Kernel

December 8, 2021

Road map

Important References

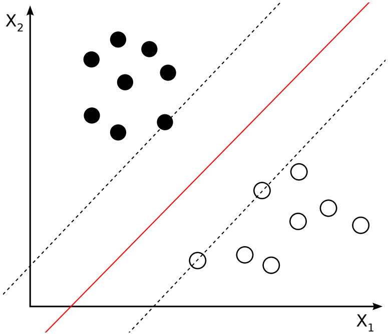
Books

- Learning with kernel
  `https://mitpress.mit.edu/books/learning-kernels`
- Kernel methods for Pattern Analysis. J. Shawe-Taylor et al.
- Reproducing kernel Hilbert spaces in Probability and Statistics. Berlinet et al.

Lecture notes

- J. Mairal and JP. Vert, MVA
- S. Canu, G. Gasso, B. Gauzere, INSA de Rouen

Linear SVM



- Inputs are vectors of $\mathbb{R}^d$
- Linear SVM seeks linear classification function

Limitations

- Data are not always vectors: (string, time series, graphs, images . . .)
- The decision function can not always be linear (text categorization; email filtering; gene detection; protein image classification; handwriting recognition; prediction of loan defaulting)

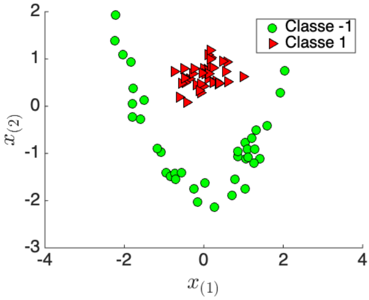
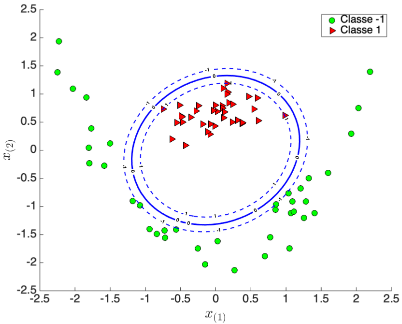


Figure: How do you classify these data?

From linear to non-linear decision function

Data might be separable with a non-linear function



- Non-linear embedding of $\mathrm{x} = \begin{pmatrix} x_{(1)} \\ x_{(2)} \end{pmatrix}$

$$\begin{array}{ccc} \mathbb{R}^2 & \rightarrow & \mathcal{H} \\ \mathrm{x} & \mapsto & \Phi(\mathrm{x}) = \begin{pmatrix} x_{(1)}^2 \\ x_{(2)}^2 \\ \sqrt{2}x_{(1)}x_{(2)} \end{pmatrix} \end{array}$$

- Train a linear SVM with samples $\{(\Phi(\mathrm{x}_i), y_i)\}$

Resulting SVM model

$$f(\mathrm{x}) = b + \sum_{i \in SV} \alpha_i y_i \Phi(\mathrm{x}_i)^\top \Phi(\mathrm{x})$$

Non-linear decision function

Decision function

$$f(\mathrm{x}) = b + \sum_{i \in SV} \alpha_i y_i \underbrace{\Phi(\mathrm{x}_i)^\top \Phi(\mathrm{x})}_{\text{Kernel } k(\mathrm{x}_i, \mathrm{x})}$$

Kernel function: the trick

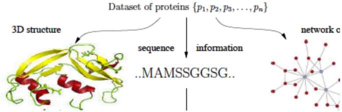
- No explicit knowledge of $\Phi(\mathrm{x})$
- We only need to define a function $k(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

Problem linearly non-separable in the original space $\mathcal{X}$

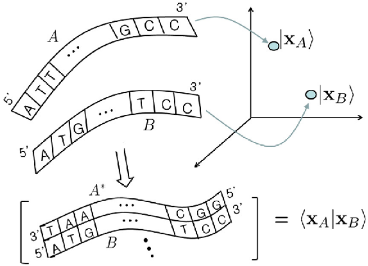
but linear separable in the space $\mathcal{H}$ induced by the kernel k

Also. . .

How do we classify dataset composed of proteins?



