UniGe | MaLGa

Lab 4 – Kernel RLS

Regularized Least Squares

$$w_\lambda = \arg \min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda w^T w$$

$$\hat{f}_\lambda(x) = w^T x$$

$$\underbrace{\boxed{(X_n^\top X_n + \lambda n I)}}_{A} w = \underbrace{\boxed{X_n^\top Y_n}}_{b}$$

$$Aw = b$$

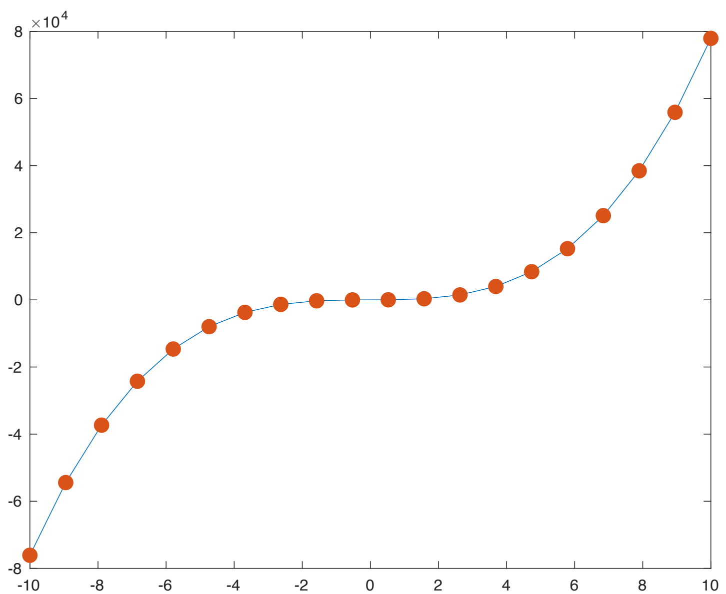
What if the function f is not linear?

What if we have non linear functions?

Say for instance that the true function is

$$f : X \to Y$$
$$X \subseteq \mathbb{R}$$
$$Y \subseteq \mathbb{R}$$

$$f(x) = 77x^3 + 9x^2 + x + 5$$



Dealing with non linear functions

You can see non linear functions as linear with a trick.

$$f(x) = 77x^3 + 9x^2 + x + 5$$

Define an intermediate variable

$$z(x) = [x^3 \; x^2 \; x \; 1]$$ Feature map!

