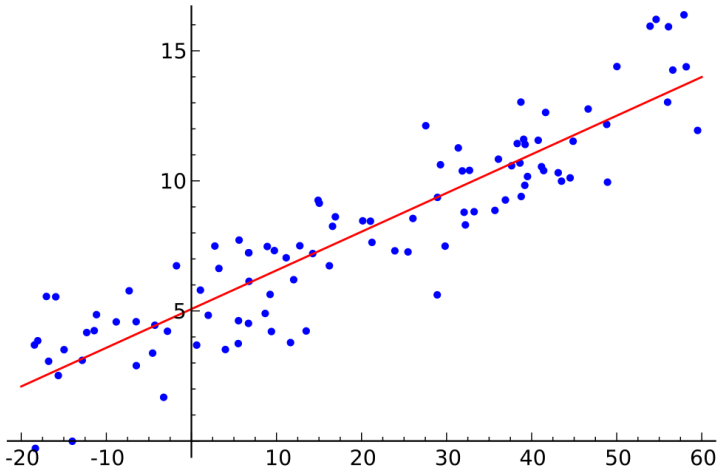
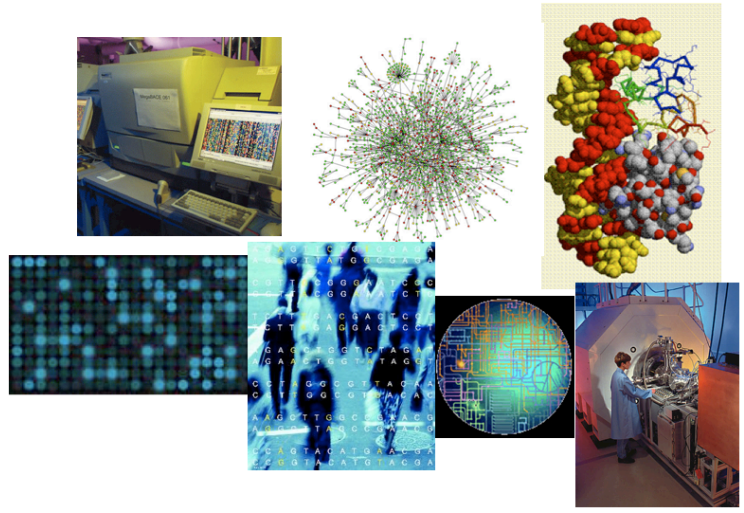
Machine Learning with Kernel Methods

Michael Arbel

The content of the course is adapted from a course at MVA.

Main goal of this course





Extend

well-understood, linear statistical learning techniques

to

real-world, complicated, structured, high-dimensional data

based on

a rigorous mathematical framework

leading to

practical modelling tools and algorithms

Organization of the course

Contents

1. Present the basic mathematical theory of kernel methods.
2. Introduce algorithms for supervised and unsupervised machine learning with kernels.
3. Develop a working knowledge of kernel engineering for specific data and applications (graphs, biological sequences, images).
4. Discuss open research topics related to kernels such as large-scale learning with kernels and "deep kernel learning".

Outline

Outline

1. Kernels and RKHS
   - Positive Definite Kernels
   - Reproducing Kernel Hilbert Spaces (RKHS)
   - Examples
   - Smoothness functional

2. Kernel tricks and applications
   - The kernel trick
   - The representer theorem
   - Kernel ridge regression
   - Kernel logistic regression
   - Kernel PCA

Outline

Part 1

Kernels and RKHS

Overview

Motivations

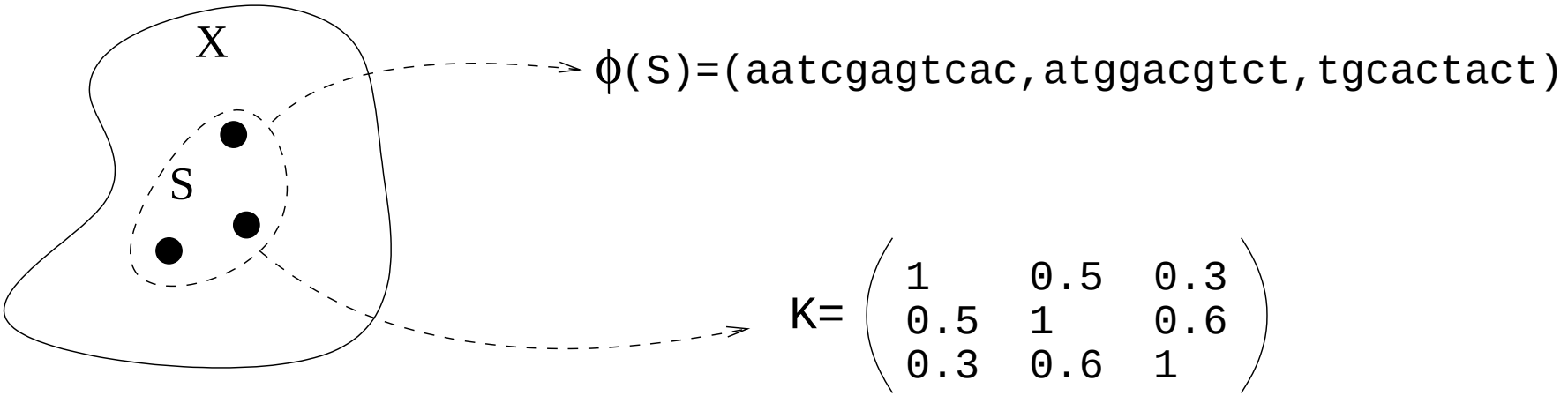
- Develop versatile algorithms to process and analyze data...
- ...without making any assumptions regarding the type of data (vectors, strings, graphs, images, ...)

The approach

- Develop methods based on pairwise comparisons.
- By imposing constraints on the pairwise comparison function (positive definite kernels), we obtain a general framework for learning from data (optimization in RKHS).

Outline

Representation by pairwise comparisons



Idea

- Define a "comparison function": $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$.
- Represent a set of n data points $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ by the $n \times n$ matrix:

$$[\mathbf{K}]_{ij} := K(\mathbf{x}_i, \mathbf{x}_j).$$

Representation by pairwise comparisons

Remarks

- $\mathbf{K}$ is always an $n \times n$ matrix, whatever the nature of data: the same algorithm will work for any type of data (vectors, strings, ...).
- Total modularity between the choice of function K and the choice of the algorithm.
- Poor scalability with respect to the dataset size (n^2 to compute and store $\mathbf{K}$)... but wait until the end of the course to see how to deal with large-scale problems
- We will restrict ourselves to a particular class of pairwise comparison functions.

Positive Definite (p.d.) Kernels

Definition

A positive definite (p.d.) kernel on a set $\mathcal{X}$ is a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that is symmetric:

$$\forall \left(\mathbf{x}, \mathbf{x}'\right) \in \mathcal{X}^2, \quad K\left(\mathbf{x}, \mathbf{x}'\right) = K\left(\mathbf{x}', \mathbf{x}\right),$$

and which satisfies, for all $N \in \mathbb{N}$, $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathcal{X}^N$ and $(a_1, a_2, \ldots, a_N) \in \mathbb{R}^N$:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \geq 0.$$

Similarity matrices of p.d. kernels

Remarks

- Equivalently, a kernel K is p.d. if and only if, for any $N \in \mathbb{N}$ and any set of points $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathcal{X}^N$, the similarity matrix $[\mathbf{K}]_{ij} := K(\mathbf{x}_i, \mathbf{x}_j)$ is positive semidefinite.
- Kernel methods are algorithms that take such matrices as input.

The simplest p.d. kernel, for real numbers

Lemma

Let $\mathcal{X} = \mathbb{R}$. The function $K : \mathbb{R}^2 \mapsto \mathbb{R}$ defined by:

$$\forall \left(x, x'\right) \in \mathbb{R}^2, \quad K\left(x, x'\right) = xx'$$

is p.d.

The simplest p.d. kernel, for real numbers

Lemma

Let $\mathcal{X} = \mathbb{R}$. The function $K : \mathbb{R}^2 \mapsto \mathbb{R}$ defined by:

$$\forall \left(x, x'\right) \in \mathbb{R}^2, \quad K\left(x, x'\right) = xx'$$

is p.d.

Proof:

- $xx' = x'x$
- $\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j x_i x_j = \left(\sum_{i=1}^{N} a_i x_i\right)^2 \geq 0$ □

The simplest p.d. kernel, for vectors

Lemma

Let $\mathcal{X} = \mathbb{R}^d$. The function $K : \mathcal{X}^2 \mapsto \mathbb{R}$ defined by:

$$\forall \left(\mathbf{x}, \mathbf{x}'\right) \in \mathcal{X}^2, \quad K\left(\mathbf{x}, \mathbf{x}'\right) = \left\langle \mathbf{x}, \mathbf{x}' \right\rangle_{\mathbb{R}^d}$$

is p.d. (it is often called the linear kernel).

The simplest p.d. kernel, for vectors

Lemma

Let $\mathcal{X} = \mathbb{R}^d$. The function $K : \mathcal{X}^2 \mapsto \mathbb{R}$ defined by:

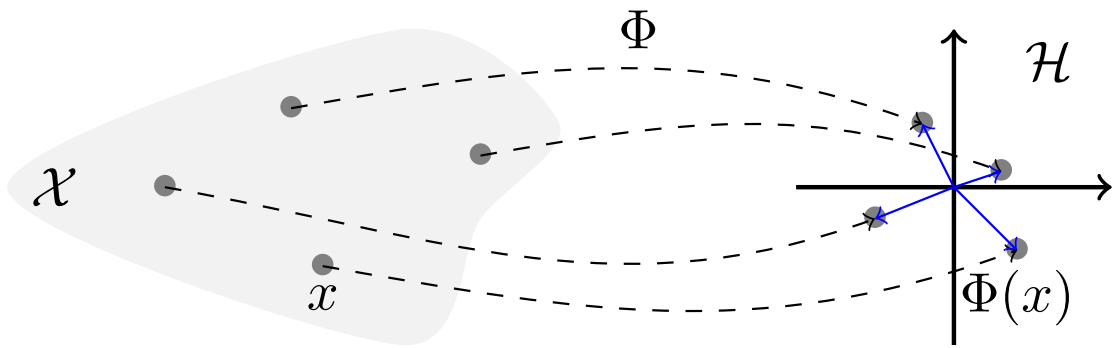
$$\forall \left(\mathbf{x}, \mathbf{x}'\right) \in \mathcal{X}^2, \quad K\left(\mathbf{x}, \mathbf{x}'\right) = \left\langle \mathbf{x}, \mathbf{x}' \right\rangle_{\mathbb{R}^d}$$

is p.d. (it is often called the linear kernel).

Proof:

- $\langle \mathbf{x}, \mathbf{x}' \rangle_{\mathbb{R}^d} = \langle \mathbf{x}', \mathbf{x} \rangle_{\mathbb{R}^d}$
- $\sum_{i=1}^N \sum_{j=1}^N a_i a_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle_{\mathbb{R}^d} = \| \sum_{i=1}^N a_i \mathbf{x}_i \|_{\mathbb{R}^d}^2 \geq 0$ □

A more ambitious p.d. kernel

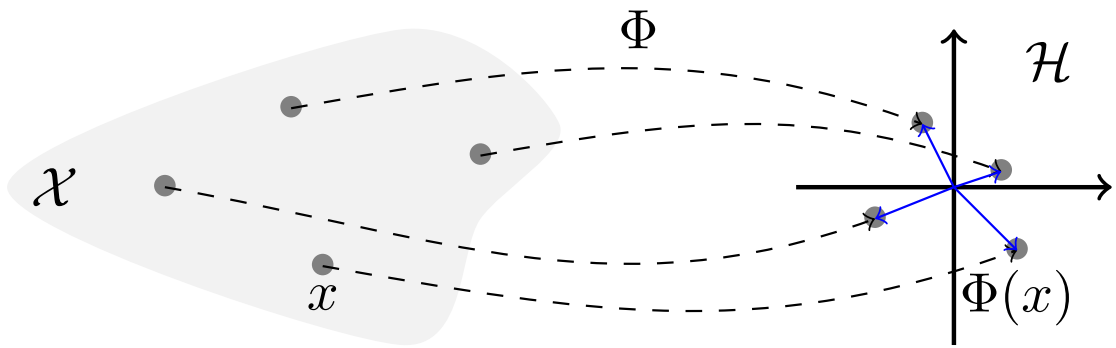


Lemma

Let $\mathcal{X}$ be any set, and $\Phi : \mathcal{X} \mapsto \mathbb{R}^d$. Then, the function $K : \mathcal{X}^2 \mapsto \mathbb{R}$ defined as follows is p.d.:

$$\forall \left(\mathbf{x}, \mathbf{x}'\right) \in \mathcal{X}^2, \quad K\left(\mathbf{x}, \mathbf{x}'\right) = \left\langle \Phi\left(\mathbf{x}\right), \Phi\left(\mathbf{x}'\right) \right\rangle_{\mathbb{R}^d} .$$

A more ambitious p.d. kernel



Lemma

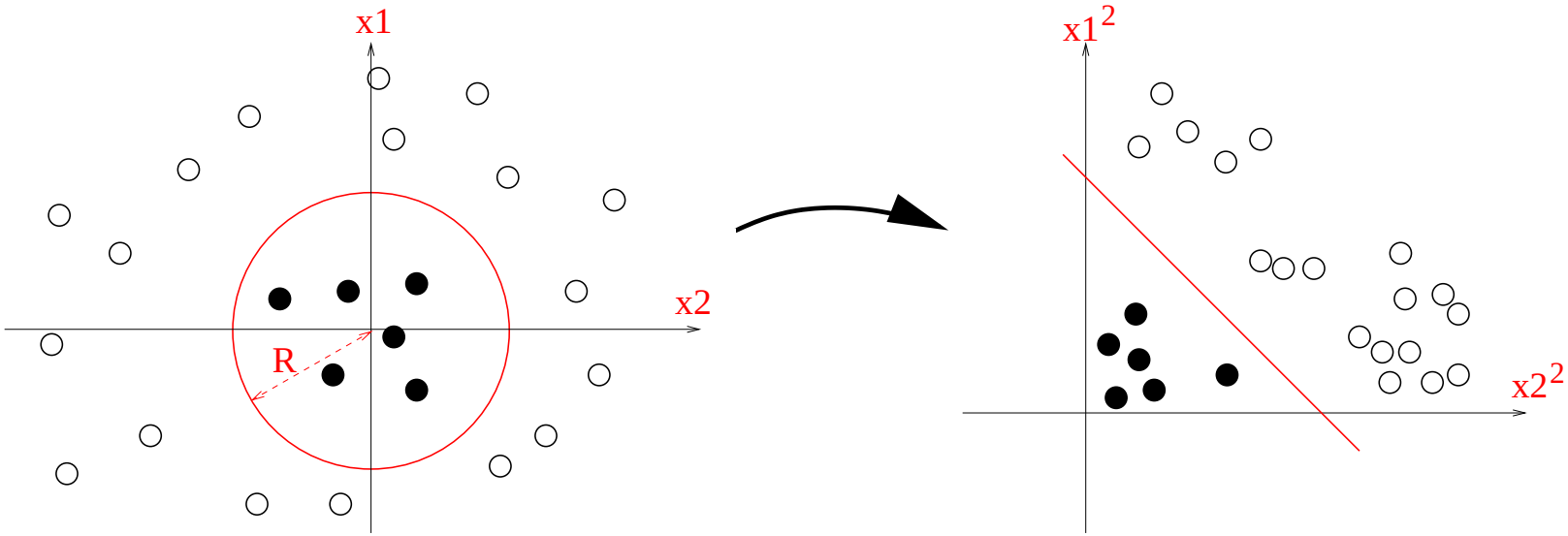
Let $\mathcal{X}$ be any set, and $\Phi : \mathcal{X} \mapsto \mathbb{R}^d$. Then, the function $K : \mathcal{X}^2 \mapsto \mathbb{R}$ defined as follows is p.d.:

$$\forall \left(\mathbf{x}, \mathbf{x}'\right) \in \mathcal{X}^2, \quad K\left(\mathbf{x}, \mathbf{x}'\right) = \left\langle \Phi\left(\mathbf{x}\right), \Phi\left(\mathbf{x}'\right) \right\rangle_{\mathbb{R}^d} .$$