Use the kernel trick: $f(\text{protein}) = b + \sum_{i \in SV} \alpha_i y_i k(\text{protein}_i, \text{protein})$



Remark

Intuition

By simply modifying the dot product, the algorithm works in another space

Kernel Trick

1. Linear SVM relies on inner product between the SV and the sample to predict
2. $\rightarrow$ Replaces the inner product between the sample in the ambient space by a kernel $k(\cdot, \cdot)$
3. $\rightarrow$ Leads to a non-linear version of the SVM

Motivation of Kernel methods

- From
  - linear techniques
  - operating on vector spaces
- to
  - non linear prediction models
  - operating on various, structured, high-dimensional data
- Using a:
  well known mathematical framework
- leading to:
  efficient and powerful algorithm and tools

$\longrightarrow$ Kernel method

One slide summary

Motivation

- develop generic algorithms for analyzing and learning from data
- ... without making any assumptions on the type of the data

The approach

- Introducing framework based on similarities between pair of examples
- By appropriately defining the notion of similarity, we define a framework named as learning in RKHS.

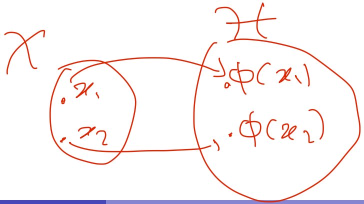
Prerequisites

decision function $f(x) = \sum_i \alpha_i \phi(x_i) \phi(x) = \boxed{\sum_{i=1}^{N} \alpha_i k(x_i, x)}$

Some definitions and notations

- $\mathcal{X}$: non empty input space ($\mathbb{R}^N$, graphs, objects, ...)
- $x \in \mathcal{X}$,
- $\mathcal{H}$: feature space endowed with a dot product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$
- $\Phi : \mathcal{X} \to \mathcal{H}$: embedding function from $\mathcal{X}$ to $\mathcal{H}$

$\to k(\cdot,\cdot)$



Kernel as a similarity

Kernel

A kernel k is a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that in some way $k(\mathrm{x}, \mathrm{z})$ captures the "similarity" between x and z.

Ideas

- the matrix K where each $\mathrm{K}_{i,j}$ represents all pairwise similarities
- For n data, K is of size $n \times n$ matrix of $\mathbb{R}$ regardless of the data type.
- Modularity between the data representation (through K and the choice of the algorithm (that uses K)

Positive Definite Kernels (1)

Positive definite kernel

A kernel $k(\mathrm{x}, \mathrm{z})$ on $\mathcal{X} \times \mathcal{X}$ is said to be positive definite

- if it is symmetric: $k(\mathrm{x}, \mathrm{z}) = k(\mathrm{z}, \mathrm{x})$
- and if for any finite positive integer n:

$$\forall \{\alpha_i\}_{i=1,n} \in \mathbb{R}, \forall \{\mathrm{x}_i\}_{i=1,n} \in \mathcal{X}, \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(\mathrm{x}_i, \mathrm{x}_j) \geq 0$$

$\alpha^T K \alpha \geq 0$

it is strictly positive definite if for $\alpha_i \neq 0$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(\mathrm{x}_i, \mathrm{x}_j) > 0$$

$\alpha^T K \alpha > 0$

let $K_{ij} = k(x_i, x_j)$, $K \in \mathbb{R}^{N \times N}$

equivalently, the gram matrix K are p.s.d.

Positive Definite Kernels (2)

Gram Matrix

Given a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and samples $\{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$, the Gram Matrix K is a $n \times n$ matrix with entries $\mathrm{K}_{i,j} := k(\mathrm{x}_i, \mathrm{x}_j)$

Another way to characterize positive definite kernel

For any set of $n \in \mathbb{N}$ samples $\{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$ the associated Gram Matrix $\mathrm{K} \in \mathbb{R}^{n \times n}$ is positive definite, iff k is a positive definite kernel on $\mathcal{X}$.

Kernel method

We define kernel methods as algorithms taking positive definite matrix as input.

Linear Kernel

$k(\mathrm{x}, \mathrm{z}) = \mathrm{x}^\top \mathrm{z}$

$\phi(x) = x$

- $\mathrm{x}, \mathrm{z} \in \mathbb{R}^d$
- symmetric: $\mathrm{x}^\top \mathrm{z} = \mathrm{x}^\top \mathrm{z}$
- positive:

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k(\mathrm{x}_i, \mathrm{x}_j) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j \mathrm{x}_i^\top \mathrm{x}_j \\ &= \left(\sum_{i=1}^{n}\alpha_i \mathrm{x}_i\right)^\top \left(\sum_{j=1}^{n}\alpha_j \mathrm{x}_j\right) \\ &= \left\|\sum_{i=1}^{n}\alpha_i \mathrm{x}_i\right\|^2\end{aligned}$$