Such that you can write

$$f(x) = w^T z(x)$$

where

$$w = [77 \; 9 \; 1 \; 5]$$

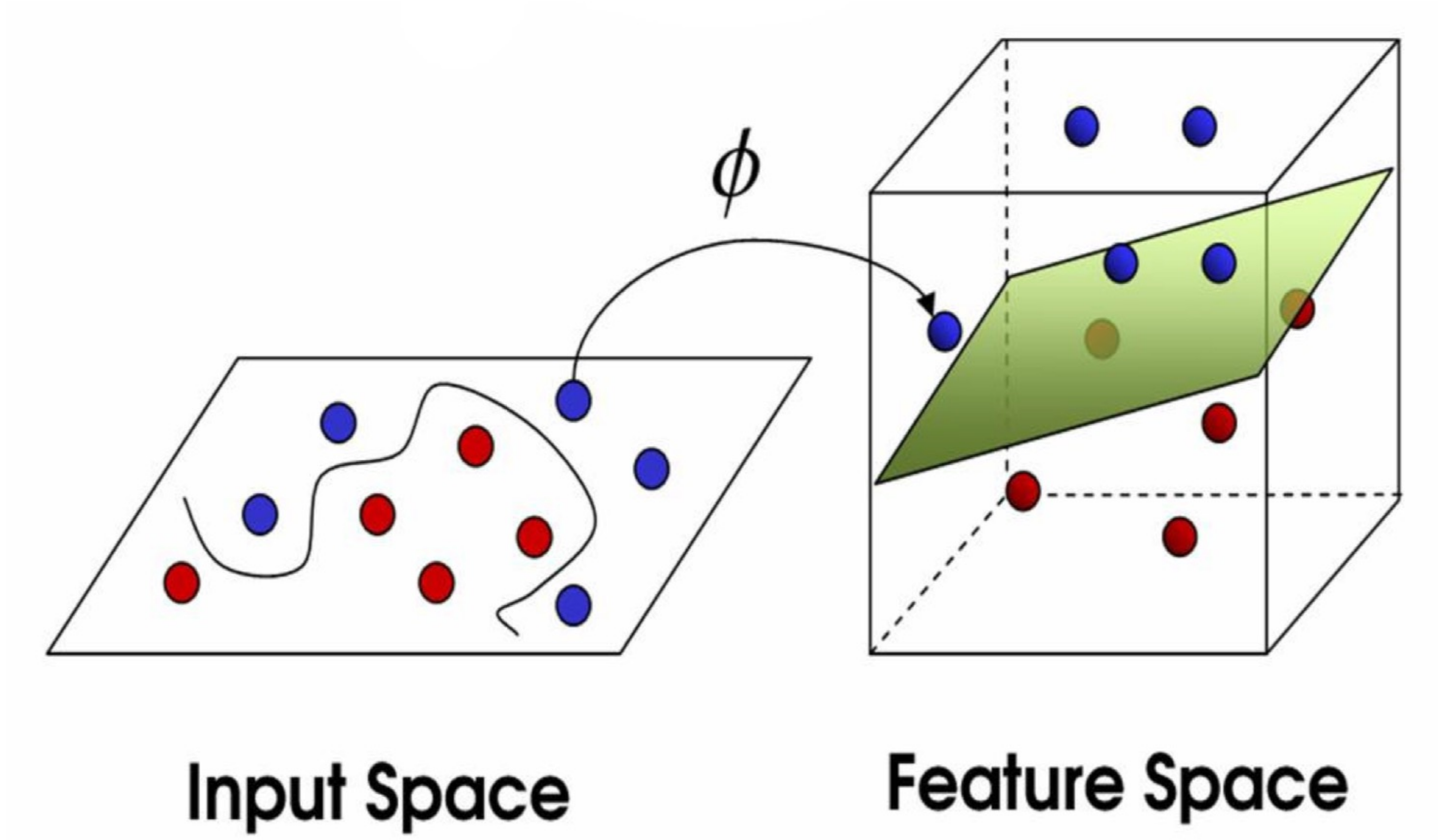
Feature maps

- We resort to the use of feature maps

$$\Phi : \mathbb{R}^D \to \mathbb{R}^P$$

$$x \to \Phi(x)$$

The intuition



Feature space and linear RLS

$$w_\lambda = \arg \min_{w \in \mathbb{R}^P} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T \Phi(x_i))^2 + \lambda w^T w$$

$$f(x) = \Phi(x)^T w$$

Representer Theorem & Kernel functions

$$w_\lambda = \arg\min_{w \in \mathbb{R}^P} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T \Phi(x_i))^2 + \lambda w^T w$$

$$f(x) = \Phi(x)^T w$$

$$\Downarrow$$

$$c_\lambda = \arg\min_{c \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n} K(x_i, x_j) c_j)^2 + \lambda \sum_{i,j=1}^{n} K(x_i, x_j) c_i c_j$$

$$\hat{f}_\lambda(x) = \sum_{i=1}^{n} K(x_i, x) c_i$$

Representer Theorem & Kernel functions

$$\hat{K}_{ij} = K(x_i, x_j)$$ `kernel_matrix`

$$\hat{c}_\lambda = (\hat{K} + \lambda n I)^{-1} \hat{Y}$$ `krls_train`

$$\hat{f}_\lambda(x) = \sum_{i=1}^{n} K(x_i, x) c_i$$ `krls_predict`

Different Kernels

- Linear kernel

$$K(x, x') = x^T x'$$

- Polynomial kernel

$$K(x, x') = (x^T x' + 1)^d$$

- Gaussian kernel

$$K(x, x') = e^{-\frac{||x - x'||^2}{2\sigma^2}}$$

Your objectives today

- Generating data for a non-linear problem
- Try with RLS with linear functions
- Try with feature maps + RLS with linear functions
    - On this we suggest a simple map, related with the type of non-linearity in the data… for more general solutions you may adopt the Python function ***sklearn.preprocessing.PolynomialFeatures*** (link ***https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html)***
- Try with kernel RSL

UniGe

MaLGa