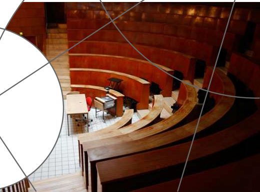
Kernels

Statistical Methods for Machine Learning

Christian Igel
Department of Computer Science

igel@diku.dk



Outline

Outline

Motivation

- Complexity of learning problem depends on representation – e.g., on data encoding
- Idea: make learning easier by changing representation
  $\mathcal{X}$: input space $\to$ $\mathcal{H}$: feature space
  Φ: $\mathcal{X} \to \mathcal{H}$ (feature map)
  and do the classification / regression in $\mathcal{H}$
- Both increasing and reducing the dimensionality can be reasonable
- Example: data may be separable by a linear function in $\mathcal{H}$

Example I

- Polynomial classifiers: suppose the n-dimensional $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^n$ are best represented by the dth order products (monomials) of the components x_j of $\boldsymbol{x}$, i.e., by the

$$x_{j_1} \cdot x_{j_2} \cdot \ldots \cdot x_{j_d} \quad ,$$

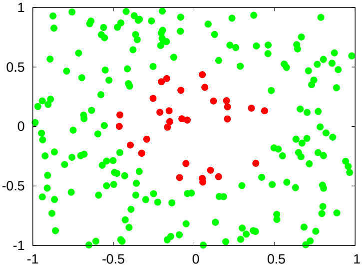
where $j_i, \ldots, j_d \in \{1, \ldots, n\}$

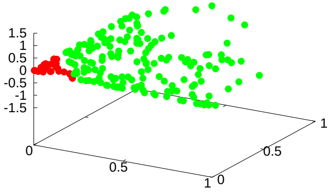
- Example: 2nd order monomials

$$\begin{aligned} \Phi_2 : \quad & \mathbb{R}^2 \to \mathbb{R}^3 \\ & \Phi_2((x_1, x_2)) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2) \end{aligned}$$

(here the order of monomials is not considered and a weighting factor is used)

Example II





(x_1, x_2)

$(x_1^2, x_2^2, \sqrt{2}x_1x_2)$

Curse of dimensionality / Kernel trick

- Problem: for n-dimensional $\mathcal{X}$ there exist

$$\binom{d+n-1}{d} = \frac{(d+n-1)!}{d!(n-1)!}$$

  dth order monomials

- Observation: many algorithms just require computing dot products $\langle \Phi(x), \Phi(x') \rangle$ in feature spaces ($\rightarrow$ perceptron, nearest neighbor, mean classifier)
- Idea: find efficient way to compute the dot product by a kernel

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

Kernel trick example

- Consider 2nd order monomials

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle^2 \qquad \text{(and } \langle \boldsymbol{x}, \boldsymbol{x}' \rangle^d \text{ for } d\text{th order)}$$

- Feature space is not unique

$$\begin{aligned}
\Phi_2 : \quad & \mathbb{R}^2 \to \mathbb{R}^3 \\
& \Phi_2((x_1, x_2)) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2) \\
\tilde{\Phi}_2 : \quad & \mathbb{R}^2 \to \mathbb{R}^4 \\
& \tilde{\Phi}_2((x_1, x_2)) = (x_1^2, x_2^2, x_1 x_2, x_2 x_1)
\end{aligned}$$

$$k(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{x}, \boldsymbol{z} \rangle^2 = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{z}) \rangle = \left\langle \tilde{\Phi}(\boldsymbol{x}), \tilde{\Phi}(\boldsymbol{z}) \right\rangle$$

Outline

Vector space

A set $\mathcal{H}$ is called a *vector space over* $\mathbb{R}$ if addition and scalar multiplication are defined, and satisfy $\forall \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{H}, \lambda, \lambda' \in \mathbb{R}$:

1. $\boldsymbol{x} + (\boldsymbol{x}' + \boldsymbol{x}'') = (\boldsymbol{x} + \boldsymbol{x}') + \boldsymbol{x}''$,
2. $\boldsymbol{x} + \boldsymbol{x}' = \boldsymbol{x}' + \boldsymbol{x} \in \mathcal{H}$,
3. $\mathbf{0} \in \mathcal{H}$, $\boldsymbol{x} + \mathbf{0} = \boldsymbol{x}$,
4. $-\boldsymbol{x} \in \mathcal{H}$, $-\boldsymbol{x} + \boldsymbol{x} = \mathbf{0}$,
5. $\lambda \boldsymbol{x} \in \mathcal{H}$,
6. $1\boldsymbol{x} = \boldsymbol{x}$,
7. $\lambda(\lambda' \boldsymbol{x}) = (\lambda\lambda')\boldsymbol{x}$,
8. $\lambda(\boldsymbol{x} + \boldsymbol{x}') = \lambda\boldsymbol{x} + \lambda\boldsymbol{x}'$,
9. $(\lambda + \lambda')\boldsymbol{x} = \lambda\boldsymbol{x} + \lambda'\boldsymbol{x}$.

Dot product

A *symmetric bilinear form* on a vector space $\mathcal{H}$ is a symmetric function $Q : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ with the property that $\forall \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{H}, \lambda, \lambda' \in \mathbb{R}$ we have

$$Q((\lambda \boldsymbol{x} + \lambda' \boldsymbol{x}'), \boldsymbol{x}'') = \lambda Q(\boldsymbol{x}, \boldsymbol{x}'') + \lambda' Q(\boldsymbol{x}', \boldsymbol{x}'') \enspace .$$

A *dot product* on a vector space $\mathcal{H}$ is a symmetric bilinear form $\langle ., . \rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ that is strictly positive definite, i.e., $\forall \boldsymbol{x} \in \mathcal{H} : \langle \boldsymbol{x}, \boldsymbol{x} \rangle \geq 0$ with equality only for $\boldsymbol{x} = \boldsymbol{0}$.

Any dot product defines a corresponding norm via $\|\boldsymbol{x}\| := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ and norm defines a metric d via $d(\boldsymbol{x}, \boldsymbol{x}') := \|\boldsymbol{x} - \boldsymbol{x}'\|$.

Positive definite matrix

A real symmetric $m \times m$ matrix $\boldsymbol{K}$ satisfying

$$\forall c_1, \ldots, c_m \in \mathbb{R} : \sum_{i,j=1}^{m} c_i c_j K_{ij} \geq 0$$

or equivalently

$$\forall \boldsymbol{x} \in \mathbb{R}^m : \boldsymbol{x}^T \boldsymbol{K} \boldsymbol{x} \geq 0$$

is called *positive definite*.

Positive definite kernels

Given a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and patterns $x_1, \dots, x_m \in \mathcal{X}$, the $m \times m$ matrix $\boldsymbol{K}$ with elements

$$K_{ij} = k(x_i, x_j)$$

is called *Gram / kernel matrix* of k with respect to $x_1, \dots, x_m$.

A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\mathcal{X} \neq \emptyset$, which for all $m \in \mathbb{N}$ and all $x_1, \dots, x_m \in \mathcal{X}$ gives raise to a positive definite Gram matrix is called a *(positive definite) kernel*.

Outline

Kernel functions and feature spaces

- Given some kernel k, can we construct a feature space $\mathcal{H}$ such that k computes the dot product in $\mathcal{H}$?

- Given a mapping Φ into a feature space $\mathcal{H}$, can we find a kernel computing the dot product in $\mathcal{H}$?

Function spaces

A function space is a space of made of functions. Each function in this space can be thought of as a point.

Example: L_2, the set of all square integrable functions $f : \mathbb{R} \to \mathbb{R}$

f is square integrable iff $\int f^2(x)\mathrm{d}x < \infty$.