Product kernel

$k(\mathrm{x}, \mathrm{z}) = g(\mathrm{x})g(\mathrm{z})$

x, z are graphs.
g(x) = nb of connected components in x

- $\mathrm{x}, \mathrm{z} \in \mathcal{X}$
- for some $g : \mathcal{X} \to \mathbb{R}$
- symmetric: by construction
- positive:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(\mathrm{x}_i, \mathrm{x}_j) = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j g(\mathrm{x}_i) g(\mathrm{x}_j)$$

$$= \left(\sum_{i=1}^{n} \alpha_i g(\mathrm{x}_i)\right)\left(\sum_{j=1}^{n} \alpha_j g(\mathrm{x}_j)\right)$$

$$= \left(\sum_{i=1}^{n} \alpha_i g(\mathrm{x}_i)\right)^2$$

Finite kernel

let $\phi_j, j = 1, p$ be a finite dictionary of functions from $\mathcal{X}$ to $\mathbb{R}$ (polynomials, wavelets...)

the feature map and linear kernel

feature map:
$$\begin{array}{rccl} \Phi: & \mathcal{X} & \to & \mathbb{R}^p \\ & \mathrm{x} & \mapsto & \Phi = \big(\phi_1(\mathrm{x}), ..., \phi_p(\mathrm{x})\big) \end{array}$$

kernel in the feature space is a positive definite kernel:
$$k(\mathrm{x}, \mathrm{z}) = \big(\phi_1(\mathrm{x}), ..., \phi_p(\mathrm{x})\big)^\top \big(\phi_1(\mathrm{z}), ..., \phi_p(\mathrm{z})\big)$$

$$\sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) \geq 0$$

The quadratic kernel

For $\mathrm{x}, \mathrm{z} \in \mathbb{R}^d$,, we define the feature map as

Feature map

$$\begin{array}{rccl} \Phi: & \mathbb{R}^d & \to & \mathbb{R}^{p=\frac{d(d+1)}{2}} \\ & \mathrm{x} & \mapsto & \Phi = \left(x_1^2, ..., x_j^2, ..., x_d^2, ..., \sqrt{2}x_i x_j, ...\right) \end{array}$$

Here the x_j represent the variables of $\mathrm{x} \in \mathbb{R}^d$

The kernel $k(\mathrm{x}, \mathrm{z})$ can be shown to be

$$k(\mathrm{x}, \mathrm{z}) = (\mathrm{x}^\top \mathrm{z})^2$$

Kernels as inner products

Theorem (Aronszajn, 1950)

k is a positive definite kernel on $\mathcal{X}$ if and only if,there exists a Hilbert space $\mathcal{H}$ and a mapping

$$\Phi : \mathcal{X} \mapsto \mathcal{H}$$

such that for any x and z in $\mathcal{X}$:

$$k(\mathrm{x}, \mathrm{z}) = \langle \Phi(\mathrm{x}), \Phi(\mathrm{z}) \rangle_{\mathcal{H}}$$

Reproducing Kernel Map

Preliminaries

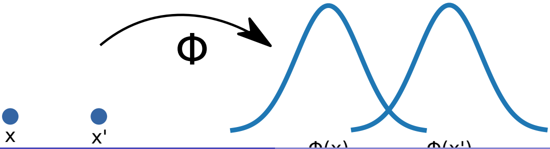
- We need a particular Hilbert space as defined by Aronszjan's theorem
- Φ maps each point $x \in \mathcal{X}$ to a function in $\mathcal{H}$
- $\mathcal{H}$ is a space of function from $\mathcal{X} \to \mathbb{R}$
- Build Φ from the psd kernel $k(\cdot, \cdot)$

Gaussian kernel or RBF kernel

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

$$\Phi : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}$$
$$x \mapsto k(\cdot, x)$$

- Example : $\mathcal{X} = \mathbb{R}$ and $k(\cdot, x) = y \mapsto e^{-\frac{(x-y)^2}{2\sigma^2}}$



RKHS Definition

Definition

Let $\mathcal{X}$ be a set and $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ be a class of function forming a real Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. The function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is called a reproducing kernel (r.k.) of $\mathcal{H}$ if

functions in F can be written as $f(x) = \sum \alpha_i k(x_i, x)$

1. $\mathcal{H}$ contains all functions of the form

$$\forall x \in \mathcal{X}, \quad k(x, \cdot) : z \mapsto k(x, z)$$

$$\Downarrow \quad \|f\|^2_{\mathcal{H}} = \langle \sum \alpha_i k(x_i, x), \sum \alpha_i k(x_i, x) \rangle = \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j)$$

2. For every $x \in \mathcal{X}$ and $f \in \mathcal{H}$, the reproducing property holds :

$$f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$$

$$\Rightarrow \|k(x,\cdot)\|_{\mathcal{H}} = \langle k(x,\cdot), k(x,\cdot) \rangle = k(x,x)$$

if a r.k exists, then $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS).