Proof:

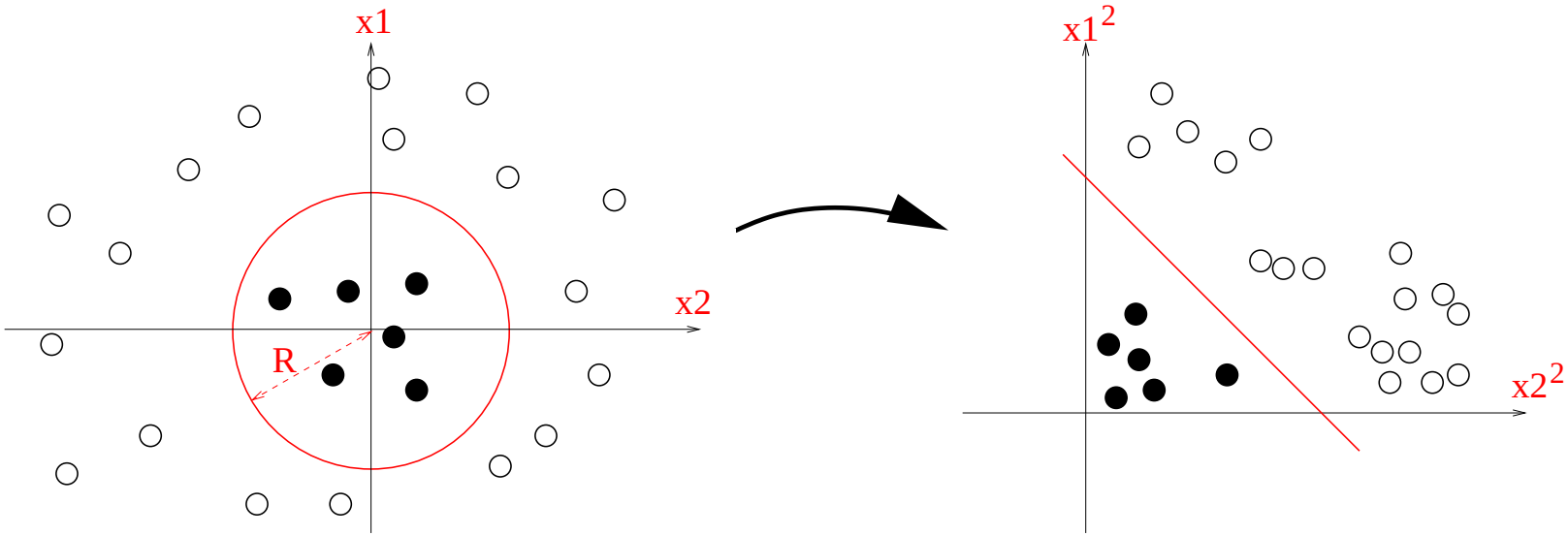
- $\left\langle \Phi\left(\mathbf{x}\right), \Phi\left(\mathbf{x}'\right) \right\rangle_{\mathbb{R}^d} = \left\langle \Phi\left(\mathbf{x}'\right), \Phi\left(\mathbf{x}\right) \right\rangle_{\mathbb{R}^d}$
- $\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j \left\langle \Phi\left(\mathbf{x}_i\right), \Phi\left(\mathbf{x}_j\right) \right\rangle_{\mathbb{R}^d} = \| \sum_{i=1}^{N} a_i \Phi\left(\mathbf{x}_i\right) \|_{\mathbb{R}^d}^2 \geq 0$ □

Example: polynomial kernel



For $\mathbf{x} = (x_1, x_2)^\top \in \mathbb{R}^2$, let $\Phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \in \mathbb{R}^3$:

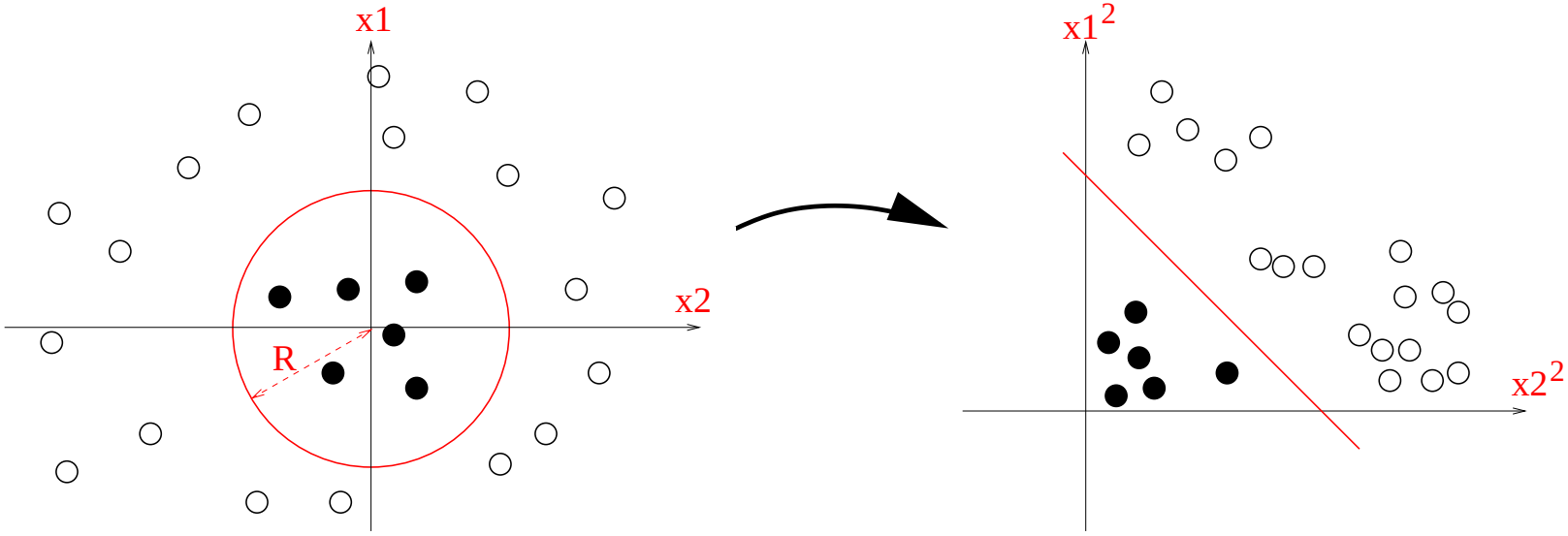
Example: polynomial kernel



For $\mathbf{x} = (x_1, x_2)^\top \in \mathbb{R}^2$, let $\Phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \in \mathbb{R}^3$:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= x_1^2 x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2 x_2'^2 \\ &= \left(x_1x_1' + x_2x_2'\right)^2 \\ &= \left\langle \mathbf{x}, \mathbf{x}' \right\rangle_{\mathbb{R}^2}^2 \ . \end{aligned}$$

Example: polynomial kernel



For $\mathbf{x} = (x_1, x_2)^\top \in \mathbb{R}^2$, let $\Phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \in \mathbb{R}^3$:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= x_1^2 x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2 x_2'^2 \\ &= \left(x_1x_1' + x_2x_2'\right)^2 \\ &= \left\langle \mathbf{x}, \mathbf{x}' \right\rangle_{\mathbb{R}^2}^2 \ . \end{aligned}$$

Exercise: show that $\langle \mathbf{x}.\mathbf{x}' \rangle_{\mathbb{R}^p}^d$ *is p.d. on* $\mathcal{X} = \mathbb{R}^p$ *for any* $d \in \mathbb{N}$.

Conversely: Kernels as inner products

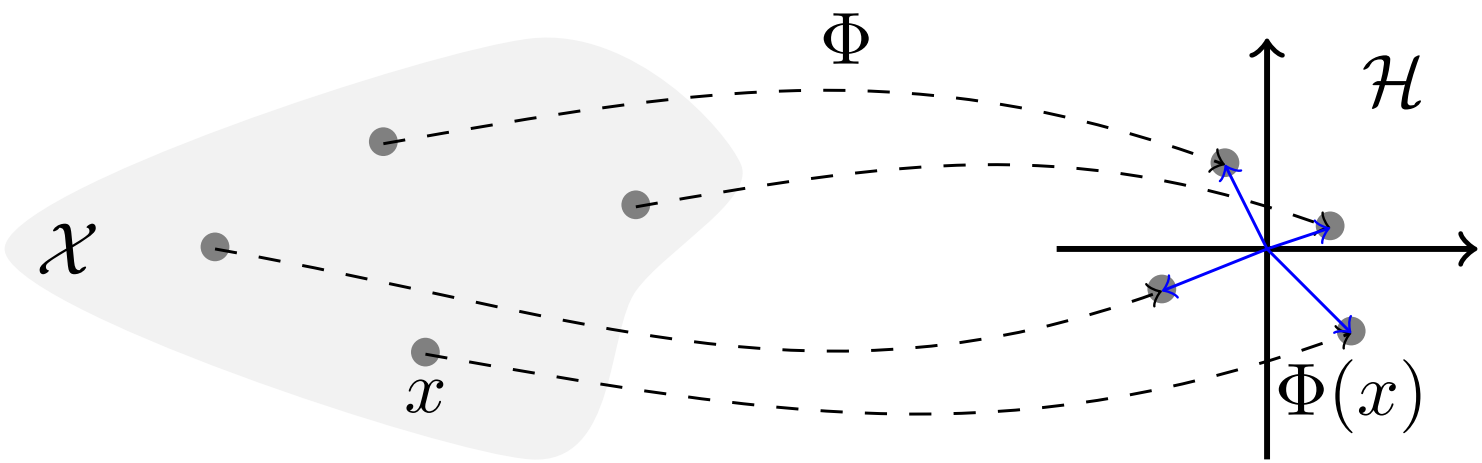
Theorem (Aronszajn, 1950)

K is a p.d. kernel on the set $\mathcal{X}$ if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping

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

such that, for any $\mathbf{x}, \mathbf{x}'$ in $\mathcal{X}$:

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \left\langle \Phi\left(\mathbf{x}\right), \Phi\left(\mathbf{x}'\right) \right\rangle_{\mathcal{H}} .$$



In case of ...

Definitions

- An inner product on an $\mathbb{R}$-vector space $\mathcal{H}$ is a mapping $(f, g) \mapsto \langle f, g \rangle_{\mathcal{H}}$ from $\mathcal{H}^2$ to $\mathbb{R}$ that is bilinear, symmetric and such that $\langle f, f \rangle_{\mathcal{H}} > 0$ for all $f \in \mathcal{H} \backslash \{0\}$.
- A vector space endowed with an inner product is called pre-Hilbert.

  It is endowed with a norm defined as $\| f \|_{\mathcal{H}} = \langle f, f \rangle_{\mathcal{H}}^{\frac{1}{2}}$.
- A Cauchy sequence $(f_n)_{n \geq 0}$ is a sequence whose elements become progressively arbitrarily close to each other:

$$\lim_{N \to +\infty} \sup_{n,m \geq N} \|f_n - f_m\|_{\mathcal{H}} = 0.$$

- A Hilbert space is a pre-Hilbert space complete for the norm $\|.\|_{\mathcal{H}}$. That is, any Cauchy sequence in $\mathcal{H}$ converges in $\mathcal{H}$.

Completeness is necessary to keep "good" convergence properties of Euclidean spaces in an infinite-dimensional context.

Proof: finite case

- Assume $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is finite of size N.
- Any p.d. kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is entirely defined by the $N \times N$ symmetric positive semidefinite matrix $[\mathbf{K}]_{ij} := K(\mathbf{x}_i, \mathbf{x}_j)$.
- It can therefore be diagonalized on an orthonormal basis of eigenvectors $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N)$, with non-negative eigenvalues $0 \leq \lambda_1 \leq \ldots \leq \lambda_N$, i.e.,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left[\sum_{l=1}^{N} \lambda_l \mathbf{u}_l \mathbf{u}_l^\top\right]_{ij} = \sum_{l=1}^{N} \lambda_l [\mathbf{u}_l]_i [\mathbf{u}_l]_j = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathbb{R}^N},$$

with

$$\Phi(\mathbf{x}_i) = \begin{pmatrix} \sqrt{\lambda_1}[\mathbf{u}_1]_i \\ \vdots \\ \sqrt{\lambda_N}[\mathbf{u}_N]_i \end{pmatrix}. \quad \square$$

Proof: general case

- Mercer (1909) for $\mathcal{X} = [a, b] \subset \mathbb{R}$ (more generally $\mathcal{X}$ compact) and K continuous.
- Kolmogorov (1941) for $\mathcal{X}$ countable.
- Aronszajn (1944, 1950) for the general case.

We will go through the proof of the general case by introducing the concept of Reproducing Kernel Hilbert Spaces (RKHS).

Outline

Functional spaces for machine learning

Before we go into formal details

- Among the Hilbert spaces $\mathcal{H}$ mentioned in Aronszjan's theorem, we will see that one of them, called RKHS, is of interest to us.
- This is a space of functions from $\mathcal{X}$ to $\mathbb{R}$.
- In other words, each data point $\mathbf{x}$ in $\mathcal{X}$ will be represented by a function $\Phi(\mathbf{x}) = K_{\mathbf{x}}$ in $\mathcal{H}$.

Functional spaces for machine learning

Before we go into formal details

- Among the Hilbert spaces $\mathcal{H}$ mentioned in Aronszjan's theorem, we will see that one of them, called RKHS, is of interest to us.
- This is a space of functions from $\mathcal{X}$ to $\mathbb{R}$.
- In other words, each data point $\mathbf{x}$ in $\mathcal{X}$ will be represented by a function $\Phi(\mathbf{x}) = K_{\mathbf{x}}$ in $\mathcal{H}$.

Example of functional mapping

- Consider $\mathcal{X} = \mathbb{R}$. We could decide to represent each scalar x in $\mathbb{R}$ as a Gaussian function centered at x:

$$K_x : y \mapsto e^{-\frac{1}{2\alpha}(x-y)^2}.$$

- What would be the corresponding $\mathcal{H}$ (if it exists)? What would be the inner-product?

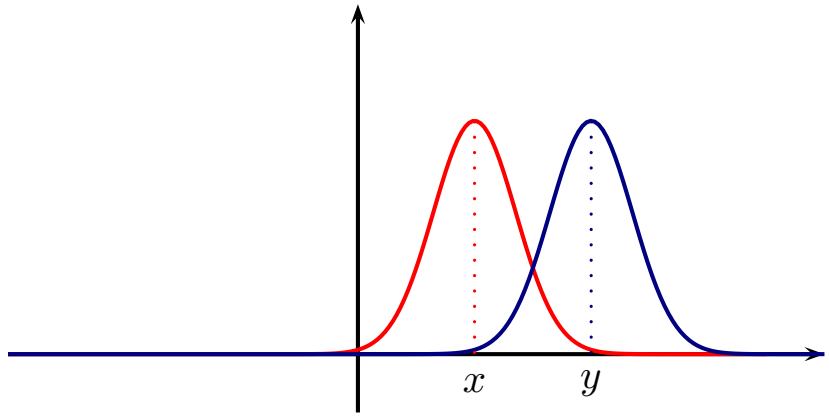
Functional spaces for machine learning

What does it mean to map a data point to a function?

Ex: if x, y in $\mathbb{R}$ and $K(x,y) = e^{-\frac{1}{\sigma^2}(x-y)^2}$ is the Gaussian kernel,

$$\Phi(x) : t \mapsto e^{-\frac{1}{2\alpha^2}(x-t)^2}$$

$$\Phi(y) : t \mapsto e^{-\frac{1}{2\alpha^2}(y-t)^2}$$



- Data points are mapped to Gaussian functions living in a Hilbert space $\mathcal{H}$.
- But $\mathcal{H}$ is much richer and contains much more than Gaussian functions!
- Prediction functions f live in $\mathcal{H}$: $f(x) = \langle f, \Phi(x) \rangle$.

RKHS Definition

Definition

Let $\mathcal{X}$ be a set and $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ be a class of functions forming a (real) Hilbert space with inner product $\langle ., . \rangle_{\mathcal{H}}$. The function $K : \mathcal{X}^2 \mapsto \mathbb{R}$ is called a reproducing kernel (r.k.) of $\mathcal{H}$ if

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

$$\forall \mathbf{x} \in \mathcal{X}, \quad K_{\mathbf{x}} : \mathbf{t} \mapsto K(\mathbf{x}, \mathbf{t}) .$$

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

$$f(\mathbf{x}) = \langle f, K_{\mathbf{x}} \rangle_{\mathcal{H}} .$$

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

RKHS: why do we care?