Kernel to feature map

1. Define map Φ given kernel k
2. Turn image of Φ into vector space
3. Define dot product
4. Show that $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$

1. & 2.: Feature map & vector space

Feature map:

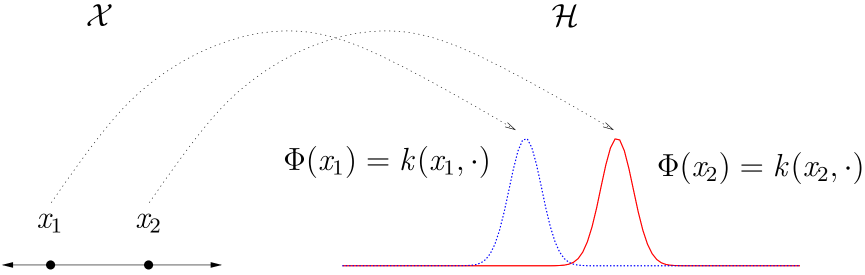
$$\begin{aligned} \Phi: \quad & \mathcal{X} \to \mathbb{R}^{\mathcal{X}} := \{f : \mathcal{X} \to \mathbb{R}\} \\ & \Phi(x)(\cdot) = k(\cdot, x) \end{aligned}$$

Vector space:
$\mathrm{span}\{k(x,\cdot) \,|\, x \in \mathcal{X}\}$ consisting of all functions

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i)$$

for any $m \in \mathbb{N}$ and $x_1, \dots, x_m \in \mathcal{X}$, $\alpha_1, \dots, \alpha_m \in \mathbb{R}$

Mapping points to functions



3. & 4.: Dot product & equivalence

Dot product: well-defined, symmetric, bilinear, positive definite

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \qquad g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x'_j)$$

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

$$\langle f, f \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \alpha_j k(x_i, x_j) \geq 0$$

We have

$$\langle k(\cdot, x), f \rangle = f(x) \qquad \text{(reproducing property)}$$

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

Feature map to kernel

Given $\Phi : \mathcal{X} \to \mathcal{H}$ we define

$$k(x, x') := \langle \Phi(x), \Phi(x') \rangle \quad ,$$

which is positive definite as for all $m \in \mathbb{N}, c_i \in \mathbb{R}, x_i \in \mathcal{X}, i = 1, \dots, m$ and obeys:

$$\sum_{i,j=1}^{m} c_i c_j k(x_i, x_j) = \left\langle \sum_{i=1}^{m} c_i \Phi(x_i), \sum_{j=1}^{m} c_j \Phi(x_j) \right\rangle$$
$$= \left\| \sum_{i=1}^{m} c_i \Phi(x_i) \right\|^2 \geq 0$$

Kernel trick

Given an algorithm formulated in terms of a positive definite kernel k, one can construct an alternative algorithm by replacing k by an alternative kernel.

Outline

Hilbert spaces

A space is called *complete* if all Cauchy sequences in the space converge. A *Hilbert space* is a complete space endowed with a dot product.

In Hilbert spaces orthogonal projections onto closed subspaces exist.

Examples:

- $\mathbb{R}^n$ is a Hilbert space,
- L_2 is a Hilbert space (but no RKHS)

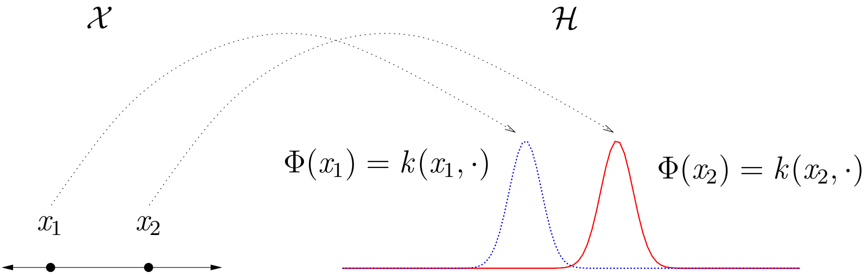
RKHS

A Hilbert space $\mathcal{H}$ of functions $f : \mathcal{X} \to \mathbb{R}$, $\mathcal{X} \neq \emptyset$ is called a *reproducing kernel Hilbert space* (RKHS) with dot product $\langle ., . \rangle$ and norm $\|f\| := \sqrt{\langle f, f \rangle}$ if there is a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

1. satisfying $\langle f, k(x, \cdot) \rangle = f(x)$ for all $f \in \mathcal{H}$ and
2. spanning $\mathcal{H}$, i.e., $\mathcal{H} = \overline{\mathrm{span}\{k(x, \cdot) \,|\, x \in \mathcal{X}\}}$.