RKHS and Machine Learning

Using RKHS as a hypothesis space for machine learning problem leads to a simple recipe for non-linear models

1. maps data $x \in \mathcal{X}$ to a high-dimensional r.k. Hilbert space $\mathcal{H}$ through the mapping $\Phi : \mathcal{X} \mapsto \mathcal{H}$ with $\Phi(x) = k(x, \cdot)$
2. in $\mathcal{H}$, consider linear model with $f(x) = \langle f, k(x, \cdot)\rangle_{\mathcal{H}}$
3. use this linear model in your learning framework. For supervised learning, we would have

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}}^2$$

Uniqueness of r.k and RKHS

Theorem

- if $\mathcal{H}$ is a RKHS then it has an unique r.k
- A function $k(\cdot,\cdot)$ can be the r.k of at most one RKHS

RKHS, reproducing kernel and positive definite kernel

Theorem

A function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is positive definite if and only if it is a reproducing kernel.

Proof : an r.k is pd

- a r.k is symmetric
- a r.k leads to a pd matrix for any n subset of $\{\mathrm{x}_i\}_{i=1}^{n}$

A positive definite kernel is a reproducing kernel I

We aim to encode the image of Φ into a vector space

The vector space

Linear combinations of $k(\cdot, \mathrm{x})$:

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, \mathrm{x}_i) \quad \textit{with any} \quad \mathrm{x}_1, \dots, \mathrm{x}_m \in \mathcal{X}$$

The dot product

- given $g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, \mathrm{x}'_j)$ with $\mathrm{x}'_1, \dots, \mathrm{x}'_{m'} \in \mathcal{X}$ we define the dot product as

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(\mathrm{x}_i, \mathrm{x}'_j)$$

A positive definite kernel is a reproducing kernel II

We have:

$$f(\mathsf{x}'_j) = \sum \alpha_i k(\mathsf{x}'_j, \mathsf{x}_i) = \sum \alpha_i k(\mathsf{x}_i, \mathsf{x}'_j)$$

Hence:

$$\langle f, g \rangle = \sum \beta_j f(\mathsf{x}'_j)$$

Note that it does not depend on the expansion of f. Similarly we have

$$\langle f, g \rangle = \sum \alpha_i g(\mathsf{x}_i)$$

It is easy to show that our dot product is:

- bilinear
- symmetric
- positive definite

and thus constitutes a valid dot product on the vector space $\mathbb{R}^{\mathcal{X}}$.

A positive definite kernel is a reproducing kernel III

Reproducing kernel

$$\langle k(\cdot, \mathrm{x}), f \rangle = \sum \alpha_i k(\mathrm{x}_i, \mathrm{x}) = f(\mathrm{x})$$

Considering $f(\cdot) = k(\cdot, \mathrm{x}')$, we have:

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

k is thus the reproducing kernel of $\mathcal{H}$ and corresponds to a dot product in the vector space of functions $\mathcal{H}$.

The RKHS associated to a p.d kernel k.

Hilbert Space

- $\mathcal{H}$ is Hilbert Space obtained from
  - pre-Hilbert space of functions defined as above endowed with a inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ inducing a norm $\|f\| := \sqrt{\langle f, f \rangle}$
  - and the space completed with all limits of Cauchy sequences

$\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS) associated to kernel k

Smoothness functional

A simple inequality

- By Cauchy-Schwarz, we have for any function $f \in \mathcal{H}$ and two points $\mathrm{x}, \mathrm{z} \in \mathcal{X}$:

$$
\begin{aligned}
|f(\mathrm{x}) - f(\mathrm{z})| &= \langle f, k(\mathrm{x}, \cdot) - k(\mathrm{z}, \cdot) \rangle_{\mathcal{H}} \\
&\leq \| f \|_{\mathcal{H}} \times \| k(\mathrm{x}, \cdot) - k(\mathrm{z}, \cdot) \|_{\mathcal{H}} \\
&= \| f \|_{\mathcal{H}} \times d_k(\mathrm{x}, \mathrm{z})
\end{aligned}
$$

- the norm of a function f in the RKHS controls the variation of f over $\mathcal{X}$ with respect to the geometry defined by the kernel.