The principle of RKHS gives us a simple recipe to do machine learning:

- Map data $\mathbf{x}$ in $\mathcal{X}$ to a high-dimensional Hilbert space $\mathcal{H}$ (the RKHS) through a kernel mapping $\Phi : \mathcal{X} \to \mathcal{H}$, with $\Phi(\mathbf{x}) = K_{\mathbf{x}}$.
- In $\mathcal{H}$, consider simple linear models $f(\mathbf{x}) = \langle f, \Phi(\mathbf{x}) \rangle_{\mathcal{H}}$.
- If $\mathcal{X} = \mathbb{R}^p$, a linear function in $\Phi(\mathbf{x})$ may be nonlinear in $\mathbf{x}$.
- For instance, for supervised learning, given training data $(y_i, \mathbf{x}_i)_{i=1,\dots,n}$, we may want to minimize the empirical risk.

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

Positive Definite Kernels $\iff$ Reproducing Kernels

Theorem

A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is p.d. if and only if it is a r.k.

Proof

A r.k. is p.d.

1. A r.k. is symmetric because, for any $(\mathbf{x}, \mathbf{y}) \in \mathcal{X}^2$:

$$K(\mathbf{x}, \mathbf{y}) = \langle K_{\mathbf{x}}, K_{\mathbf{y}} \rangle_{\mathcal{H}} = \langle K_{\mathbf{y}}, K_{\mathbf{x}} \rangle_{\mathcal{H}} = K(\mathbf{y}, \mathbf{x}).$$

2. It is p.d. because for any $N \in \mathbb{N}$,$(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathcal{X}^N$, and $(a_1, a_2, \ldots, a_N) \in \mathbb{R}^N$:

$$\begin{aligned}\sum_{i,j=1}^{N} a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) &= \sum_{i,j=1}^{N} a_i a_j \left\langle K_{\mathbf{x}_i}, K_{\mathbf{x}_j} \right\rangle_{\mathcal{H}} \\ &= \| \sum_{i=1}^{N} a_i K_{\mathbf{x}_i} \|_{\mathcal{H}}^2 \\ &\geq 0. \quad \square\end{aligned}$$

Proof

A p.d. kernel is a r.k. (1/4)

- Let $\mathcal{H}_0$ be the vector subspace of $\mathbb{R}^{\mathcal{X}}$ spanned by the functions $\{K_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$.
- For any $f, g \in \mathcal{H}_0$, given by:

$$f = \sum_{i=1}^{m} a_i K_{\mathbf{x}_i}, \quad g = \sum_{j=1}^{n} b_j K_{\mathbf{y}_j},$$

let:

$$\langle f, g \rangle_{\mathcal{H}_0} := \sum_{i,j} a_i b_j K(\mathbf{x}_i, \mathbf{y}_j).$$

Proof

A p.d. kernel is a r.k. (2/4)

- $\langle f, g \rangle_{\mathcal{H}_0}$ does not depend on the expansion of f and g because:

$$\langle f, g \rangle_{\mathcal{H}_0} = \sum_{i=1}^{m} a_i g(\mathbf{x}_i) = \sum_{j=1}^{n} b_j f(\mathbf{y}_j).$$

- This also shows that $\langle ., . \rangle_{\mathcal{H}_0}$ is a symmetric bilinear form.
- This also shows that for any $\mathbf{x} \in \mathcal{X}$ and $f \in \mathcal{H}_0$:

$$\langle f, K_{\mathbf{x}} \rangle_{\mathcal{H}_0} = f(\mathbf{x}).$$

Proof

A p.d. kernel is a r.k. (3/4)

- K is assumed to be p.d., therefore:

$$\| f \|^2_{\mathcal{H}_0} = \sum_{i,j=1}^{m} a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \,.$$

  In particular Cauchy-Schwarz is valid with $\langle ., . \rangle_{\mathcal{H}_0}$.
- By Cauchy-Schwarz, we deduce that $\forall \mathbf{x} \in \mathcal{X}$:

$$| f(\mathbf{x}) | = \left| \langle f, K_{\mathbf{x}} \rangle_{\mathcal{H}_0} \right| \leq \| f \|_{\mathcal{H}_0} . K(\mathbf{x}, \mathbf{x})^{\frac{1}{2}} \,,$$

  therefore $\| f \|_{\mathcal{H}_0} = 0 \implies f = 0$.
- $\mathcal{H}_0$ is therefore a pre-Hilbert space endowed with the inner product $\langle ., . \rangle_{\mathcal{H}_0}$.

Proof

A p.d. kernel is a r.k. (4/4)

- For any Cauchy sequence $(f_n)_{n\geq 0}$ in $\left(\mathcal{H}_0, \langle ., . \rangle_{\mathcal{H}_0}\right)$, we note that:

$$\forall (\mathbf{x}, m, n) \in \mathcal{X} \times \mathbb{N}^2, \quad |f_m(\mathbf{x}) - f_n(\mathbf{x})| \leq \|f_m - f_n\|_{\mathcal{H}_0} . K(\mathbf{x}, \mathbf{x})^{\frac{1}{2}} .$$

  Therefore for any $\mathbf{x}$ the sequence $(f_n(\mathbf{x}))_{n\geq 0}$ is Cauchy in $\mathbb{R}$ and has therefore a limit.

- If we add to $\mathcal{H}_0$ the functions defined as the pointwise limits of Cauchy sequences, then the space becomes complete and is therefore a Hilbert space, with K as r.k. (up to a few technicalities, left as exercise). $\square$

Application: back to Aronzsajn's theorem

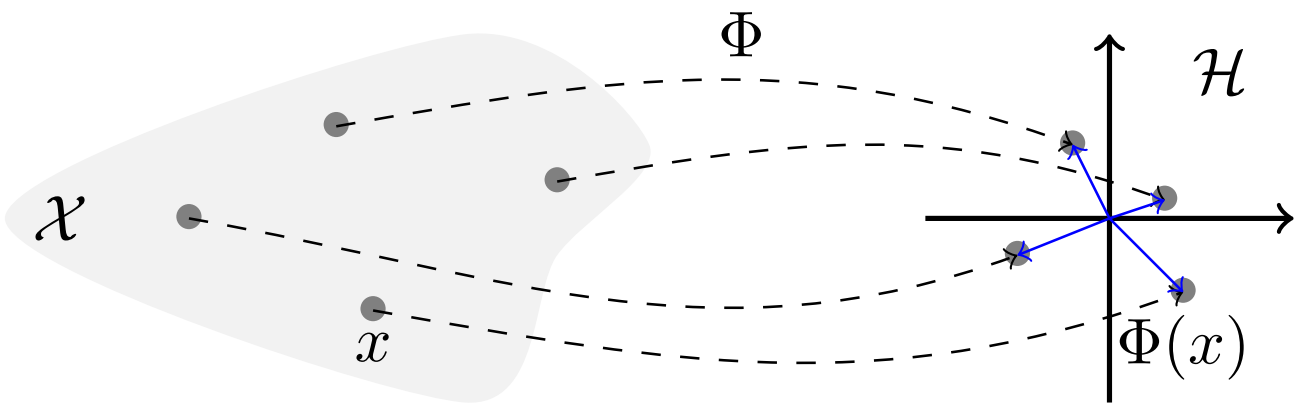
Theorem (Aronszajn, 1950)

K is a p.d. kernel on the set $\mathcal{X}$ if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping

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

such that, for any $\mathbf{x}, \mathbf{x}'$ in $\mathcal{X}$:

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \left\langle \Phi\left(\mathbf{x}\right), \Phi\left(\mathbf{x}'\right) \right\rangle_{\mathcal{H}} \ .$$



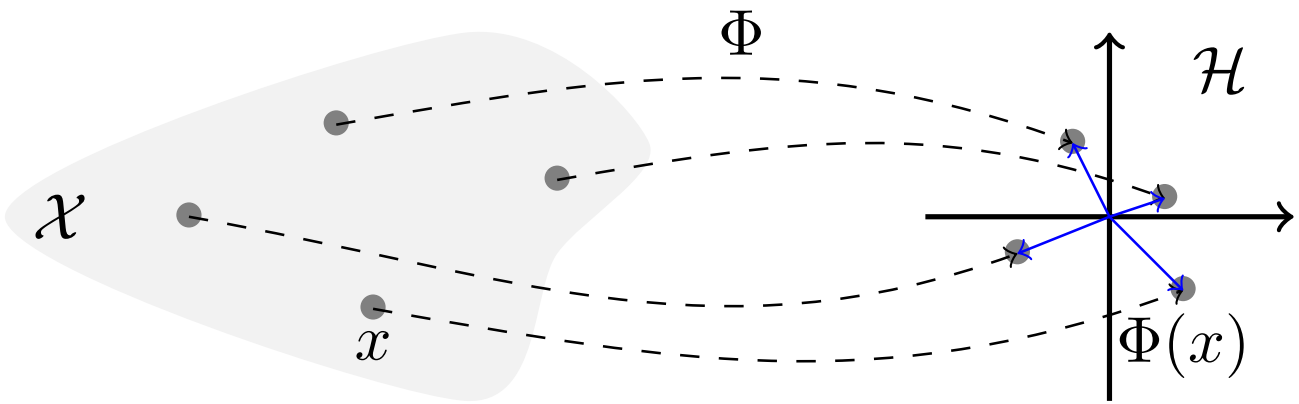
Proof of Aronzsajn's theorem

- If K is p.d. over a set $\mathcal{X}$ then it is the r.k. of a Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$.
- Let the mapping $\Phi : \mathcal{X} \to \mathcal{H}$ defined by:

$$\forall \mathbf{x} \in \mathcal{X}\,, \quad \Phi(\mathbf{x}) = K_{\mathbf{x}}\,.$$

- By the reproducing property we have:

$$\forall\,(\mathbf{x}, \mathbf{y}) \in \mathcal{X}^2, \quad \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle_{\mathcal{H}} = \langle K_{\mathbf{x}}, K_{\mathbf{y}} \rangle_{\mathcal{H}} = K\,(\mathbf{x}, \mathbf{y})\,. \quad \square$$



An equivalent definition of RKHS

Theorem

The Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ is a RKHS if and only if for any $\mathbf{x} \in \mathcal{X}$, the (linear) mapping:

$$\begin{aligned} F: \quad \mathcal{H} &\to \mathbb{R} \\ f &\mapsto f(\mathbf{x}) \end{aligned}$$

is continuous.

An equivalent definition of RKHS

Theorem

The Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ is a RKHS if and only if for any $\mathbf{x} \in \mathcal{X}$, the (linear) mapping:

$$\begin{aligned} F: \quad \mathcal{H} &\to \mathbb{R} \\ f &\mapsto f(\mathbf{x}) \end{aligned}$$

is continuous.

Corollary

Convergence in a RKHS implies pointwise convergence, i.e., if $(f_n)_{n\in\mathbb{N}}$ converges to f in $\mathcal{H}$, then $(f_n(\mathbf{x}))_{n\in\mathbb{N}}$ converges to $f(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{X}$.

Proof

If $\mathcal{H}$ is a RKHS then $f \mapsto f(\mathbf{x})$ is continuous

If a r.k. K exists, then for any $(\mathbf{x}, f) \in \mathcal{X} \times \mathcal{H}$:

$$\begin{aligned} |f(\mathbf{x})| &= |\langle f, K_{\mathbf{x}} \rangle_{\mathcal{H}}| \\ &\leq \|f\|_{\mathcal{H}} . \|K_{\mathbf{x}}\|_{\mathcal{H}} \qquad \text{(Cauchy-Schwarz)} \\ &\leq \|f\|_{\mathcal{H}} . K(\mathbf{x}, \mathbf{x})^{\frac{1}{2}} \ , \end{aligned}$$

because $\|K_{\mathbf{x}}\|_{\mathcal{H}}^2 = \langle K_{\mathbf{x}}, K_{\mathbf{x}} \rangle_{\mathcal{H}} = K(\mathbf{x}, \mathbf{x})$. Therefore $f \in \mathcal{H} \mapsto f(\mathbf{x}) \in \mathbb{R}$ is a continuous linear mapping. □

Since F is linear, it is indeed sufficient to show that $f \to 0 \Rightarrow f(x) \to 0$.

Proof (Converse)

If $f \mapsto f(\mathbf{x})$ is continuous then $\mathcal{H}$ is a RKHS

Conversely, let us assume that for any $\mathbf{x} \in \mathcal{X}$ the linear form $f \in \mathcal{H} \mapsto f(\mathbf{x})$ is continuous.
Then by Riesz representation theorem (general property of Hilbert spaces) there exists a unique $g_{\mathbf{x}} \in \mathcal{H}$ such that:

$$f(\mathbf{x}) = \langle f, g_{\mathbf{x}} \rangle_{\mathcal{H}} .$$

The function $K(\mathbf{x}, \mathbf{y}) = g_{\mathbf{x}}(\mathbf{y})$ is then a r.k. for $\mathcal{H}$. □

Uniqueness of r.k. and RKHS

Theorem

- If $\mathcal{H}$ is a RKHS, then it has a unique r.k.
- Conversely, a function K can be the r.k. of at most one RKHS.

Uniqueness of r.k. and RKHS

Theorem

- If $\mathcal{H}$ is a RKHS, then it has a unique r.k.
- Conversely, a function K can be the r.k. of at most one RKHS.

Consequence

This shows that we can talk of "the" kernel of a RKHS, or "the" RKHS of a kernel.

Outline

The linear kernel

Take $\mathcal{X} = \mathbb{R}^d$ and the linear kernel:

$$K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{R}^d} .$$

Theorem

The RKHS of the linear kernel is the set of linear functions of the form

$$f_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle_{\mathbb{R}^d} \quad \text{for} \quad \mathbf{w} \in \mathbb{R}^d ,$$

endowed with the inner product

$$\forall \mathbf{w}, \mathbf{v} \in \mathbb{R}^d, \quad \langle f_{\mathbf{w}}, f_{\mathbf{v}} \rangle_{\mathcal{H}} = \langle \mathbf{w}, \mathbf{v} \rangle_{\mathbb{R}^d}$$

and corresponding norm

$$\forall \mathbf{w} \in \mathbb{R}^d, \quad \| f_{\mathbf{w}} \|_{\mathcal{H}} = \| \mathbf{w} \|_2 .$$

Proof

The set $\mathcal{H}$ of functions described in the theorem is the dual of $\mathbb{R}^d$, hence it is a Hilbert space:

$$\mathcal{H} = \left\{ f_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle_{\mathbb{R}^d} : \mathbf{w} \in \mathbb{R}^d \right\}.$$

- $\mathcal{H}$ contains all functions of the form $K_{\mathbf{w}} : \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle_{\mathbb{R}^d}$.
- For every $\mathbf{x}$ in $\mathbb{R}^d$, and $f_{\mathbf{w}}$ in $\mathcal{H}$,

$$f_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle_{\mathbb{R}^d} = \langle f_{\mathbf{w}}, K_{\mathbf{x}} \rangle_{\mathcal{H}}.$$

$\mathcal{H}$ is thus the RKHS of the linear kernel.

The polynomial kernel

Let us find the RKHS of the polynomial kernel of degree 2:

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \quad K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{R}^d}^2 = \left(\mathbf{x}^\top \mathbf{y} \right)^2$$

The polynomial kernel

Let us find the RKHS of the polynomial kernel of degree 2:

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \quad K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{R}^d}^2 = \left(\mathbf{x}^\top \mathbf{y} \right)^2$$

First step: Look for an inner-product.

$$\begin{aligned} K(\mathbf{x}, \mathbf{y}) &= \operatorname{trace}\left(\mathbf{x}^\top \mathbf{y} \, \mathbf{x}^\top \mathbf{y} \right) \\ &= \operatorname{trace}\left(\mathbf{y}^\top \mathbf{x} \, \mathbf{x}^\top \mathbf{y} \right) \\ &= \operatorname{trace}\left(\mathbf{x}\mathbf{x}^\top \mathbf{y}\mathbf{y}^\top \right) \\ &= \left\langle \mathbf{x}\mathbf{x}^\top, \mathbf{y}\mathbf{y}^\top \right\rangle_{\mathsf{F}}, \end{aligned}$$

where F is the Froebenius norm for matrices in $\mathbb{R}^{d \times d}$. Note that we have proven here that K is p.d.

