KERNELS

Nonlinear Feature Maps

One way to create a nonlinear method for regression or classification is to first transform the feature vectors via a nonlinear map

$$\Phi : \mathbb{R}^d \longrightarrow \mathbb{R}^m$$

and apply a linear method to the transformed data $\Phi(x_1), \ldots, \Phi(x_n)$.

In regression, the nonlinear regression function is

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

where $w \in \mathbb{R}^m$, $b \in \mathbb{R}$.

Example Determine the least squares cubic polynomial to $(x_1, y_1), \ldots, (x_n, y_n)$ where $x_i \in \mathbb{R}$, $y_i \in \mathbb{R}$.

Let's write $f(x) = a + bx + cx^2 + dx^3$

$$= w^T \underline{\Phi}(x) + a$$

where

$$w = \begin{bmatrix} b \\ c \\ d \end{bmatrix}, \quad \underline{\Phi}(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

nonlinear

Since

$$\sum_i (y_i - f(x_i))^2 = \sum_i (y_i - w^T \underline{\Phi}(x_i) - a)^2$$

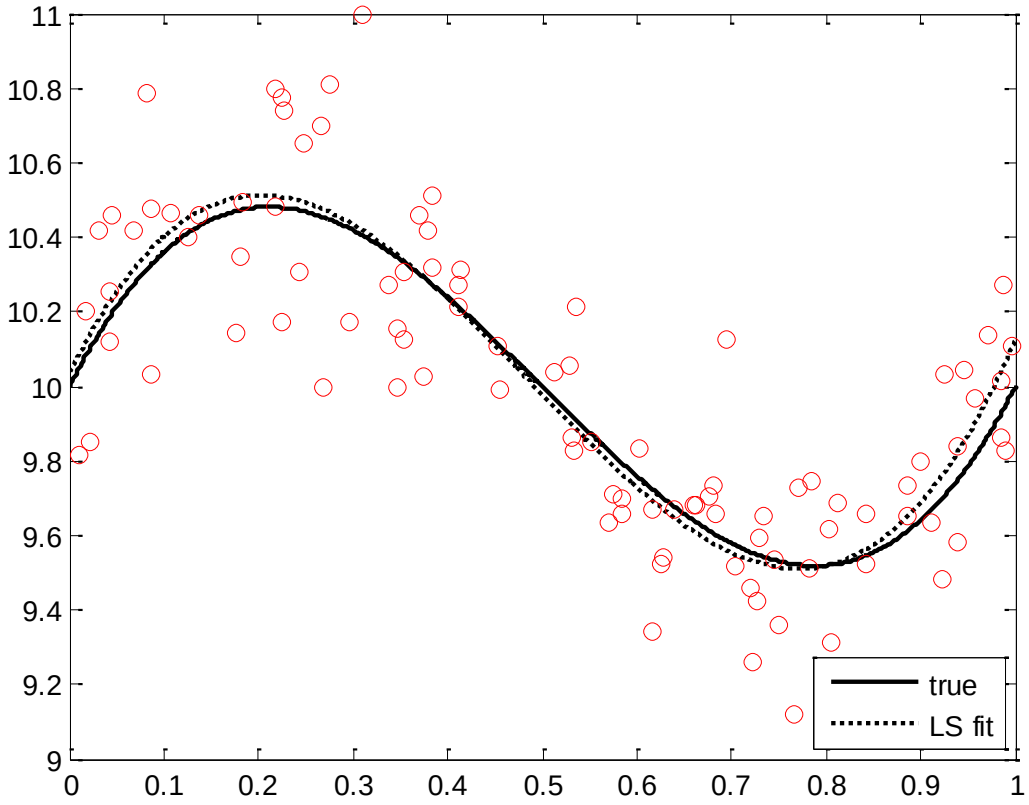
the minimizer is

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = (X^T X)^{-1} X^T y$$

where

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & & & \\ 1 & x_n & x_n^2 & x_n^3 \end{bmatrix}$$

$(n \times 1)$ $(n \times 4)$



Remark] This example is a good motivation for regularization. As the polynomial degree increases, the matrix $X^T X$ becomes ill-conditioned, and regularization is necessary to avoid overfitting.

