Kernel methods

Course of Machine Learning
Master Degree in Computer Science
University of Rome “Tor Vergata”

Giorgio Gambosi

a.a. 2017-2018

Dual representation

Many linear models for regression and classification can be reformulated in terms of a dual representation.

Example: regularized sum of squares

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}\left(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) - t_i\right)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$
$$= \frac{1}{2}(\boldsymbol{\Phi}\mathbf{w} - \mathbf{t})^T(\boldsymbol{\Phi}\mathbf{w} - \mathbf{t}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

setting $\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \sum_{i=1}^{n}\left(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) - t_i\right)\boldsymbol{\phi}(\mathbf{x}_i) + \lambda\mathbf{w} = \mathbf{0}$, the resulting solution is

$$\mathbf{w} = -\frac{1}{\lambda}\sum_{i=1}^{n}\left(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) - t_i\right)\boldsymbol{\phi}(\mathbf{x}_i) = \sum_{i=1}^{n} a_i\boldsymbol{\phi}(\mathbf{x}_i) = \boldsymbol{\Phi}^T\mathbf{a}$$

where

$$a_i = -\frac{1}{\lambda}\left(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) - t_i\right)$$

Dual representation

By substituting $\mathbf{\Phi}^T\mathbf{a}$ to $\mathbf{w}$ we express the cost function in terms of $\mathbf{a}$, instead of $\mathbf{w}$, introducing a dual representation of J.

$$
\begin{aligned}
J(\mathbf{a}) &= \frac{1}{2}(\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a}-\mathbf{t})^T(\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a}-\mathbf{t}) + \frac{\lambda}{2}(\mathbf{\Phi}^T\mathbf{a})^T\mathbf{\Phi}^T\mathbf{a} \\
&= \frac{1}{2}(\mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T - \mathbf{t}^T)(\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a}-\mathbf{t}) + \frac{\lambda}{2}\mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a} \\
&= \frac{1}{2}\left(\mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a} + \mathbf{t}^T\mathbf{t} - \mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{t} - \mathbf{t}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a}\right) + \frac{\lambda}{2}\mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a} \\
&= \frac{1}{2}\mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a} + \frac{1}{2}\mathbf{t}^T\mathbf{t} - \mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{a}
\end{aligned}
$$

Gram matrix

- Given Φ, let us define the Gram matrix as the symmetric matrix $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$
- The elements of the Gram matrix are the dot products

$$k_{ij} = \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\phi}(\mathbf{x}_j) = \sum_{l=1}^{m} \phi_l(\mathbf{x}_i)\phi_l(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

- $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function corresponding to the base functions $\boldsymbol{\phi}$.

Gram matrix

- The cost function can be written in terms of the Gram matrix as

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T\mathbf{K}\mathbf{K}\mathbf{a} + \frac{1}{2}\mathbf{t}^T\mathbf{t} - \mathbf{a}^T\mathbf{K}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^T\mathbf{K}\mathbf{a}$$

- setting the gradient of $J(\mathbf{a})$ wrt $\mathbf{a}$ to $\mathbf{0}$ it results

$$\frac{\partial J(\mathbf{a})}{\partial \mathbf{a}} = \mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{K}\mathbf{t} + \lambda\mathbf{K}\mathbf{a} = \mathbf{K}(\mathbf{K}\mathbf{a} - \mathbf{t} + \lambda\mathbf{a}) = \mathbf{K}((\mathbf{K} + \mathbf{I}\lambda)\mathbf{a} - \mathbf{t}) = \mathbf{0}$$

that is,

$$\mathbf{a} = (\mathbf{K} + \mathbf{I}\lambda)^{-1}\mathbf{t}$$

By substituting in the linear regression model, the prediction corresponding to a given in put $\mathbf{x}$ can be written as

$$y(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) = \mathbf{a}^T \boldsymbol{\Phi}\phi(\mathbf{x}) = \mathbf{t}^T (\mathbf{K} + \mathbf{I}\lambda)^{-1} \boldsymbol{\Phi}\phi(\mathbf{x}) = \mathbf{t}^T \mathbf{H}\mathbf{k}(\mathbf{x})$$

where

$$\begin{aligned}\mathbf{k}(\mathbf{x}) &= (\boldsymbol{\phi}(\mathbf{x}_1)^T \boldsymbol{\phi}(\mathbf{x}), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)^T \boldsymbol{\phi}(\mathbf{x}))^T \\ &= (\kappa(\mathbf{x}_1, \mathbf{x}), \ldots, \kappa(\mathbf{x}_n, \mathbf{x}))^T\end{aligned}$$