The polynomial kernel

Second step: propose a candidate RKHS.

We know that $\mathcal{H}$ contains all the functions

$$f(\mathbf{x}) = \sum_i a_i K(\mathbf{x}_i, \mathbf{x}) = \sum_i a_i \left\langle \mathbf{x}_i \mathbf{x}_i^\top, \mathbf{x}\mathbf{x}^\top \right\rangle_{\mathsf{F}} = \left\langle \sum_i a_i \mathbf{x}_i \mathbf{x}_i^\top, \mathbf{x}\mathbf{x}^\top \right\rangle_{\mathsf{F}}.$$

Any symmetric matrix in $\mathbb{R}^{d\times d}$ may be decomposed as $\sum_i a_i \mathbf{x}_i \mathbf{x}_i^\top$. Our candidate RKHS $\mathcal{H}$ will be the set of quadratic functions

$$f_{\mathbf{S}}(\mathbf{x}) = \left\langle \mathbf{S}, \mathbf{x}\mathbf{x}^\top \right\rangle_{\mathsf{F}} = \mathbf{x}^\top \mathbf{S}\mathbf{x} \quad \text{for} \quad \mathbf{S} \in \mathcal{S}^{d\times d},$$

where $\mathcal{S}^{d\times d}$ is the set of symmetric[1] matrices in $\mathbb{R}^{d\times d}$, endowed with the inner-product $\langle f_{\mathbf{S}_1}, f_{\mathbf{S}_2} \rangle_{\mathcal{H}} = \langle \mathbf{S}_1, \mathbf{S}_2 \rangle_{\mathsf{F}}$.

[1] Why is it important?

The polynomial kernel

Third step: check that the candidate is a Hilbert space.

This step is trivial in the present case since it is easy to see that $\mathcal{H}$ a Euclidean space, isomorphic to $\mathcal{S}^{d\times d}$ by $\Phi : \mathbf{S} \mapsto f_{\mathbf{S}}$. Sometimes, things are not so simple and we need to prove the completeness explicitly.

Fourth step: check that $\mathcal{H}$ is the RKHS.

1. $\mathcal{H}$ contains all the functions $K_{\mathbf{x}} : \mathbf{t} \mapsto K(\mathbf{x}, \mathbf{t}) = \left\langle \mathbf{x}\mathbf{x}^\top, \mathbf{t}\mathbf{t}^\top \right\rangle_{\mathsf{F}}$.
2. For all $f_{\mathbf{S}}$ in $\mathcal{H}$ and $\mathbf{x}$ in $\mathcal{X}$,

$$f_{\mathbf{S}}(\mathbf{x}) = \left\langle \mathbf{S}, \mathbf{x}\mathbf{x}^\top \right\rangle_{\mathsf{F}} = \langle f_{\mathbf{S}}, f_{\mathbf{x}\mathbf{x}^\top} \rangle_{\mathcal{H}} = \langle f_{\mathbf{S}}, K_{\mathbf{x}} \rangle_{\mathcal{H}} \ .$$

□

Remark

All points $\mathbf{x}$ in $\mathcal{X}$ are mapped to a rank-one matrix $\mathbf{x}\mathbf{x}^\top$, hence to a function $K_{\mathbf{x}} = f_{\mathbf{x}\mathbf{x}^\top}$ in $\mathcal{H}$. However, most of points in $\mathcal{H}$ do not admit a pre-image (why?).

Exercise: what is the RKHS of the general polynomial kernel?

Combining kernels

Theorem

- If K_1 and K_2 are p.d. kernels, then:

$$K_1 + K_2,$$
$$K_1 K_2, \text{ and}$$
$$cK_1, \text{ for } c \geq 0,$$

  are also p.d. kernels

- If $(K_i)_{i \geq 1}$ is a sequence of p.d. kernels that converges pointwisely to a function K:

$$\forall \left(\mathbf{x}, \mathbf{x}'\right) \in \mathcal{X}^2, \quad K\left(\mathbf{x}, \mathbf{x}'\right) = \lim_{n \to \infty} K_i\left(\mathbf{x}, \mathbf{x}'\right),$$

  then K is also a p.d. kernel.

Proof: for $K_1 K_2$, see next slide; otherwise, left as exercise

Proof for $K_1 K_2$ is p.d.

Proof.

Consider n points in $\mathcal{X}$ and the corresponding $n \times n$ p.s.d. kernel matrices $\mathbf{K}_1$ and $\mathbf{K}_2$. As p.s.d. matrices, they admit factorizations $\mathbf{K}_1 = \mathbf{X}^\top \mathbf{X}$ and $\mathbf{K}_2 = \mathbf{Y}^\top \mathbf{Y}$. Then,

$$\begin{aligned}
[\mathbf{K}]_{ij} &= [\mathbf{K}_1]_{ij} [\mathbf{K}_2]_{ij} \\
&= \text{trace}\left((\mathbf{x}_i^\top \mathbf{x}_j)(\mathbf{y}_j^\top \mathbf{y}_i) \right) \\
&= \text{trace}\left((\mathbf{y}_i \mathbf{x}_i^\top)(\mathbf{x}_j \mathbf{y}_j^\top) \right) \\
&= \left\langle \mathbf{x}_i \mathbf{y}_i^\top, \mathbf{x}_j \mathbf{y}_j^\top \right\rangle_{\mathsf{F}}. \\
&= \langle \mathbf{z}_i, \mathbf{z}_j \rangle_{\mathbb{R}^{n^2}},
\end{aligned}$$

where the $\mathbf{x}_i$'s and the $\mathbf{y}_i$'s are the columns of $\mathbf{X}$ and $\mathbf{Y}$, respectively and $\mathbf{z}_i = \text{vec}(\mathbf{x}_i \mathbf{y}_i^\top)$. Thus, $\mathbf{K}$ is p.s.d. and $K = K_1 K_2$ is a p.d. kernel. □

Examples

Theorem

If K is a kernel, then e^K is a kernel too.

Examples

Theorem

If K is a kernel, then e^K is a kernel too.

Proof:

$$e^{K(\mathbf{x},\mathbf{x}')} = \lim_{n \to +\infty} \sum_{i=0}^{n} \frac{K(\mathbf{x},\mathbf{x}')^i}{i!}$$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \log(1 + xx')$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \log(1 + xx')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \exp\left(-|x - x'|^2\right)$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \log(1 + xx')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \exp\left(-|x - x'|^2\right)$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x + x')$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \log(1 + xx')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \exp\left(-|x - x'|^2\right)$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x + x')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x - x')$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \log(1 + xx')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \exp\left(-|x - x'|^2\right)$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x + x')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x - x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \min(x, x')$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \log(1 + xx')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \exp\left(-|x - x'|^2\right)$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x + x')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x - x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \min(x, x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \max(x, x')$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \log(1 + xx')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \exp\left(-|x - x'|^2\right)$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x + x')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x - x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \min(x, x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \max(x, x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \min(x, x') / \max(x, x')$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \log(1 + xx')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \exp\left(-|x - x'|^2\right)$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x + x')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x - x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \min(x, x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \max(x, x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \min(x, x') / \max(x, x')$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = GCD(x, x')$

Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \log(1 + xx')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \exp\left(-|x - x'|^2\right)$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x + x')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x - x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \min(x, x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \max(x, x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \min(x, x') / \max(x, x')$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = GCD(x, x')$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = LCM(x, x')$

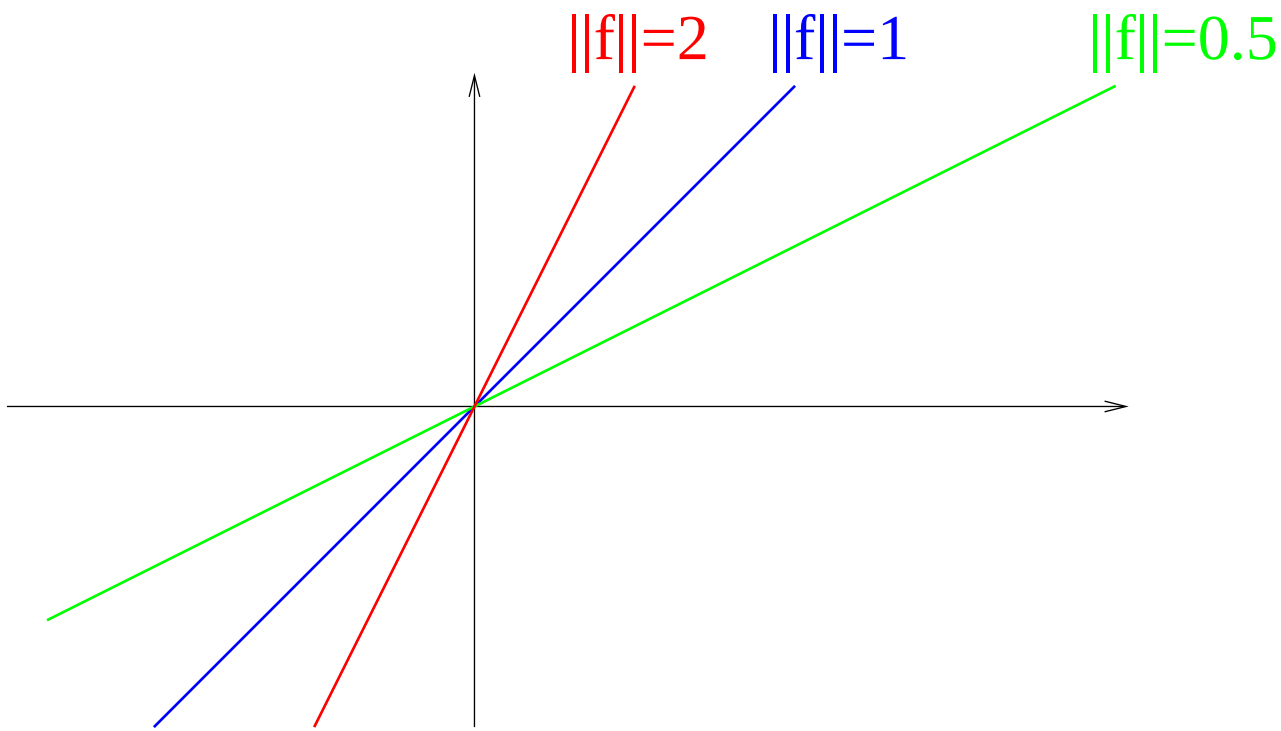
Quizz : which of the following are p.d. kernels?

- $\mathcal{X} = (-1, 1), \quad K(x, x') = \frac{1}{1-xx'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{x+x'}$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = 2^{xx'}$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \log(1 + xx')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \exp\left(-|x - x'|^2\right)$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x + x')$
- $\mathcal{X} = \mathbb{R}, \quad K(x, x') = \cos(x - x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \min(x, x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \max(x, x')$
- $\mathcal{X} = \mathbb{R}_+, \quad K(x, x') = \min(x, x') / \max(x, x')$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = GCD(x, x')$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = LCM(x, x')$
- $\mathcal{X} = \mathbb{N}, \quad K(x, x') = GCD(x, x') / LCM(x, x')$

Outline

Remember the RKHS of the linear kernel

$$\begin{cases} K_{lin}(\mathbf{x}, \mathbf{x}') & = \mathbf{x}^\top \mathbf{x}' \, . \\ f(\mathbf{x}) & = \mathbf{w}^\top \mathbf{x} \, , \\ \| f \|_{\mathcal{H}} & = \| \mathbf{w} \|_2 \, . \end{cases}$$



Smoothness functional

A simple inequality

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

$$\begin{aligned} \left| f(\mathbf{x}) - f(\mathbf{x}') \right| &= \left| \langle f, K_{\mathbf{x}} - K_{\mathbf{x}'} \rangle_{\mathcal{H}} \right| \\ &\leq \| f \|_{\mathcal{H}} \times \| K_{\mathbf{x}} - K_{\mathbf{x}'} \|_{\mathcal{H}} \\ &= \| f \|_{\mathcal{H}} \times d_K(\mathbf{x}, \mathbf{x}') \ . \end{aligned}$$

- The norm of a function in the RKHS controls how fast the function varies over $\mathcal{X}$ with respect to the geometry defined by the kernel (Lipschitz with constant $\| f \|_{\mathcal{H}}$).

Important message

Small norm $\Longrightarrow$ slow variations.

Kernels and RKHS: Summary

- P.d. kernels can be thought of as inner product after embedding the data space $\mathcal{X}$ in some Hilbert space. As such a p.d. kernel defines a metric on $\mathcal{X}$.
- A realization of this embedding is the RKHS, valid without restriction on the space $\mathcal{X}$ nor on the kernel.
- The RKHS is a space of functions over $\mathcal{X}$. The norm of a function in the RKHS is related to its degree of smoothness w.r.t. the metric defined by the kernel on $\mathcal{X}$.
- We will now see some applications of kernels and RKHS in statistics, before coming back to the problem of choosing (and eventually designing) the kernel.

Kernel tricks

Motivations

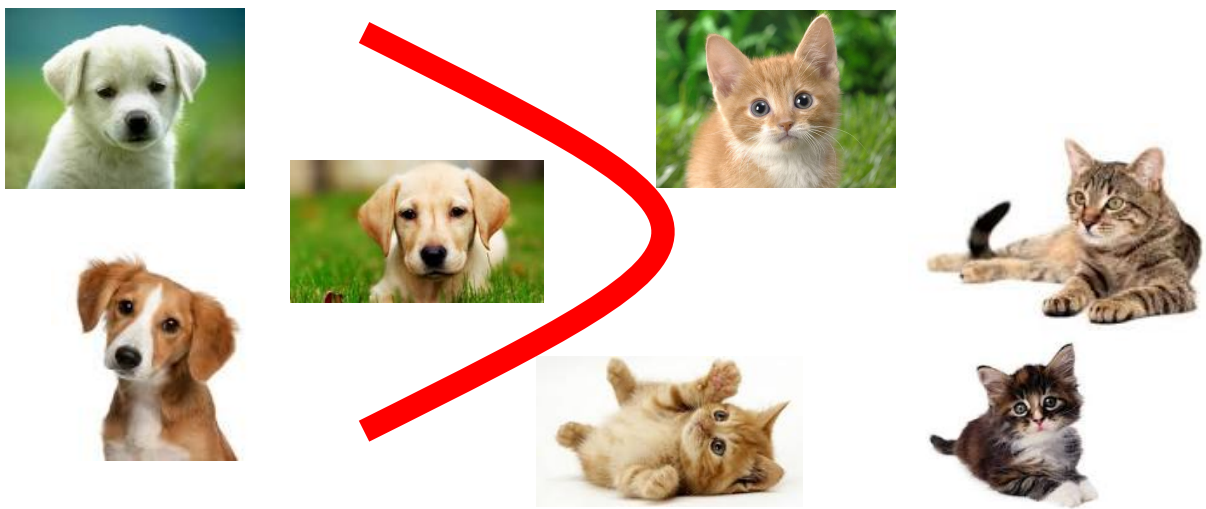
Two theoretical results underpin a family of powerful algorithms for data analysis using p.d. kernels, collectively known as kernel methods:

- The kernel trick, based on the representation of p.d. kernels as inner products;
- The representer theorem, based on some properties of the regularization functional defined by the RKHS norm.

Motivation from supervised learning

For instance, in supervised learning, the goal is to learn a **prediction function** $f : \mathcal{X} \to \mathcal{Y}$ given labeled training data $(\mathbf{x}_i, y_i)_{i=1,\dots,n}$ with $\mathbf{x}_i$ in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}} .$$



(Vapnik, 1995)...

Motivation from supervised learning

For instance, in supervised learning, the goal is to learn a **prediction function** $f : \mathcal{X} \to \mathcal{Y}$ given labeled training data $(\mathbf{x}_i, y_i)_{i=1,\dots,n}$ with $\mathbf{x}_i$ in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}} .$$

The labels y_i are, for instance, in

- $\{-1, +1\}$ for **binary** classification problems.
- $\{1, \dots, K\}$ for **multi-class** classification problems.
- $\mathbb{R}$ for **regression** problems.
- $\mathbb{R}^k$ for **multivariate regression** problems.