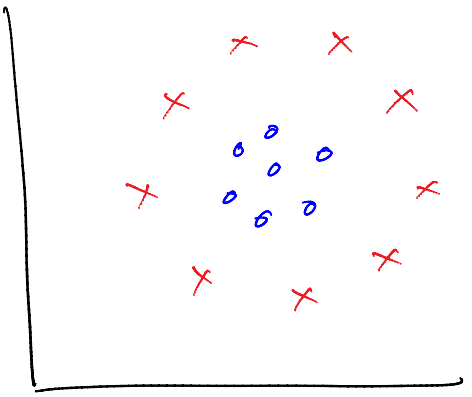
In classification, a nonlinear classifier can be expressed

$$y \longmapsto \text{sign}\left\{ w^T \underline{\Phi}(x) + b \right\}$$

Example 1 $\quad T x^{(1)} \;]$ $\longrightarrow$

Example

$$x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} \in \mathbb{R}^2$$



Consider

$$\underline{\Phi}(x) = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(1)} x^{(2)} \\ (x^{(1)})^2 \\ (x^{(2)})^2 \end{bmatrix}$$

The classes are separated by a circular classifier

$$y \longmapsto \text{sign}\left\{ (x^{(1)} - c^{(1)})^2 + (x^{(2)} - c^{(2)})^2 - r^2 \right\}$$

for a certain center $c = \begin{bmatrix} c^{(1)} \\ c^{(2)} \end{bmatrix}$ and radius r.

This is a linear classifier in the transformed space where

$$w = \begin{bmatrix} -2c^{(1)} \\ -2c^{(2)} \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad b = (c^{(1)})^2 + (c^{(2)})^2 - r^2$$

In both of the preceding examples, it is also possible to eliminate the offset term and just incorporate a constant into $\underline{\Phi}$.

Inner Product Kernels

One issue with the above approach is the way m explodes as d increases. You can't compute $\underline{\Phi}(x)$ explicitly. Fortunately, the following two facts save us:

- Many ML algorithms depend on $\underline{\Phi}(x)$ only via inner products $\langle \underline{\Phi}(x), \underline{\Phi}(x') \rangle$
- For certain $\underline{\Phi}$, the function

$$k(x,x') = \langle \underline{\Phi}(x), \underline{\Phi}(x') \rangle$$

can be computed efficiently even if m is huge or even infinite!

Example $d = 2$

$$k(u,v) = (u^T v)^2$$

$$= \left([u^{(1)} \; u^{(2)}] \begin{bmatrix} v^{(1)} \\ v^{(2)} \end{bmatrix} \right)^2$$

$$= \left(u^{(1)} v^{(1)} + u^{(2)} v^{(2)} \right)^2$$

$$= (u^{(1)})^2 (v^{(1)})^2 + 2u^{(1)}u^{(2)}v^{(1)}v^{(2)} + (v^{(1)})^2 (v^{(2)})^2$$

$$= \langle \Phi(u), \Phi(v) \rangle$$

where

$$\Phi(u) = \begin{bmatrix} (u^{(1)})^2 \\ \sqrt{2}\, u^{(1)} u^{(2)} \\ (u^{(2)})^2 \end{bmatrix}.$$

Example d arbitrary

$$k(u,v) = (u^T v)^2$$

$$= \left(\sum_{i=1}^{d} u^{(i)} v^{(i)} \right)^2$$

$$= \left(\sum_i u^{(i)} v^{(i)} \right) \left(\sum_j u^{(j)} v^{(j)} \right)$$

$$= \sum_{i=1}^{d} \sum_{j=1}^{d} u^{(i)} v^{(i)} u^{(j)} v^{(j)}$$

$$= \sum_{i=1}^{d} \sum_{j=1}^{d} u^{(i)} u^{(j)} v^{(i)} v^{(j)}$$

$$= \langle \underline{\phi}(u), \underline{\phi}(v) \rangle$$

where

$$\underline{\phi}(u) = \left[(u^{(1)})^2, \ldots, (u^{(d)})^2, \sqrt{2} u^{(1)} u^{(2)}, \ldots, \sqrt{2} u^{(d-1)} u^{(d)} \right]^T$$

so $m = d + \frac{d(d-1)}{2}$.