The RKHS uniquely determines k ($\langle k(x, \cdot), k'(x, \cdot) \rangle = k(x, x) = k'(x, x)$).

RKHS feature mapping



Projections

Let $\mathcal{H}$ be a Hilbert space and M a closed subspace. Then every $\boldsymbol{x} \in \mathcal{H}$ can be written uniquely as $\boldsymbol{x} = \boldsymbol{z} + \boldsymbol{z}_\perp$, where $\boldsymbol{z} \in M$ and $\langle \boldsymbol{z}_\perp, \boldsymbol{t} \rangle = 0$ for all $\boldsymbol{t} \in M$. The vector $\boldsymbol{z}$ is the unique element of M minimizing $\|\boldsymbol{x} - \boldsymbol{z}\|$; it is called the (orthogonal) projection of $\boldsymbol{x}$ onto M.

In a RKHS $\mathcal{H}$ with kernel k on $\mathcal{X}$, the projection of $\Phi(x) = k(x, \cdot)$, $x \in \mathcal{X}$, onto $\boldsymbol{w} \in \mathcal{H}$ is given by

$$\frac{\langle \boldsymbol{w}, \Phi(x) \rangle}{\|\boldsymbol{w}\|^2} \boldsymbol{w} \quad .$$

Outline

Examples of kernels

Let $\mathcal{X} = \mathbb{R}^n$. Then typical kernels are:

- Gaussian kernels

$$k(\boldsymbol{x}, \boldsymbol{z}) = e^{-(\boldsymbol{x}-\boldsymbol{z})^T \boldsymbol{M} (\boldsymbol{x}-\boldsymbol{z})}$$

  with positive definite matrix $\boldsymbol{M}$, e.g., $\boldsymbol{M} = \gamma \boldsymbol{I}, \gamma > 0$
  (corresponding Gram matrices always have full rank)

- Polynomial kernels

$$k(\boldsymbol{x}, \boldsymbol{z}) = (\langle \boldsymbol{x}, \boldsymbol{z} \rangle + c)^d$$

  including the linear kernel

$$k(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{x}, \boldsymbol{z} \rangle$$

Making kernels from kernels

Let $k_1, k_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $k_3 : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ be kernels. Let $a \in \mathbb{R}^+$, $f : \mathcal{X} \to \mathbb{R}$ and $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^m$. Then the following functions are kernels:

1. $k(x, z) = ak_1(x, z)$,
2. $k(x, z) = k_1(x, z) + k_2(x, z)$,
3. $k(x, z) = k_1(x, z)k_2(x, z)$,
4. $k(x, z) = e^{k_1(x,z)}$,
5. $k(x, z) = f(x)f(z)$,
6. $k(x, z) = k_3(\boldsymbol{\phi}(x), \boldsymbol{\phi}(z))$,
7. $k(x, z) = \frac{k_1(x,z)}{\sqrt{k_1(x,x)k_1(z,z)}}$.

Summary

- Kernel trick allows efficient formulation of nonlinear variants of any algorithm that can be expressed in terms of dot products.
- For any positive definite kernel, a RKHS can be constructed.
- The kernel defines the feature space, especially neighborhood relations. Choosing a proper kernel is crucial for the performance of a kernel-based algorithm.
- Kernel functions provide a clean interface between general and problem specific aspects of the learning machine.

References:

B. Schölkopf and A. J. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.