Motivation from supervised learning

For instance, in supervised learning, the goal is to learn a **prediction function** $f : \mathcal{X} \to \mathcal{Y}$ given labeled training data $(\mathbf{x}_i, y_i)_{i=1,\dots,n}$ with $\mathbf{x}_i$ in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}} .$$

Example with linear models: logistic regression, *etc.*

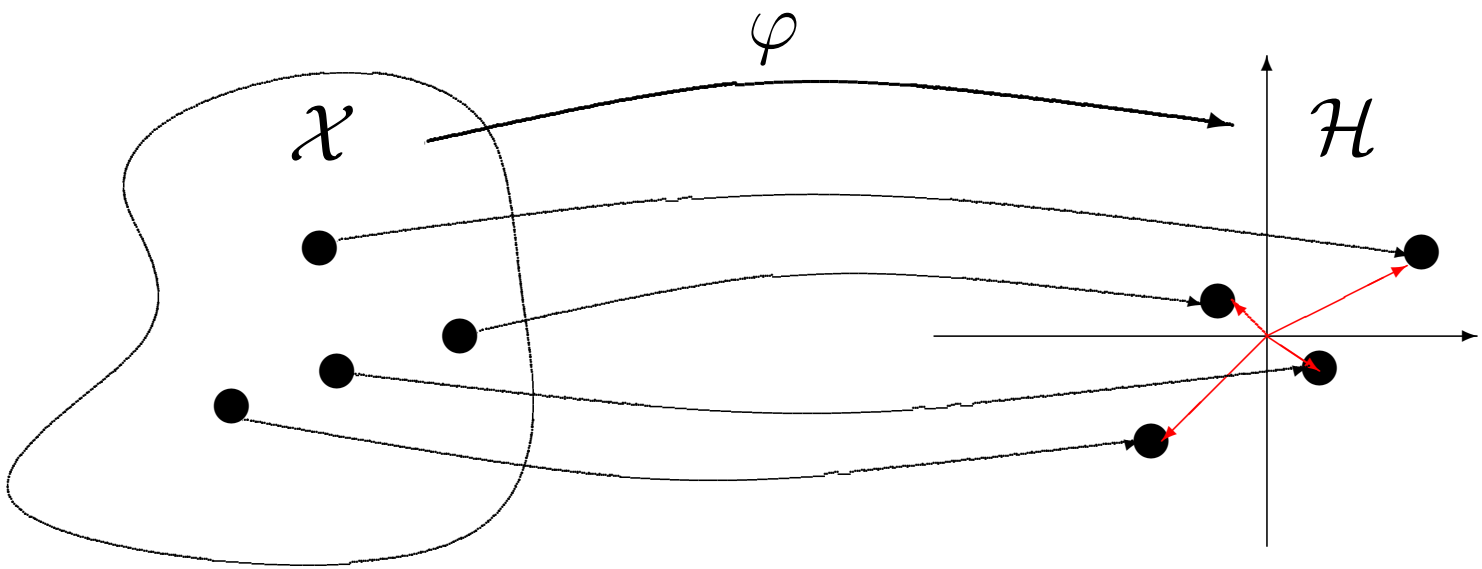
- assume there exists a linear relation between y and features $\mathbf{x}$ in $\mathbb{R}^p$.
- $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ is parametrized by $\mathbf{w}, b$ in $\mathbb{R}^{p+1}$;
- L is often a **convex** loss function;
- $\Omega(f)$ is often the squared ℓ_2-norm $\|\mathbf{w}\|^2$.

Motivation from supervised learning

$$\min_{f \in \mathcal{H}} \quad \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) \; + \; \lambda \|f\|_{\mathcal{H}}^2.$$

- Kernel methods allow you to **map** data x in $\mathcal{X}$ to a Hilbert space and work with **linear forms**:

$$\Phi : \mathcal{X} \to \mathcal{H} \qquad \text{and} \qquad f(\mathbf{x}) = \langle \Phi(\mathbf{x}), f \rangle_{\mathcal{H}}.$$



Motivation from supervised learning

$$\min_{f \in \mathcal{H}} \quad \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) \; + \; \lambda \|f\|_{\mathcal{H}}^2.$$

First purpose: embed data in a vectorial space where

- many **geometrical operations** exist (angle computation, projection on linear subspaces, definition of barycenters....).
- one may learn potentially **rich infinite-dimensional models**.
- **regularization** is natural and theoretically grounded.

Motivation from supervised learning

$$\min_{f \in \mathcal{H}} \quad \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) \; + \; \lambda \|f\|_{\mathcal{H}}^2.$$

First purpose: embed data in a vectorial space where

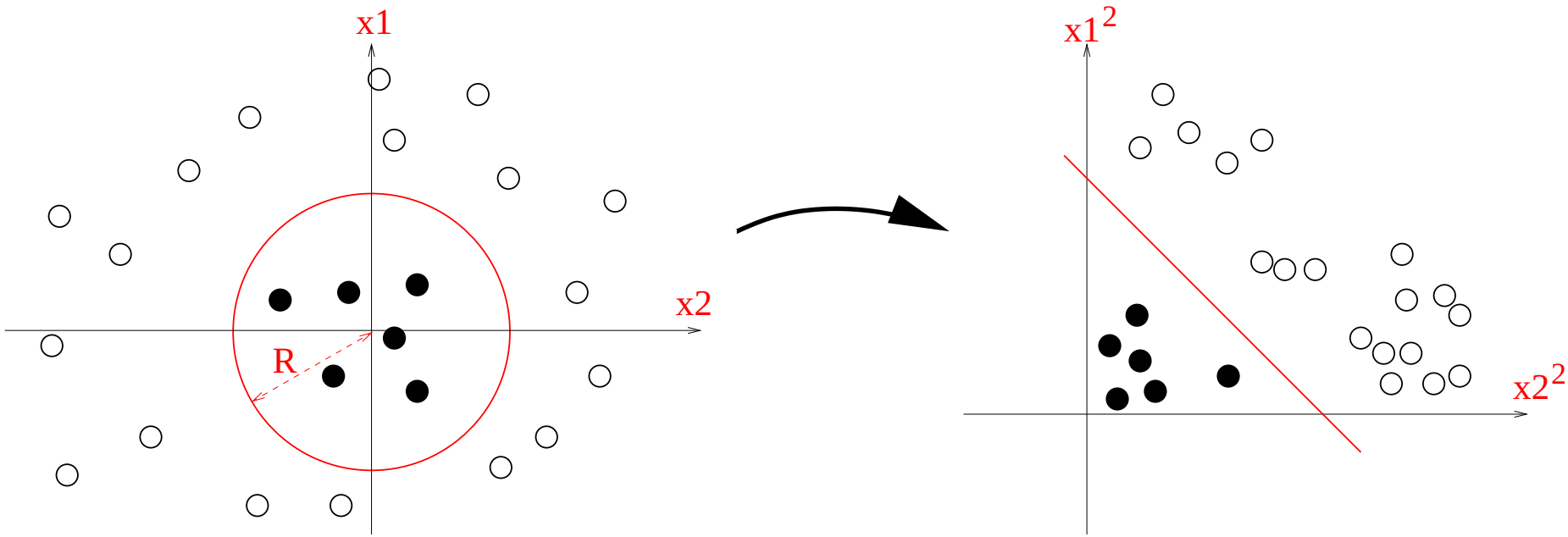
- many **geometrical operations** exist (angle computation, projection on linear subspaces, definition of barycenters....).
- one may learn potentially **rich infinite-dimensional models**.
- **regularization** is natural and theoretically grounded.

The principle is **generic** and does not assume anything about the nature of the set $\mathcal{X}$ (vectors, sets, graphs, sequences).

Motivation from supervised learning

Second purpose: unhappy with the current Euclidean structure?

- lift data to a higher-dimensional space with **nicer properties** (e.g., linear separability, clustering structure).
- then, the **linear** form $f(\mathbf{x}) = \langle \Phi(\mathbf{x}), f \rangle_{\mathcal{H}}$ in $\mathcal{H}$ may correspond to a **non-linear** model in $\mathcal{X}$.



Outline

The kernel trick

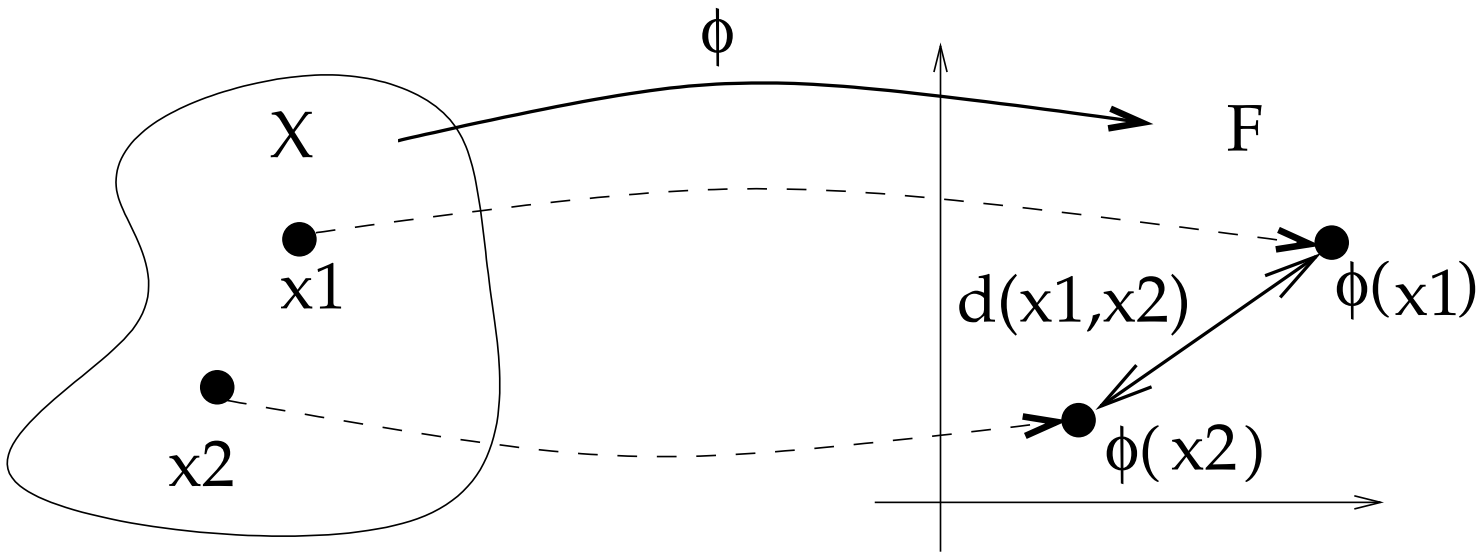
Proposition

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 feature space of a p.d. kernel by replacing each inner product evaluation by a kernel evaluation.

Remarks:

- The proof of this proposition is trivial, because the kernel is exactly the inner product in the feature space.
- This trick has huge practical applications.
- Vectors in the feature space are only manipulated implicitly, through pairwise inner products.

Example 1: computing distances in the feature space



$$
\begin{aligned}
d_K(\mathbf{x}_1, \mathbf{x}_2)^2 &= \| \Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) \|^2_{\mathcal{H}} \\
&= \langle \Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) \rangle_{\mathcal{H}} \\
&= \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_1) \rangle_{\mathcal{H}} + \langle \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_2) \rangle_{\mathcal{H}} - 2 \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle_{\mathcal{H}}
\end{aligned}
$$

$$
d_K(\mathbf{x}_1, \mathbf{x}_2)^2 = K(\mathbf{x}_1, \mathbf{x}_1) + K(\mathbf{x}_2, \mathbf{x}_2) - 2K(\mathbf{x}_1, \mathbf{x}_2)
$$

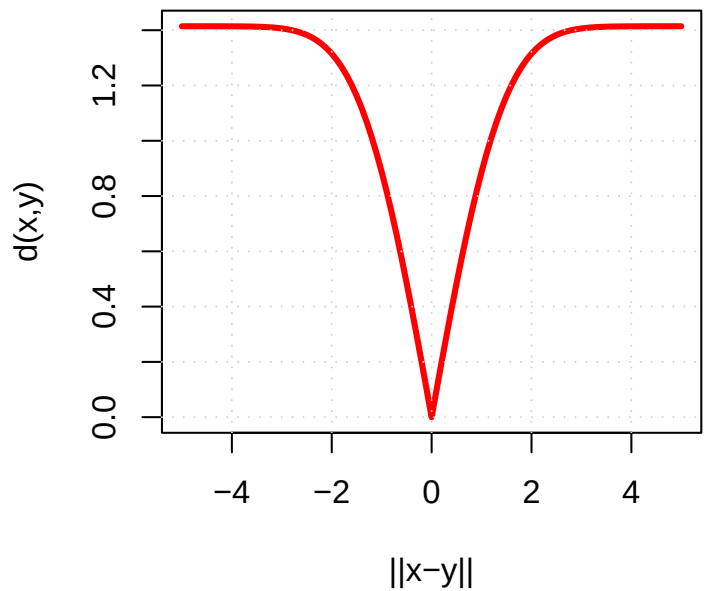
Distance for the Gaussian kernel

- The Gaussian kernel with bandwidth σ on $\mathbb{R}^d$ is:

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

- $K(\mathbf{x}, \mathbf{x}) = 1 = \| \Phi(\mathbf{x}) \|^2_{\mathcal{H}}$, so all points are on the unit sphere in the feature space.
- The distance between the images of two points $\mathbf{x}$ and $\mathbf{y}$ in the feature space is given by:

$$d_K(\mathbf{x}, \mathbf{y}) = \sqrt{2\left[1 - e^{-\frac{\| \mathbf{x}-\mathbf{y} \|^2}{2\sigma^2}}\right]}$$



Example 2: distance between a point and a set

Problem

- Let $\mathcal{S} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ be a finite set of points in $\mathcal{X}$.
- How to define and compute the similarity between any point $\mathbf{x}$ in $\mathcal{X}$ and the set $\mathcal{S}$?

Example 2: distance between a point and a set

Problem

- Let $\mathcal{S} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ be a finite set of points in $\mathcal{X}$.
- How to define and compute the similarity between any point $\mathbf{x}$ in $\mathcal{X}$ and the set $\mathcal{S}$?

A solution:

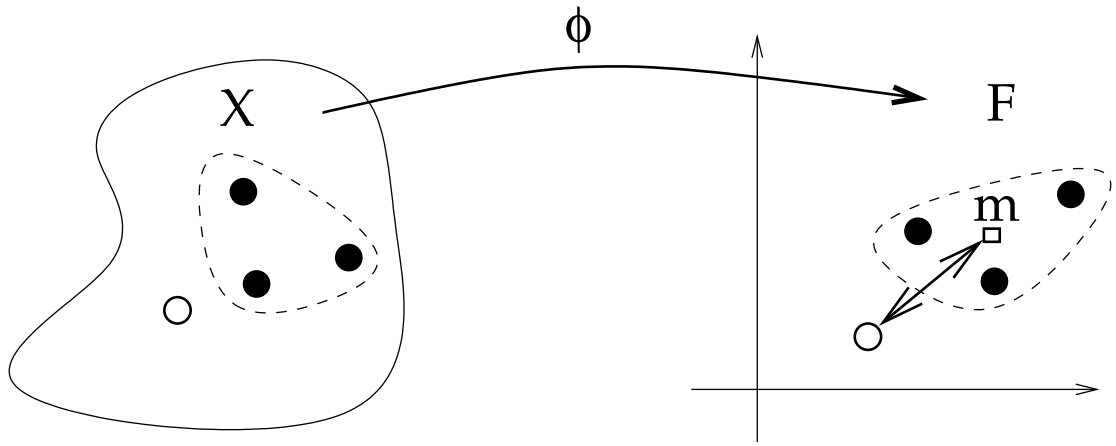
- Map all points to the feature space.
- Summarize $\mathcal{S}$ by the barycenter of the points:

$$\boldsymbol{\mu} := \frac{1}{n} \sum_{i=1}^{n} \Phi(\mathbf{x}_i) \ .$$

- Define the distance between $\mathbf{x}$ and $\mathcal{S}$ by:

$$d_K(\mathbf{x}, \mathcal{S}) := \| \Phi(\mathbf{x}) - \boldsymbol{\mu} \|_{\mathcal{H}} \ .$$

Computation



$$d_K(\mathbf{x}, \mathcal{S}) = \left\| \Phi(\mathbf{x}) - \frac{1}{n} \sum_{i=1}^{n} \Phi(\mathbf{x}_i) \right\|_{\mathcal{H}}$$

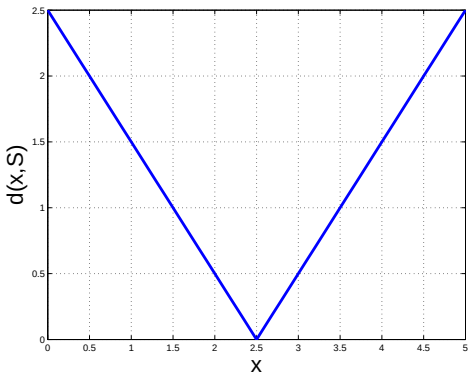
$$= \sqrt{K(\mathbf{x}, \mathbf{x}) - \frac{2}{n} \sum_{i=1}^{n} K(\mathbf{x}, \mathbf{x}_i) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} K(\mathbf{x}_i, \mathbf{x}_j)}.$$

Remark

The barycentre $\boldsymbol{\mu}$ only exists in the feature space in general: it does not necessarily have a pre-image $\mathbf{x}_{\boldsymbol{\mu}}$ such that $\Phi\left(\mathbf{x}_{\boldsymbol{\mu}}\right) = \boldsymbol{\mu}$.