and

$$\mathbf{H} = (\mathbf{K} + \mathbf{I}\lambda)^{-1}$$

- The prediction for a new element $\mathbf{x}$ can be expressed just in terms of a linear combination of dot products of base functions (or, equivalently, of kernel functions)

$$y(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{a}_i \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}_i) = \sum_{i=1}^{n} \mathbf{a}_i \kappa(\mathbf{x}, \mathbf{x}_i)$$

  where

$$\mathbf{a}_i = \sum_{j=1}^{n} h_{ji} t_j$$

- observe that knowing the base functions $\boldsymbol{\phi}$ is sufficient to compute $y(\mathbf{x})$, but not necessary
- $y(\mathbf{x})$ can be computed also by just knowing the kernel function κ
- the kernel function can be derived from $\boldsymbol{\phi}$ (since $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \boldsymbol{\phi}(\mathbf{x}_1)^T \boldsymbol{\phi}(\mathbf{x}_2)$), but it is strictly less informative than the set of base functions, since it is not possible to derive $\boldsymbol{\Phi}$ from κ

Why referring to the dual representation?

- While in the original formulation of linear regression $\mathbf{w}$ can be derived by inverting the $m \times m$ matrix $\mathbf{\Phi}^T\mathbf{\Phi}$, in the dual formulation computing $\mathbf{a}$ requires inverting the $n \times n$ matrix $\mathbf{K} + \mathbf{I}\lambda$.
- Since usually $n \gg m$, this seems to lead to a loss of efficiency.
- However, the dual approach makes it possible to refer only to the kernel function κ, and not to the set of m base functions $\mathbf{\Phi}$: this makes it possible to implicitly use feature space of very high dimension (much larger than n, even infinite).

First approach

Choose a mapping of the feature space, in terms of a set of m base functions $\boldsymbol{\phi}$. Derive a kernel function as

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \boldsymbol{\phi}(\mathbf{x}_1)^T \boldsymbol{\phi}(\mathbf{x}_2) = \sum_{i=1}^{m} \phi_i(\mathbf{x}_1)\phi_i(\mathbf{x}_2)$$

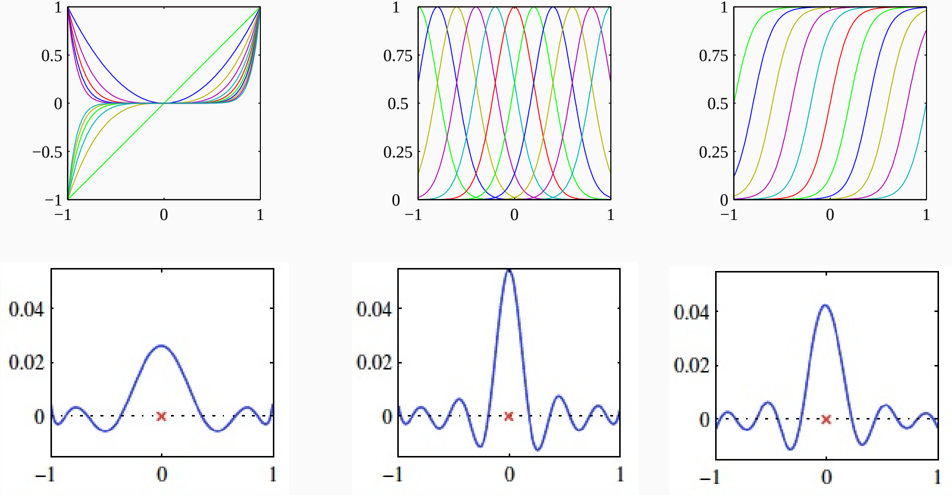
Second approach

Construct a kernel function directly: we must ensure that our function is a valid kernel function, that is it may be expressed as a scalar product in some (whatever high-dimensional) feature space resulting from the application of a set fo base functions.

That is, given κ, there must exist some mapping $\boldsymbol{\phi}$ such that $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \boldsymbol{\phi}(\mathbf{x}_1)^T \boldsymbol{\phi}(\mathbf{x}_2)$

Constructing kernel functions

Kernel functions from different types of base functions.



