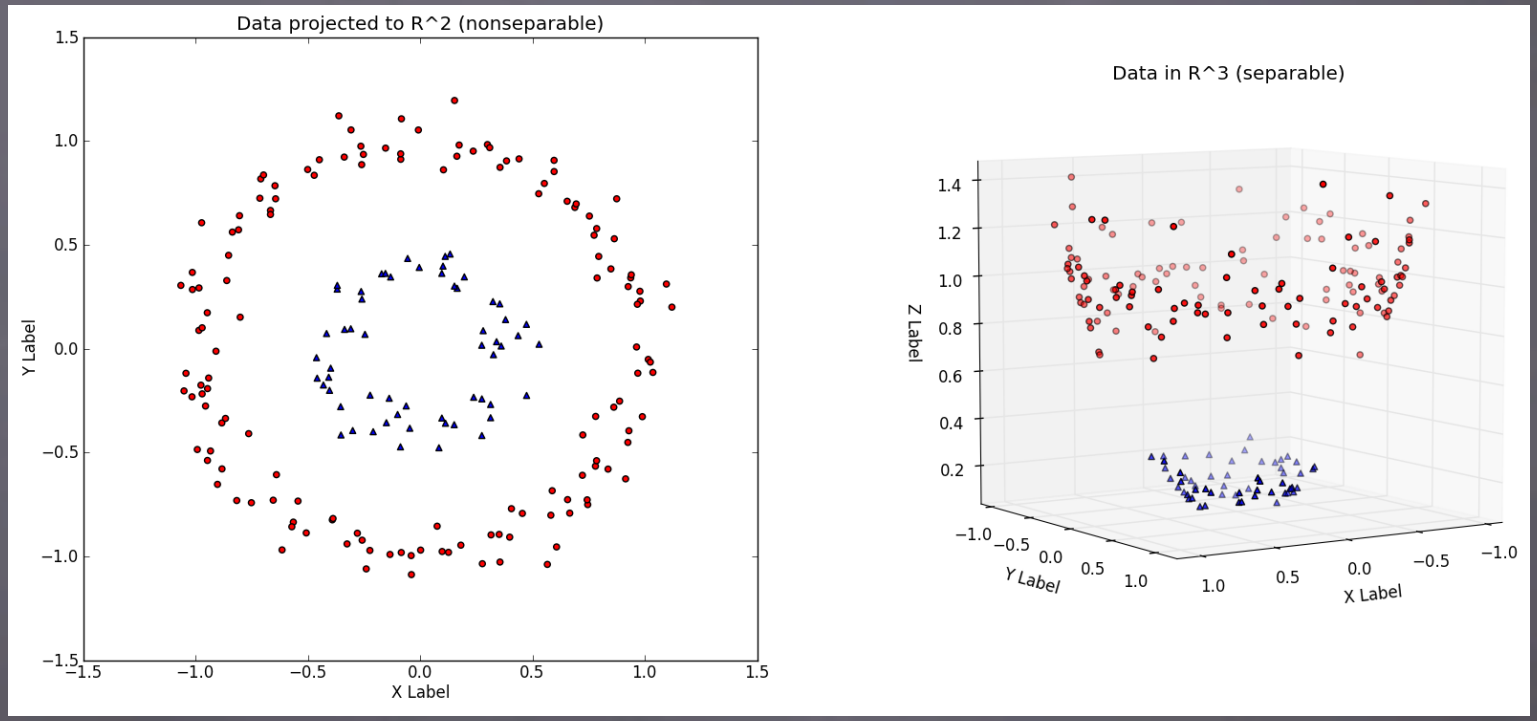
$$K(x_1, x_2) = (x_1^T x_2 + 1)^d \qquad d \in \mathbb{I} \tag{7}$$

Assuming x_1 and x_2 are two input samples in the original $\mathbb{R}^p$ space and they are mapped into the feature space as $\varphi(x_1)$ and $\varphi(x_2)$, the inner product in the feature space has an equivalent kernel in the original input space, i.e.

$$\varphi(x_1)^T \varphi(x_2) = K(x_1, x_2) \tag{8}$$

This is called the kernel trick, where $K(.,.)$ is a positive definite function that satisfies Mercer's conditions. Plugging this equation into the regression formula 1 yields its 'kernelized' form, in this example effectively acting in $\mathbb{R}^3$ space.

The following principle illustrates this feature transformation principle graphically.



Take away message

In a nutshell, kernel transformations allows us to construct non-linear, high potential classifiers. At the same time, the accurate working of the used optimization routine (and all other algorithms involved) becomes increasingly important.

Robust Support Vector Machines