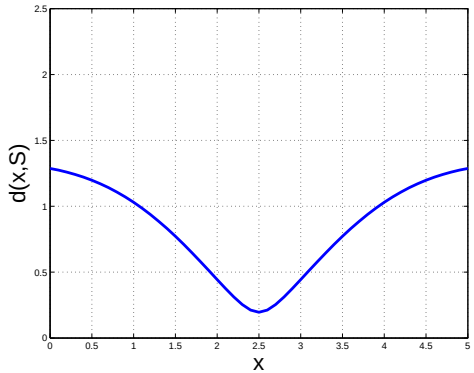
1D illustration

- $\mathcal{S} = \{2, 3\}$
- Plot $f(x) = d(x, \mathcal{S})$



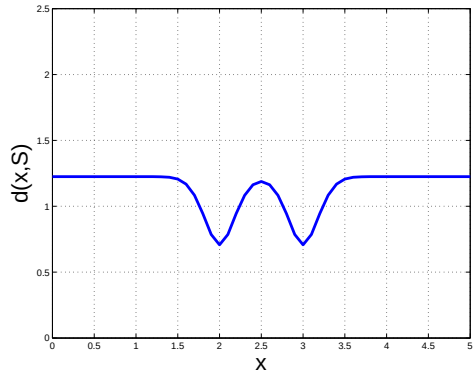
$K(x, y) = xy.$

(linear)



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

with $\sigma = 1.$

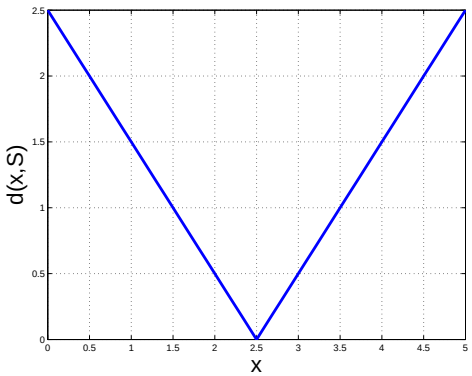


$K(x, y) = e^{-\frac{(x-y)^2}{2\sigma^2}}.$

with $\sigma = 0.2.$

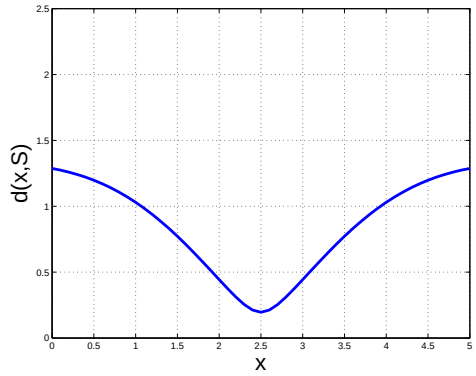
1D illustration

- $\mathcal{S} = \{2, 3\}$
- Plot $f(x) = d(x, \mathcal{S})$



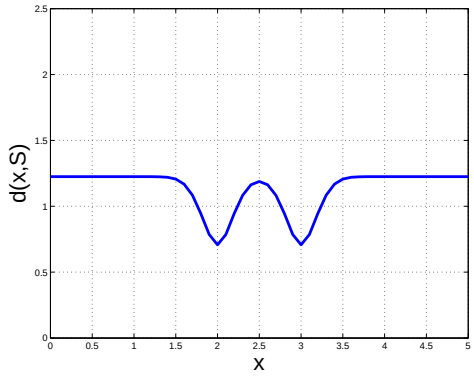
$K(x, y) = xy.$

(linear)



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

with $\sigma = 1$.



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

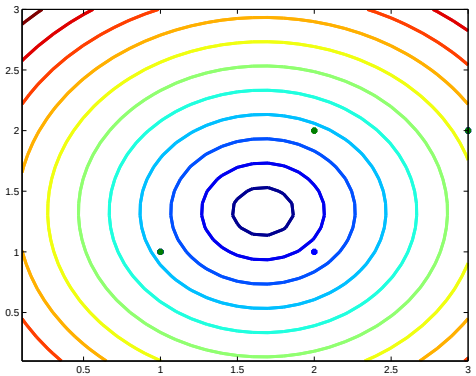
with $\sigma = 0.2$.

Remarks

- for the linear kernel, $\mathcal{H} = \mathbb{R}$, $\mu = 2.5$ and $d(x, \mathcal{S}) = |x - \mu|$.
- for the Gaussian kernel $d(x, \mathcal{S}) = \sqrt{C - \frac{2}{n} \sum_{i=1}^{n} K(x_i, x)}$.

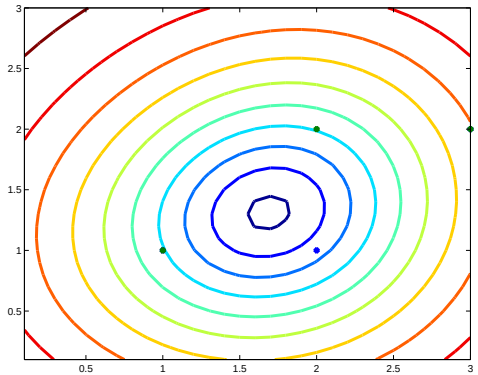
2D illustration

- $\mathcal{S} = \{(1,1)', (1,2)', (2,2)'\}$
- Plot $f(\mathbf{x}) = d(\mathbf{x}, \mathcal{S})$



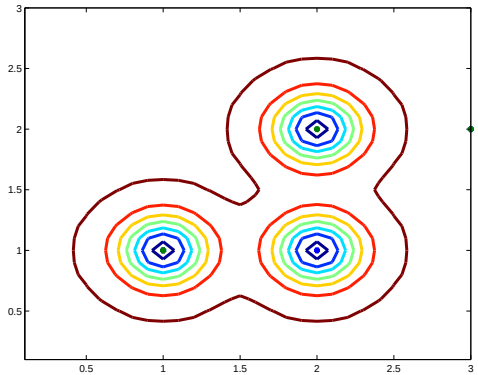
$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}\mathbf{y}$.

(linear)



$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}$.

with $\sigma = 1$.

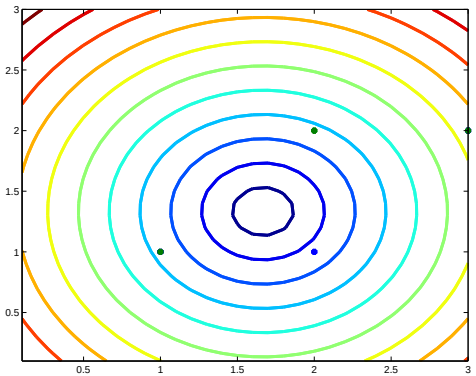


$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}$.

with $\sigma = 0.2$.

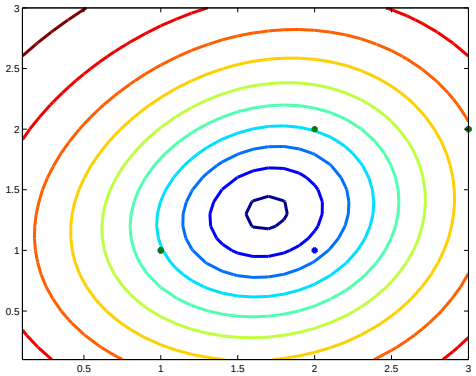
2D illustration

- $\mathcal{S} = \{(1,1)', (1,2)', (2,2)'\}$
- Plot $f(\mathbf{x}) = d(\mathbf{x}, \mathcal{S})$



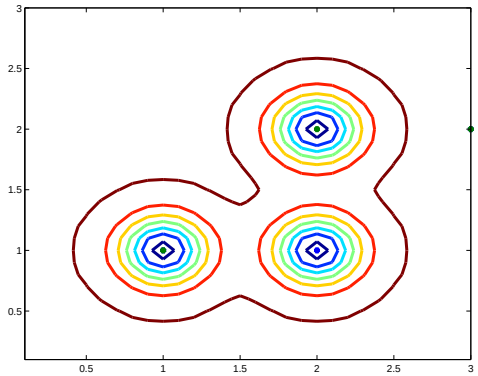
$K(\mathbf{x}, \mathbf{y}) = \mathbf{xy}.$

(linear)



$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$

with $\sigma = 1$.



$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$

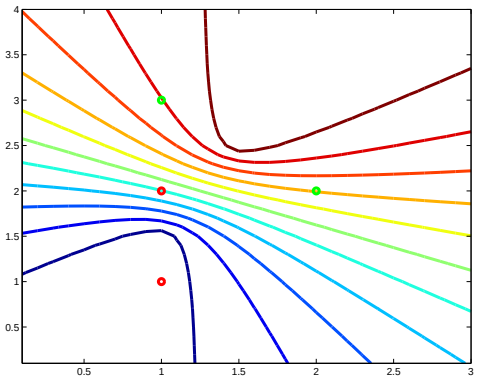
with $\sigma = 0.2$.

Remark

- as before, the barycenter μ in $\mathcal{H}$ (which is a single point in $\mathcal{H}$) may carry a lot of information about the training data.

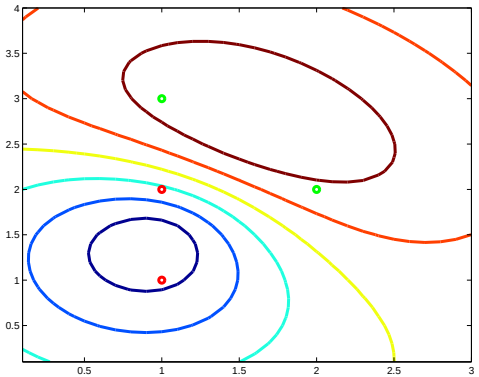
Basic application in discrimination

- $\mathcal{S}_1 = \{(1,1)', (1,2)'\}$ and $\mathcal{S}_2 = \{(1,3)', (2,2)'\}$
- Plot $f(\mathbf{x}) = d(\mathbf{x}, \mathcal{S}_1)^2 - d(\mathbf{x}, \mathcal{S}_2)^2$



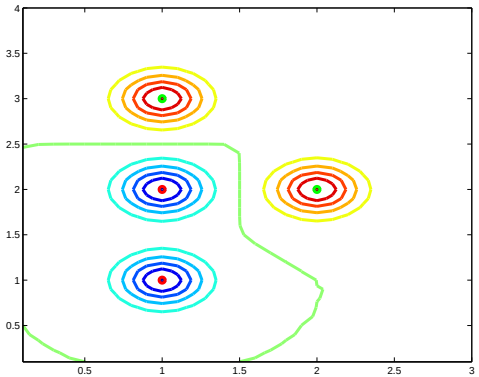
$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}\mathbf{y}$.

(linear)



$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}$.

with $\sigma = 1$.



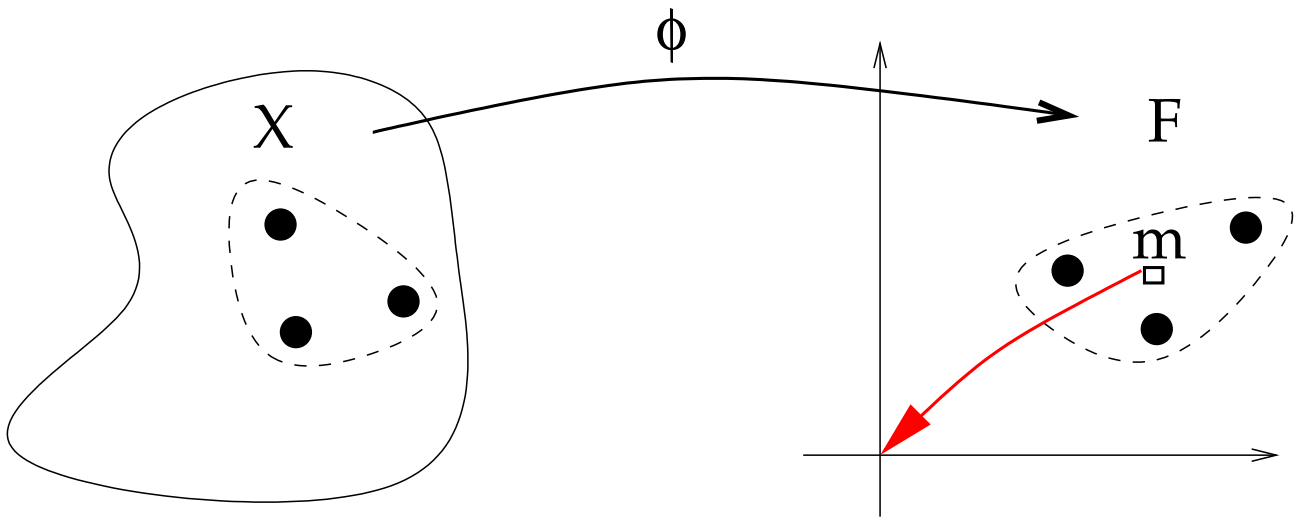
$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}$.

with $\sigma = 0.2$.

Example 3: Centering data in the feature space

Problem

- Let $\mathcal{S} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ be a finite set of points in $\mathcal{X}$ endowed with a p.d. kernel K. Let $\mathbf{K}$ be their $n \times n$ Gram matrix: $[\mathbf{K}]_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.
- Let $\boldsymbol{\mu} = 1/n \sum_{i=1}^{n} \Phi(\mathbf{x}_i)$ their barycenter, and $\mathbf{u}_i = \Phi(\mathbf{x}_i) - \boldsymbol{\mu}$ for $i = 1, \ldots, n$ be centered data in $\mathcal{H}$.
- How to compute the centered Gram matrix $[\mathbf{K}^c]_{i,j} = \langle \mathbf{u}_i, \mathbf{u}_j \rangle_{\mathcal{H}}$?



Computation

- A direct computation gives, for $0 \leq i, j \leq n$:

$$\begin{aligned}
\mathbf{K}^c_{i,j} &= \langle \Phi(\mathbf{x}_i) - \boldsymbol{\mu}, \Phi(\mathbf{x}_j) - \boldsymbol{\mu} \rangle_{\mathcal{H}} \\
&= \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathcal{H}} - \langle \boldsymbol{\mu}, \Phi(\mathbf{x}_i) + \Phi(\mathbf{x}_j) \rangle_{\mathcal{H}} + \langle \boldsymbol{\mu}, \boldsymbol{\mu} \rangle_{\mathcal{H}} \\
&= \mathbf{K}_{i,j} - \frac{1}{n} \sum_{k=1}^{n} (\mathbf{K}_{i,k} + \mathbf{K}_{j,k}) + \frac{1}{n^2} \sum_{k,l=1}^{n} \mathbf{K}_{k,l} \,.
\end{aligned}$$

- This can be rewritten in matricial form:

$$\mathbf{K}^c = \mathbf{K} - \mathbf{U}\mathbf{K} - \mathbf{K}\mathbf{U} + \mathbf{U}\mathbf{K}\mathbf{U} = (\mathbf{I} - \mathbf{U})\,\mathbf{K}\,(\mathbf{I} - \mathbf{U}) \,,$$

where $\mathbf{U}_{i,j} = 1/n$ for $1 \leq i, j \leq n$.