Example

Let $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^2$: $\kappa(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1^T \mathbf{x}_2)^2$ is a valid kernel function?

This can be verified by observing that

$$\begin{aligned}
\kappa(\mathbf{x}_1, \mathbf{x}_2) &= (x_{11}x_{21} + x_{12}x_{22})^2 \\
&= x_{11}^2 x_{21}^2 + x_{12}^2 x_{22}^2 + 2x_{11}x_{12}x_{21}x_{22} \\
&= (x_{11}^2, x_{12}^2, x_{11}x_{12}, x_{11}x_{12}) \cdot (x_{21}^2, x_{22}^2, x_{21}x_{22}, x_{21}x_{22}) \\
&= \boldsymbol{\phi}(\mathbf{x}_1) \cdot \boldsymbol{\phi}(\mathbf{x}_2)
\end{aligned}$$

and by defining the base functions as $\boldsymbol{\phi}(\mathbf{x}) = (x_1^2, x_2^2, x_1 x_2, x_1 x_2)^T$.

Constructing kernel functions

- In general, if $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$ then $\kappa(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 \cdot \mathbf{x}_2)^2 = \boldsymbol{\phi}(\mathbf{x}_1)^T \boldsymbol{\phi}(\mathbf{x}_2)$, where

$$\boldsymbol{\phi}(\mathbf{x}) = (x_1^2, \ldots, x_d^2, x_1 x_2, \ldots, x_1 x_d, x_2 x_1, \ldots, x_d x_{d-1})^T$$

- the d-dimensional input space is mapped onto a space with dimension $m = d^2$
- observe that computing $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ requires time $O(d)$, while deriving it from $\boldsymbol{\phi}(\mathbf{x}_1)^T \boldsymbol{\phi}(\mathbf{x}_2)$ requires $O(d^2)$ steps

Constructing kernel functions

The function $\kappa(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 \cdot \mathbf{x}_2 + c)^2$ is a kernel function. In fact,

$$\begin{aligned}
\kappa(\mathbf{x}_1, \mathbf{x}_2) &= (\mathbf{x}_1 \cdot \mathbf{x}_2 + c)^2 \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} x_{1i} x_{1j} x_{2i} x_{2j} + \sum_{i=1}^{n} (\sqrt{2} c x_{1i})(\sqrt{2} c x_{2i}) + c^2 \\
&= \boldsymbol{\phi}(\mathbf{x}_1)^T \boldsymbol{\phi}(\mathbf{x}_2)
\end{aligned}$$

for

$$\boldsymbol{\phi}(\mathbf{x}) = (x_1^2, \ldots, x_d^2, x_1 x_2, \ldots, x_1 x_d, x_2 x_1, \ldots, x_d x_{d-1}, \sqrt{2c} x_1, \ldots, \sqrt{2c} x_d, c)^T$$

This implies a mapping from a d-dimensional to a $(d+1)^2$-dimensional space.

Constructing kernel functions

Function $\kappa(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 \cdot \mathbf{x}_2 + c)^t$ is a kernel function corresponding to a mapping from a d-dimensional space to a space of dimension

$$m = \sum_{i=0}^{t} d^i = \frac{d^{t+1} - 1}{d - 1}$$

corresponding to all products $x_{i_1} x_{i_2} \ldots x_{i_l}$ with $0 \leq l \leq t$.

Observe that, even if the space has dimension $O(d^t)$, evaluating the kernel function requires just time $O(d)$.

Verifying a given function is a kernel

A necessary and sufficient condition for a function $\kappa : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$ to be a kernel is that, for all sets $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, the Gram matrix $\mathbf{K}$ such that $k_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ is semidefinite positive, that is

$$\mathbf{v}^T \mathbf{K} \mathbf{v} \geq 0$$

for all vectors $\mathbf{v}$.