Take-home message

Small norm $\Rightarrow$ Small variations

RKHS

Positive kernel $\Leftrightarrow$ RKHS

- it defines the inner product
- it defines regularity (smoothness) of $\mathcal{H}$
- there exists a "mapping" function $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ such that $\forall \mathrm{x}, \mathrm{z} \in \mathcal{X}$ the inner product $\langle \Phi(\mathrm{x}), \Phi(\mathrm{z}) \rangle_{\mathcal{H}} = k(\mathrm{x}, \mathrm{z})$ and thus it defines the function space

Let's summarize

From kernel to feature space

Given a valid kernel k, we can associate a RKHS $\mathcal{H}$ which corresponds to the feature space of k.

From feature space to kernel

Now consider that you have $\Phi : \mathcal{X} \to \mathcal{H}$ a mapping function.
A positive kernel k is defined by:

$$k(\mathrm{x}, \mathrm{x}') = \langle \Phi(\mathrm{x}), \Phi(\mathrm{x}') \rangle_{\mathcal{H}}$$

Kernel algebra

Convex cone:

The set of kernels forms a convex cone, closed under pointwise convergence.

- **Linear combination**:
  - if k_1 an k_2 are kernels, a_1,$a_2 \geq 0$, then $a_1 k_1 + a_2 k_2$ is a kernel
  - if k_1, k_2, ... are kernels, and $k(\mathrm{x}, \mathrm{x}') := \lim_{n \to \infty} k_n(\mathrm{x}, \mathrm{x}')$ exists for all x, x′, then k is a kernel
- **Product kernel**:
  if k_1 an k_2 are kernels, then $k_1 k_2(\mathrm{x}, \mathrm{x}') := k_1(\mathrm{x}, \mathrm{x}') k_2(\mathrm{x}, \mathrm{x}')$ is a kernel.

Proofs

- by linearity:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\big(a_1k_1(i,j)+k_2(i,j)\big)= a_1\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k_1(i,j)+\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k_2(i,j)$$

- assuming $\quad \exists\psi_\ell$ s.t. $k_1(\mathrm{s},\mathrm{t})=\sum_\ell \psi_\ell(\mathrm{s})\psi_\ell(\mathrm{t})$

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\ k_1(\mathrm{x}_i,\mathrm{x}_j)k_2(\mathrm{x}_i,\mathrm{x}_j) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\Big(\sum_\ell \psi_\ell(\mathrm{x}_i)\psi_\ell(\mathrm{x}_j)k_2(\mathrm{x}_i,\mathrm{x}_j)\Big)\\ &= \sum_\ell\sum_{i=1}^{n}\sum_{j=1}^{n}\big(\alpha_i\psi_\ell(\mathrm{x}_i)\big)\ \big(\alpha_j\psi_\ell(\mathrm{x}_j)\big)\ k_2(\mathrm{x}_i,\mathrm{x}_j)\end{aligned}$$

Kernel engineering: building PDK

- for any polynomial with positive coef. ϕ from $\mathbb{R}$ to $\mathbb{R}$

$$\phi\big(k(s,t)\big)$$

- if Ψ is a function from $\mathbb{R}^d$ to $\mathbb{R}^d$

$$k\big(\Psi(s),\Psi(t)\big)$$

Example : the Gaussian kernel is a PDK

$$\begin{aligned} \exp(-\|s-t\|^2) &= \exp(-\|s\|^2 - \|t\|^2 - 2s^\top t) \\ &= \exp(-\|s\|^2)\exp(-\|t\|^2)\exp(2s^\top t) \end{aligned}$$

- $s^\top t$ is a PDK and function $\exp$ as the limit of positive series expansion, so $\exp(2s^\top t)$ is a PDK
- $\exp(-\|s\|^2)\exp(-\|t\|^2)$ is a PDK as a product kernel
- the product of two PDK is a PDK

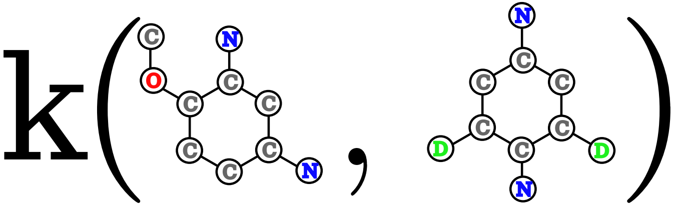
Positive definite kernels: some common examples