Kernel trick Summary

- The kernel trick is a trivial statement with important applications.
- It can be used to obtain nonlinear versions of well-known linear algorithms, e.g., by replacing the classical inner product by a Gaussian kernel.
- It can be used to apply classical algorithms to non vectorial data (e.g., strings, graphs) by again replacing the classical inner product by a valid kernel for the data.
- It allows in some cases to embed the initial space to a larger feature space and involve points in the feature space with no pre-image (e.g., barycenter).

Outline

Motivation

- An RKHS is a space of (potentially nonlinear) functions, and $\| f \|_{\mathcal{H}}$ measures the smoothness of f.
- Given a set of data $(\mathbf{x}_i \in \mathcal{X}, y_i \in \mathbb{R})_{i=1,\dots,n}$, a natural way to estimate a regression function $f : \mathcal{X} \to \mathbb{R}$ is to solve something like:

$$\min_{f \in \mathcal{H}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \| f \|_{\mathcal{H}}^2}_{\text{regularization}} . \tag{1}$$

  for a loss function ℓ such as $\ell(y, t) = (y - t)^2$.
- How to solve in practice this problem, potentially in infinite dimension?

The Theorem

Representer Theorem

- Let $\mathcal{X}$ be a set endowed with a p.d. kernel K, $\mathcal{H}$ the corresponding RKHS, and $\mathcal{S} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \subseteq \mathcal{X}$ a finite set of points in $\mathcal{X}$.
- Let $\Psi : \mathbb{R}^{n+1} \to \mathbb{R}$ be a function of $n+1$ variables, strictly increasing with respect to the last variable.
- Then, any solution to the optimization problem:

$$\min_{f \in \mathcal{H}} \Psi\left(f\left(\mathbf{x}_1\right), \cdots, f\left(\mathbf{x}_n\right), \| f \|_{\mathcal{H}}\right),$$

admits a representation of the form:

$$\forall \mathbf{x} \in \mathcal{X}, \quad f\left(\mathbf{x}\right) = \sum_{i=1}^{n} \alpha_i K\left(\mathbf{x}_i, \mathbf{x}\right) = \sum_{i=1}^{n} \alpha_i K_{\mathbf{x}_i}(\mathbf{x}).$$

In other words, the solution lives in a finite-dimensional subspace:

$$f \in \mathrm{Span}(K_{\mathbf{x}_1}, \ldots, K_{\mathbf{x}_n}).$$

Proof (1/2)

- Let $\xi(f)$ be the functional that is minimized in the statement of the representer theorem, and $\mathcal{H}_{\mathcal{S}}$ the linear span in $\mathcal{H}$ of the vectors $K_{\mathbf{x}_i}$:

$$\mathcal{H}_{\mathcal{S}} = \left\{ f \in \mathcal{H} : f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}), (\alpha_1, \cdots, \alpha_n) \in \mathbb{R}^n \right\}.$$

- $\mathcal{H}_{\mathcal{S}}$ is a finite-dimensional subspace, therefore any function $f \in \mathcal{H}$ can be uniquely decomposed as:

$$f = f_{\mathcal{S}} + f_{\perp},$$

with $f_{\mathcal{S}} \in \mathcal{H}_{\mathcal{S}}$ and $f_{\perp} \perp \mathcal{H}_{\mathcal{S}}$ (by orthogonal projection).

Proof (2/2)

- $\mathcal{H}$ being a RKHS it holds that:

$$\forall i = 1, \cdots, n, \quad f_{\perp}(\mathbf{x}_i) = \langle f_{\perp}, K_{\mathbf{x}_i} \rangle_{\mathcal{H}} = 0\,,$$

because $K_{\mathbf{x}_i} = K(\mathbf{x}_i, .) \in \mathcal{H}_{\mathcal{S}}$ and $f_{\perp} \perp \mathcal{H}_{\mathcal{S}}$, therefore:

$$\forall i = 1, \cdots, n, \quad f(\mathbf{x}_i) = f_{\mathcal{S}}(\mathbf{x}_i)\,.$$

- Pythagoras' theorem in $\mathcal{H}$ then shows that:

$$\| f \|_{\mathcal{H}}^2 = \| f_{\mathcal{S}} \|_{\mathcal{H}}^2 + \| f_{\perp} \|_{\mathcal{H}}^2\,.$$

- As a consequence, $\xi(f) \geq \xi(f_{\mathcal{S}})$, with equality if and only if $\| f_{\perp} \|_{\mathcal{H}} = 0$. The minimum of Ψ is therefore necessarily in $\mathcal{H}_{\mathcal{S}}$.

□

Remarks

Often the function Ψ has the form:

$$\Psi\left(f\left(\mathbf{x}_1\right), \cdots, f\left(\mathbf{x}_n\right), \| f \|_{\mathcal{H}}\right) = c\left(f\left(\mathbf{x}_1\right), \cdots, f\left(\mathbf{x}_n\right)\right) + \lambda \Omega\left(\| f \|_{\mathcal{H}}\right)$$

where $c(.)$ measures the "fit" of f to a given problem (regression, classification, dimension reduction, ...) and Ω is strictly increasing. This formulation has two important consequences:

- Theoretically, the minimization will enforce the norm $\| f \|_{\mathcal{H}}$ to be "small", which can be beneficial by ensuring a sufficient level of smoothness for the solution (regularization effect).
- Practically, we know by the representer theorem that the solution lives in a subspace of dimension n, which can lead to efficient algorithms although the RKHS itself can be of infinite dimension.

Practical use of the representer theorem (1/2)

- When the representer theorem holds, we know that we can look for a solution of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) , \quad \text{for some } \boldsymbol{\alpha} \in \mathbb{R}^n.$$

- For any $j = 1, \ldots, n$, we have

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

- Furthermore,

$$\| f \|_{\mathcal{H}}^2 = \left\| \sum_{i=1}^{n} \alpha_i K_{\mathbf{x}_i} \right\|_{\mathcal{H}}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}.$$

Practical use of the representer theorem (2/2)

- Therefore, a problem of the form

$$\min_{f \in \mathcal{H}} \Psi \left(f\left(\mathbf{x}_1\right), \cdots, f\left(\mathbf{x}_n\right), \| f \|_{\mathcal{H}}^2 \right)$$

  is equivalent to the following n-dimensional optimization problem:

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

- This problem can usually be solved analytically or by numerical methods; we will see many examples in the next sections.

Remarks

Dual interpretations of kernel methods

Most kernel methods have two complementary interpretations:

- A geometric interpretation in the feature space, thanks to the kernel trick. Even when the feature space is "large", most kernel methods work in the linear span of the embeddings of the points available.
- A functional interpretation, often as an optimization problem over (subsets of) the RKHS associated to the kernel.

The representer theorem has important consequences, but it is in fact rather trivial. We are looking for a function f in $\mathcal{H}$ such that for all $\mathbf{x}$ in $\mathcal{X}$, $f(\mathbf{x}) = \langle K_{\mathbf{x}}, f \rangle_{\mathcal{H}}$. The part $f^{\perp}$ that is orthogonal to the $K_{\mathbf{x}_i}$'s is thus "useless" to explain the training data.

Part 3

Kernel Methods
Supervised Learning

Supervised learning

Definition

Given:

- $\mathcal{X}$, a space of inputs,
- $\mathcal{Y}$, a space of outputs,
- $\mathcal{S}_n = (\mathbf{x}_i, y_i)_{i=1,\dots,n}$, a training set of (input,output) pairs,

the supervised learning problem is to estimate a function $h : \mathcal{X} \to \mathcal{Y}$ to predict the output for any future input.

Supervised learning

Definition

Given:

- $\mathcal{X}$, a space of inputs,
- $\mathcal{Y}$, a space of outputs,
- $\mathcal{S}_n = (\mathbf{x}_i, y_i)_{i=1,\dots,n}$, a training set of (input,output) pairs,

the supervised learning problem is to estimate a function $h : \mathcal{X} \to \mathcal{Y}$ to predict the output for any future input.

Depending on the nature of the output, this covers:

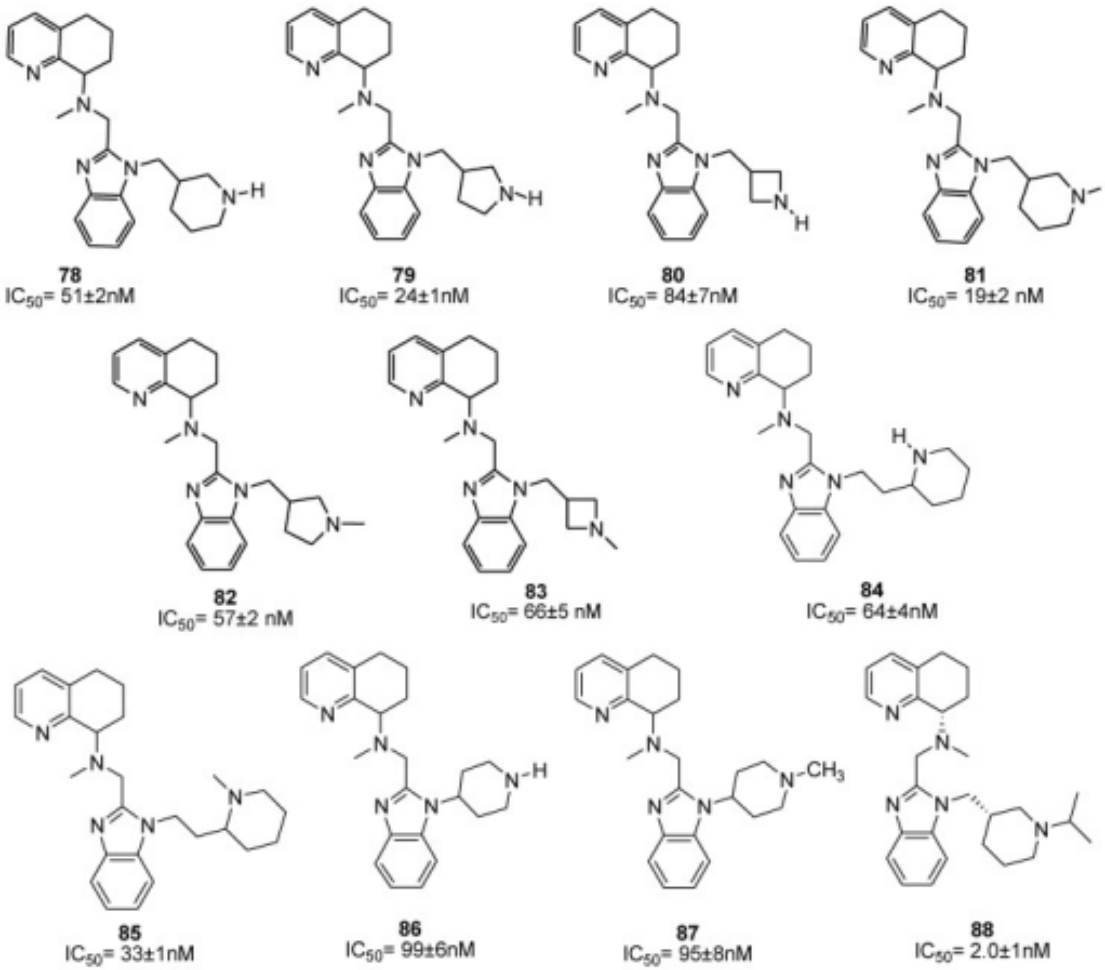
- Regression when $\mathcal{Y} = \mathbb{R}$;
- Classification when $\mathcal{Y} = \{-1, 1\}$ or any set of two labels;
- Structured output regression or classification when $\mathcal{Y}$ is more general.

Example: regression

Task: predict the capacity of a small molecule to inhibit a drug target

$\mathcal{X}$ = set of molecular structures (graphs?)

$\mathcal{Y} = \mathbb{R}$



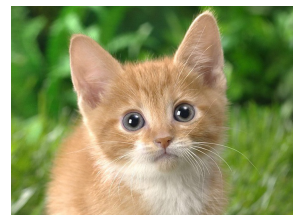
Example: classification

Task: recognize if an image is a dog or a cat

$\mathcal{X}$ = set of images ($\mathbb{R}^d$)

$\mathcal{Y} = \{\texttt{cat}, \texttt{dog}\}$





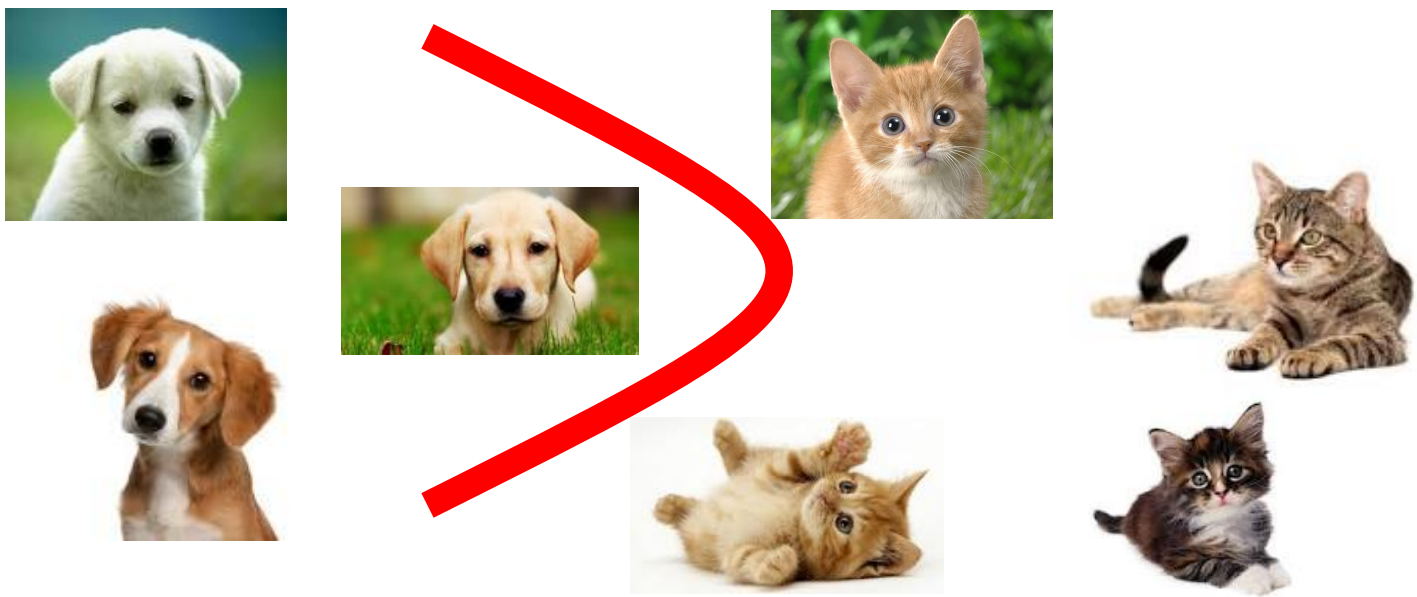


Example: classification

Task: recognize if an image is a dog or a cat

$\mathcal{X}$ = set of images ($\mathbb{R}^d$)