Techniques for constructing kernel functions

Given kernel functions $\kappa_1(\mathbf{x}_1, \mathbf{x}_2)$, $\kappa_2(\mathbf{x}_1, \mathbf{x}_2)$, the function $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ is a kernel in all the following cases

- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = e^{\kappa_1(\mathbf{x}_1, \mathbf{x}_2)}$
- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \kappa_1(\mathbf{x}_1, \mathbf{x}_2) + \kappa_2(\mathbf{x}_1, \mathbf{x}_2)$
- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \kappa_1(\mathbf{x}_1, \mathbf{x}_2)\kappa_2(\mathbf{x}_1, \mathbf{x}_2)$
- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = c\kappa_1(\mathbf{x}_1, \mathbf{x}_2)$, for any $c > 0$
- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1^T \mathbf{A} \mathbf{x}_2$, with $\mathbf{A}$ positive definite
- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = f(\mathbf{x}_1)\kappa_1(\mathbf{x}_1, \mathbf{x}_2)g(\mathbf{x}_2)$, for any $f, g : \mathbb{R}^n \mapsto \mathbb{R}$
- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = p(\kappa_1(\mathbf{x}_1, \mathbf{x}_2))$, for any polynomial $p : \mathbb{R}^q \mapsto \mathbb{R}$ with non-negative coefficients
- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \kappa_3(\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2))$, for any vector $\boldsymbol{\phi}$ of m functions $\phi_i : \mathbb{R}^n \mapsto \mathbb{R}$ and for any kernel function $\kappa_3(\mathbf{x}_1, \mathbf{x}_2)$ in $\mathbb{R}^m$

Costructing kernel functions

$\kappa(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 \cdot \mathbf{x}_2 + c)^d$ is a kernel function. In fact,

1. $\mathbf{x}_1 \cdot \mathbf{x}_2 = \mathbf{x}_1^T \mathbf{x}_2$ is a kernel function corresponding to the base functions $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)$, with $\phi_i(\mathbf{x}) = \mathbf{x}$
2. c is a kernel function corresponding to the base functions $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)$, with $\phi_i(\mathbf{x}) = \dfrac{\sqrt{c}}{n}$
3. $\mathbf{x}_1 \cdot \mathbf{x}_2 + c$ is a kernel function since it is the sum of two kernel functions
4. $(\mathbf{x}_1 \cdot \mathbf{x}_2 + c)^d$ is a kernel function since it is a polynomial with non negative coefficients (in particular $p(z) = z^d$) of a kernel function

Costructing kernel functions

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = e^{-\frac{||\mathbf{x}_1 - \mathbf{x}_2||^2}{2\sigma^2}}$$

is a kernel function. In fact,

1. since $||\mathbf{x}_1 - \mathbf{x}_2||^2 = \mathbf{x}_1^T\mathbf{x}_1 + \mathbf{x}_2^T\mathbf{x}_2 - 2\mathbf{x}_1^T\mathbf{x}_2$, it results
$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = e^{-\frac{\mathbf{x}_1^T\mathbf{x}_1}{2\sigma^2}} e^{-\frac{\mathbf{x}_2^T\mathbf{x}_2}{2\sigma^2}} e^{\frac{\mathbf{x}_1^T\mathbf{x}_2}{\sigma^2}}$$
2. $\mathbf{x}_1^T\mathbf{x}_2$ is a kernel function (see above)
3. then, $\dfrac{\mathbf{x}_1^T\mathbf{x}_2}{\sigma^2}$ is a kernel function, being the product of a kernel function with a constant $c = \dfrac{1}{\sigma^2}$
4. $e^{\frac{\mathbf{x}_1^T\mathbf{x}_2}{\sigma^2}}$ is the exponential of a kernel function, and as a consequence a kernel function itself
5. $e^{-\frac{\mathbf{x}_1^T\mathbf{x}_1}{\sigma^2}} e^{-\frac{\mathbf{x}_1^T\mathbf{x}_1}{2\sigma^2}} e^{\frac{\mathbf{x}_1^T\mathbf{x}_2}{\sigma^2}}$ is a kernel function, being the product of a kernel function with two functions $f(\mathbf{x}_1) = e^{-\frac{\mathbf{x}_1^T\mathbf{x}_1}{2\sigma^2}}$ and $g(\mathbf{x}_2) = e^{-\frac{\mathbf{x}_2^T\mathbf{x}_2}{2\sigma^2}}$

Relevant kernel functions

1. Polynomial kernel
$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 \cdot \mathbf{x}_2 + 1)^d$$
2. Sigmoidal kernel
$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \tanh\left(c_1 \mathbf{x}_1 \cdot \mathbf{x}_2 + c_2\right)$$
3. Gaussian kernel
$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{||\mathbf{x}_1 - \mathbf{x}_2||^2}{2\sigma^2}\right)$$
where $\sigma \in \mathbb{R}$

Observe that a gaussian kernel can be derived also starting from a non linear kernel function $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ instead of $\mathbf{x}_1^T \mathbf{x}_2$.