Example $d = 2$

$$k(u,v) = (u^T v)^3$$

$$= (u^{(1)})^3 (v^{(1)})^3 + 3 (u^{(1)})^2 u^{(2)} (v^{(1)})^2 v^{(2)}$$
$$+ 3 u^{(1)} (u^{(2)})^2 v^{(1)} (v^{(2)})^2 + (u^{(2)})^3 (v^{(2)})^3$$

$$= \sum_{i=0}^{3} \binom{3}{i} (u^{(1)})^{3-i} (u^{(2)})^i (v^{(1)})^{3-i} (v^{(2)})^i$$

$$= \langle \underline{\phi}(u), \underline{\phi}(v) \rangle$$

where

$$\underline{\phi}(u) = \left[(u^{(1)})^3, \sqrt{3} (u^{(1)})^2 u^{(2)}, \sqrt{3} u^{(1)} (u^{(2)})^2, (u^{(2)})^3 \right]^T.$$

Example Generalizing the above:

$$k(u,v) = (u^T v)^p$$

$$= \sum_{\substack{(j_1,\dots,j_d) \\ \sum j_i = p}} \binom{p}{j_1 \cdots j_d} (u^{(1)})^{j_1} \cdots (u^{(d)})^{j_d} (v^{(1)})^{j_1} \cdots (v^{(d)})^{j_d}$$

$\binom{p}{j_1 \cdots j_d}$ ← multinomial coefficient

$$\Longrightarrow \Phi(u) = \left[\dots, \sqrt{\binom{p}{j_1 \cdots j_d}} (u^{(1)})^{j_d} \cdots (u^{(2)})^{j_d}, \dots \right]^T$$

$\Longrightarrow$ all <u>monomials</u> of degree p.

In the preceeding examples, the inner product is just the standard dot product on $\mathbb{R}^d$.

More generally, a (real) <u>inner product space</u> is a vector space V on which we can define a function $\langle u, v \rangle$ (called an <u>inner product</u>) such that

(a) $\langle \alpha_1 u_1 + \alpha_2 u_2, v \rangle = \alpha_1 \langle u_1, v \rangle + \alpha_2 \langle u_2, v \rangle$

$\forall \alpha_1, \alpha_2 \in \mathbb{R}, \; u_1, u_2, v \in V$

(b) $\langle u, v \rangle = \langle v, u \rangle \quad \forall u, v \in V$

(c) $\langle u, u \rangle \geq 0 \quad \forall u \in V$, with equality iff $u = 0$.

We would like to know when it is possible to write a function k as

$$k(u, v) = \langle \Phi(u), \Phi(v) \rangle$$

for some inner product space V and $\Phi : \mathbb{R}^d \to V$.

This leads to the following definition:

We say $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is an inner product kernel if $\exists$ an inner product space V and a feature map $\Phi : \mathbb{R}^d \to V$ such that

$$k(u, v) = \langle \Phi(u), \Phi(v) \rangle \qquad \forall u, v \in \mathbb{R}^d.$$

Note: Φ and V are not unique

Positive Definite Kernels

One way to determine an IP kernel is to construct Φ explicitly as we did in the examples above.

We can also verify that k is an IP kernel if it satisfies the following properties.

Let $k: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. We say k is symmetric if $k(u,v) = k(v,u) \quad \forall u, v$. We say k is positive definite if k satisfies

$$[k(x_i, x_j)]_{ij} \text{ is PSD}$$

for all $x_1, \ldots, x_n \in \mathbb{R}^d$.

If k is both symmetric and positive definite, it is refered to as a symmetric, positive definite (SPD) kernel.

Theorem k is a SPD kernel $\iff$ k is an IP kernel.

Examples of Kernels

1. Homogeneous polynomial kernel

$$k(u,v) = (u^T v)^p,$$

2. Inhomogeneous polynomial kernel

$$k(u,v) = (u^T v + c)^p, \quad c > 0$$

3. Gaussian kernel

$$k(u,v) = C \exp\left(-\frac{1}{2\sigma^2}\|u-v\|^2\right), \quad C, \sigma > 0$$

For this kernel, V is infinite dimensional!

This is OK, though. Remember, we don't have to work with Φ, just k.

Big Picture – The Kernel Trick

Using kernels, we can obtain nonlinear methods from linear methods as follows.

1) Select an IP kernel k

2) Formulate your linear method such that feature vectors (i.e., the training data and a test instance) only appear via inner products $\langle x, x' \rangle$

3) Replace $\langle x, x' \rangle$ with $k(x,x')$ throughout the

algorithm.

This results in a method that is equivalent to applying the linear method to the nonlinearly transformed data $(\Phi(x_1), y_1), \ldots, (\Phi(x_n), y_n)$. With the above approach, however, we never have to compute Φ, just k.

We'll see examples of this idea in future lectures.