Type	Name	$k(\mathrm{x}, \mathrm{z})$
radial	Gaussian	$\exp\left(-\frac{\lVert \mathrm{x}-\mathrm{z}\rVert^2}{2\sigma^2}\right)$
radial	Laplacian	$\exp(-\lVert \mathrm{x}-\mathrm{z}\rVert/\sigma)$
non stat.	χ^2	$\exp(-r/\sigma), \; r = \sum_k \frac{(\mathrm{x}_k-\mathrm{z}_k)^2}{\mathrm{x}_k+\mathrm{z}_k}$
projectif	polynomial	$(\mathrm{x}^\top \mathrm{z} + \sigma)^p$
projectif	cosinus	$\mathrm{x}^\top \mathrm{z}/\lVert \mathrm{x}\rVert \lVert \mathrm{z}\rVert$
projectif	correlation	$\exp\left(\frac{\mathrm{x}^\top \mathrm{z}}{\lVert \mathrm{x}\rVert \lVert \mathrm{z}\rVert} - \sigma\right)$

- The kernel may involve hyper-parameter(s) to tune (polynom order p, bandwidth σ)
- Their value has to be set by cross-validation

Kernels on structures

- $\mathcal{X}$ may not be a vector space.
- we can define kernels on any kind of data :
  - Strings
  - Time series
  - Graphs
  - Images
  - ...



RKHS, kernel and machine learning : Kernel trick

Proposition

Any algorithm to process finite-dimensional vectors that can be expressed only in terms of pairwise inner-products can be applied to potentially infinite-dimensional vectors in the space of a p.d kernel by replacing each inner product evaluation by a kernel evaluation.

Applications

- replace inner product by kernel evaluation
- replace any inner-product induced norm by kernel evaluation

RKHS and machine learning : representer theorem

The representer theorem

- let $\mathcal{H}$ be a RKHS with k as associated kernel and $\mathcal{S} = \{\mathrm{x}_1, \cdots, \mathrm{x}_n\}$ be a finite set of points in $\mathcal{X}$
- let $\Psi : \mathbb{R}^{n+1} \mapsto \mathbb{R}$ be a function of $n+1$ variables, strictly increasing with respect to the last variable.
- Then, any solution to the optimization problem :

$$\min_{f \in clH} \Psi(f(\mathrm{x}_1), \cdots, f(\mathrm{x}_n), \|f\|_{\mathcal{H}})$$

admits a representation of the form:

$$f(\mathrm{x}) = \sum_{i=1}^{n} \alpha_i K(\mathrm{x}_i, \mathrm{x})$$

Using the Representer theorem

In practice

- When the representer theorem holds for our learning problem, then we can look for solution f of the form :

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\mathrm{x}_i, \cdot)$$

- Pointwise evaluation : for any $j = 1, \cdots \mathrm{n}$, we have

$$f(\mathrm{x}_j) = \sum_{i=1}^{n} \alpha_i k(\mathrm{x}_i, \mathrm{x}_j) = [\mathrm{K}\alpha]_j$$

- The norm

$$\|f\|_{\mathcal{H}}^2 = \langle f, f\rangle_{\mathcal{H}} = \sum_{i,j=1}^{n} \alpha_i \alpha_j k(\mathrm{x}_i, \mathrm{x}_j) = \alpha^\top \mathrm{K}\alpha$$

Using the Representer theorem

In practice

- A problem of the form

$$\min_{f \in \mathcal{H}} \Psi(f(x_1), \cdots, f(x_n), \|f\|_{\mathcal{H}})$$

  is equivalent to the n-dimensional optimization problem

$$\min_{\alpha \in \mathbb{R}^n} \Psi([K\alpha]_1, \cdots, [K\alpha]_n, \alpha^\top K\alpha)$$

- The resulting problem can be usually solved analytically or by numerical methods

Conclusion

What we seen

- Kernels corresponds to scalar product in some Hilbert space:
  - value corresponds to high dimensional scalar product,
  - on non linear embedding
  - without explicit representations of Φ
- Can be defined on any kind of data
- Kernels define the functional space (and its smoothness) in which we are looking for the solution

Key tools for ML

- Representer theorem
- Kernel trick