$\mathcal{Y} = \{\texttt{cat}, \texttt{dog}\}$

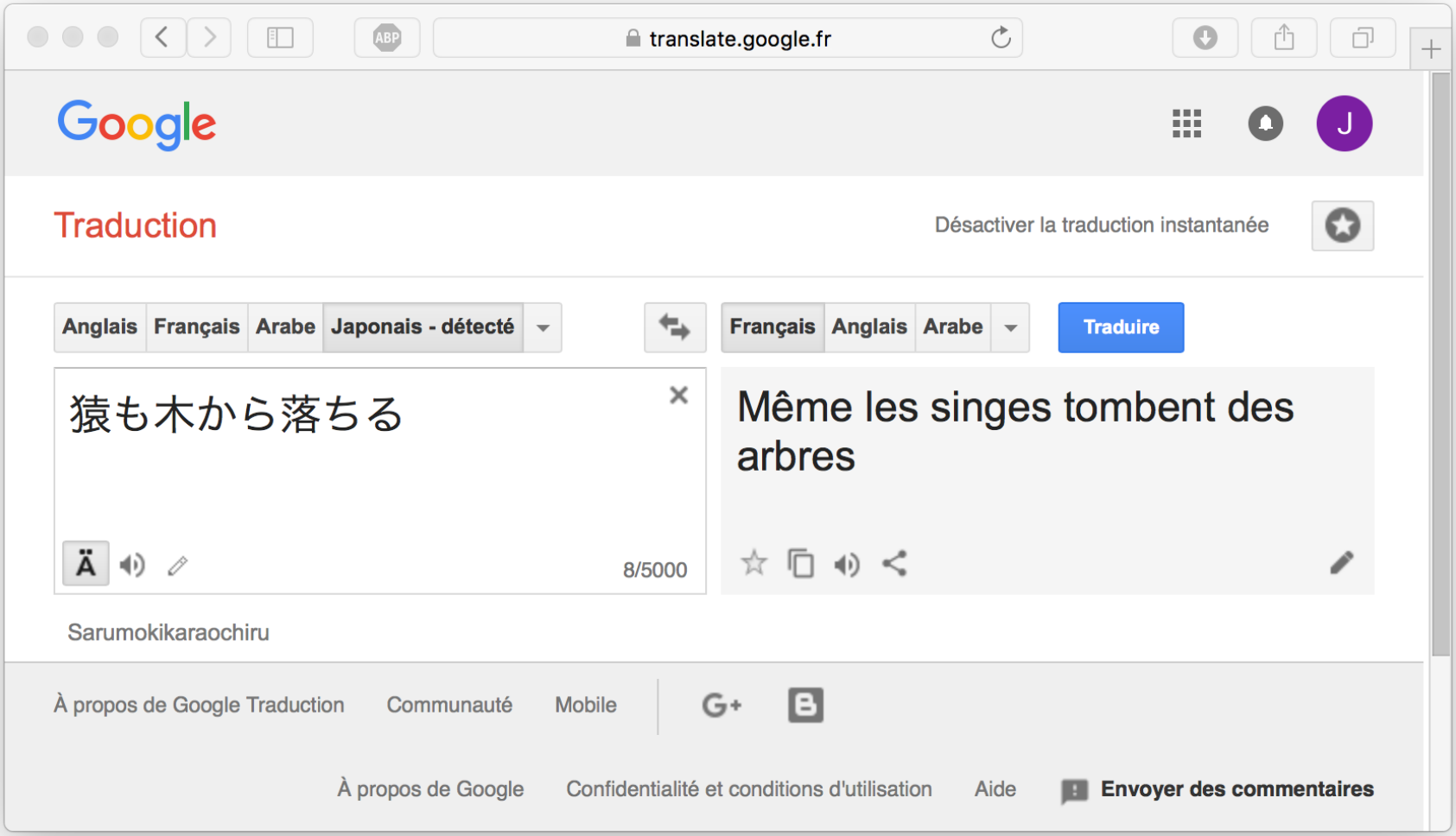


Example: structured output

Task: translate from Japanese to French

$\mathcal{X}$ = finite-length strings of japanese characters

$\mathcal{Y}$ = finite-length strings of french characters



Supervised learning with kernels: general principles

1. Express $h : \mathcal{X} \to \mathcal{Y}$ using a real-valued function $f : \mathcal{Z} \to \mathbb{R}$:
   - regression $\mathcal{Y} = \mathbb{R}$:
   $$h(\mathbf{x}) = f(\mathbf{x}) \quad \text{with} \quad f : \mathcal{X} \to \mathbb{R} \quad (\mathcal{Z} = \mathcal{X})$$
   - classification $\mathcal{Y} = \{-1, 1\}$:
   $$h(\mathbf{x}) = \text{sign}(f(\mathbf{x})) \quad \text{with} \quad f : \mathcal{X} \to \mathbb{R} \quad (\mathcal{Z} = \mathcal{X})$$
   - structured output:
   $$h(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y}) \quad \text{with} \quad f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \quad (\mathcal{Z} = \mathcal{X} \times \mathcal{Y})$$
2. Define an empirical risk function $R_n(f)$ to assess how "good" a candidate function f is on the training set $\mathcal{S}_n$, typically the average of a loss:
$$R_n(f) := \frac{1}{n} \sum_{i=1}^{n} \ell\left(f(\mathbf{x}_i), \mathbf{y}_i\right)$$
3. Define a p.d. kernel on $\mathcal{Z}$ and solve
$$\min_{f \in \mathcal{H}, \| f \|_{\mathcal{H}} \leq B} R_n(f) \quad \text{or} \quad \min_{f \in \mathcal{H}} R_n(f) + \lambda \| f \|_{\mathcal{H}}^2$$

Remarks

$$\min_{f \in \mathcal{H}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell\left(f(\mathbf{x}_i), y_i\right)}_{\text{empirical risk, data fit}} + \underbrace{\lambda \| f \|_{\mathcal{H}}^2}_{\text{regularization}} .$$

- Regularization is important, particularly in high dimension, to prevent overfitting
- When $\mathcal{Z} = \mathbb{R}^d$ and K is the linear kernel, $f = f_{\mathbf{w}}$ is a linear model and the regularization is $\|\mathbf{w}\|^2$
- Using more general spaces $\mathcal{Z}$ and kernels K allows to
  - learn non-linear functions over a functional space endowed with a natural regularization (remember, small norm in RKHS = "smooth")
  - learn functions over non-vectorial data, such as strings and graphs

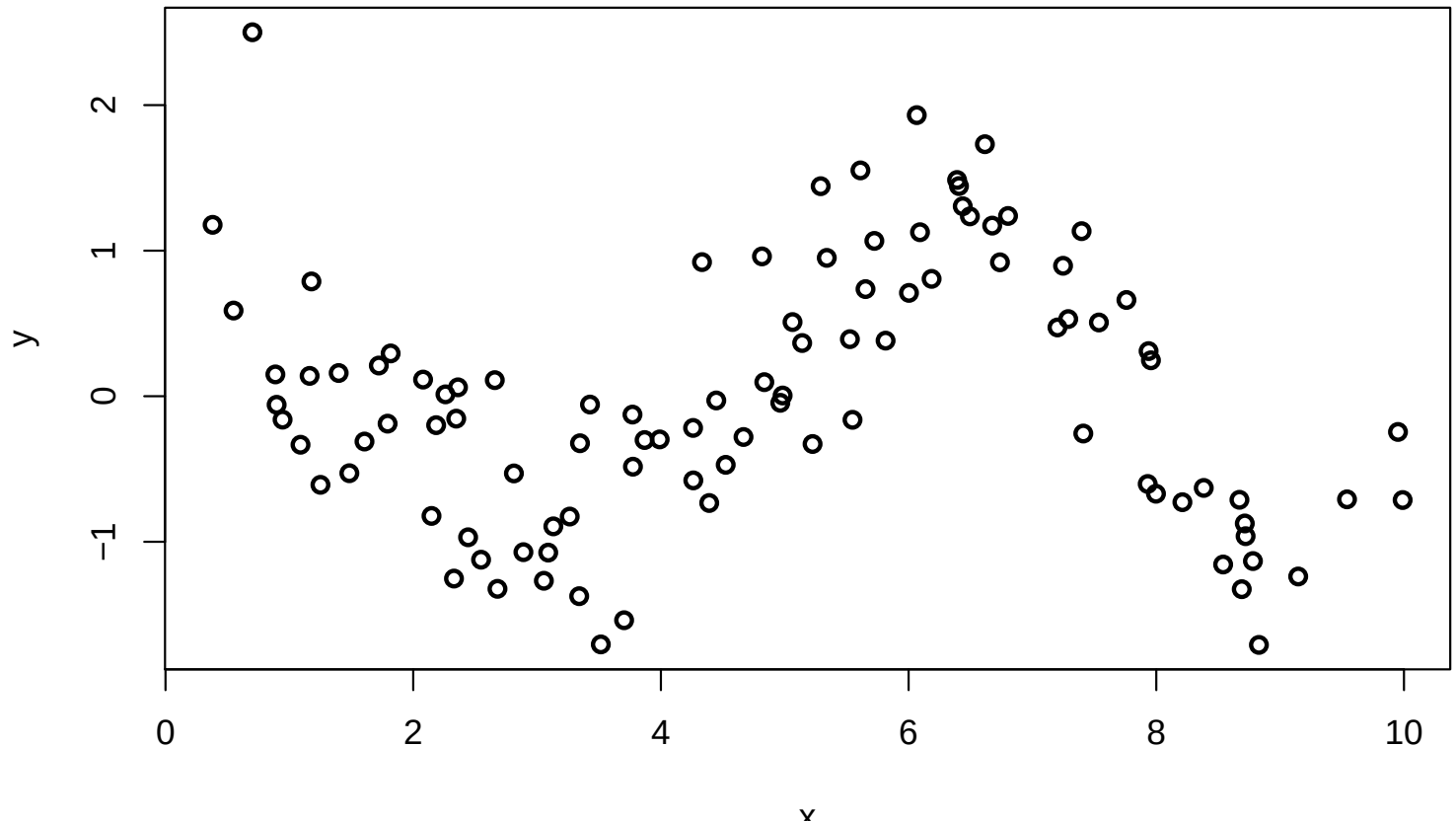
We will now see a few methods in more details

Outline

Regression

Setup

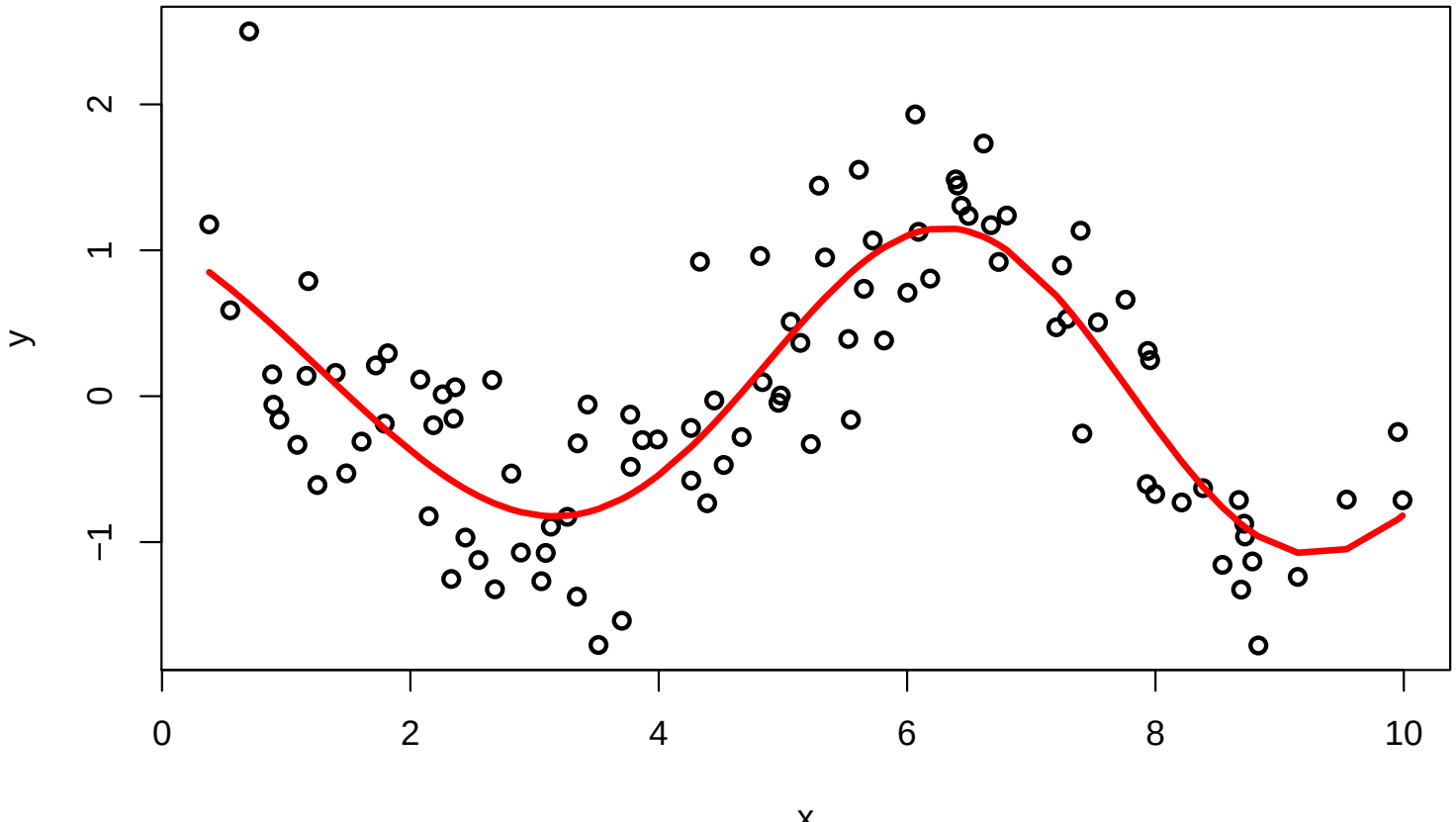
- $\mathcal{X}$ set of inputs
- $\mathcal{Y} = \mathbb{R}$ real-valued outputs
- $\mathcal{S}_n = (\mathbf{x}_i, y_i)_{i=1,\dots,n} \in (\mathcal{X} \times \mathbb{R})^n$ a training set of n pairs
- Goal = find a function $f : \mathcal{X} \to \mathbb{R}$ to predict y by $f(\mathbf{x})$



Regression

Setup

- $\mathcal{X}$ set of inputs
- $\mathcal{Y} = \mathbb{R}$ real-valued outputs
- $\mathcal{S}_n = (\mathbf{x}_i, y_i)_{i=1,\dots,n} \in (\mathcal{X} \times \mathbb{R})^n$ a training set of n pairs
- Goal = find a function $f : \mathcal{X} \to \mathbb{R}$ to predict y by $f(\mathbf{x})$



Least-square regression over a general functional space

- Let us quantify the error if f predicts $f(\mathbf{x})$ instead of y by the squared error:
$$\ell(f(\mathbf{x}), y) = (y - f(\mathbf{x}))^2$$
- Fix a set of functions $\mathcal{H}$.
- Least-square regression amounts to finding the function in $\mathcal{H}$ with the smallest empirical risk, called in this case the mean squared error (MSE):
$$\hat{f} \in \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2$$
- Issues: unstable (especially in large dimensions), overfitting if $\mathcal{H}$ is too "large".

Kernel ridge regression (KRR)

- Let us now consider a RKHS $\mathcal{H}$, associated to a p.d. kernel K on $\mathcal{X}$.
- KRR is obtained by regularizing the MSE criterion by the RKHS norm:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f\left(\mathbf{x}_i\right)\right)^2 + \lambda \| f \|_{\mathcal{H}}^2 \tag{2}$$

- *1st effect = prevent overfitting by penalizing non-smooth functions.*

Kernel ridge regression (KRR)

- Let us now consider a RKHS $\mathcal{H}$, associated to a p.d. kernel K on $\mathcal{X}$.
- KRR is obtained by regularizing the MSE criterion by the RKHS norm:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f\left(\mathbf{x}_i\right)\right)^2 + \lambda \| f \|_{\mathcal{H}}^2 \tag{2}$$

- *1st effect = prevent overfitting by penalizing non-smooth functions.*
- By the representer theorem, any solution of (2) can be expanded as

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K\left(\mathbf{x}_i, \mathbf{x}\right).$$

- *2nd effect = simplifying the solution.*

Solving KRR

- Let $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$
- Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{R}^n$
- Let $\mathbf{K}$ be the $n \times n$ Gram matrix: $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$
- We can then write:

$$\left(\hat{f}(\mathbf{x}_1), \ldots, \hat{f}(\mathbf{x}_n)\right)^\top = \mathbf{K}\boldsymbol{\alpha}$$

- The following holds as usual:

$$\| \hat{f} \|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

- The KRR problem (2) is therefore equivalent to:

$$\arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} (\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})^\top (\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}) + \lambda \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

Solving KRR

$$\arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{n}\left(\mathbf{K}\boldsymbol{\alpha}-\mathbf{y}\right)^\top\left(\mathbf{K}\boldsymbol{\alpha}-\mathbf{y}\right)+\lambda\boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}$$

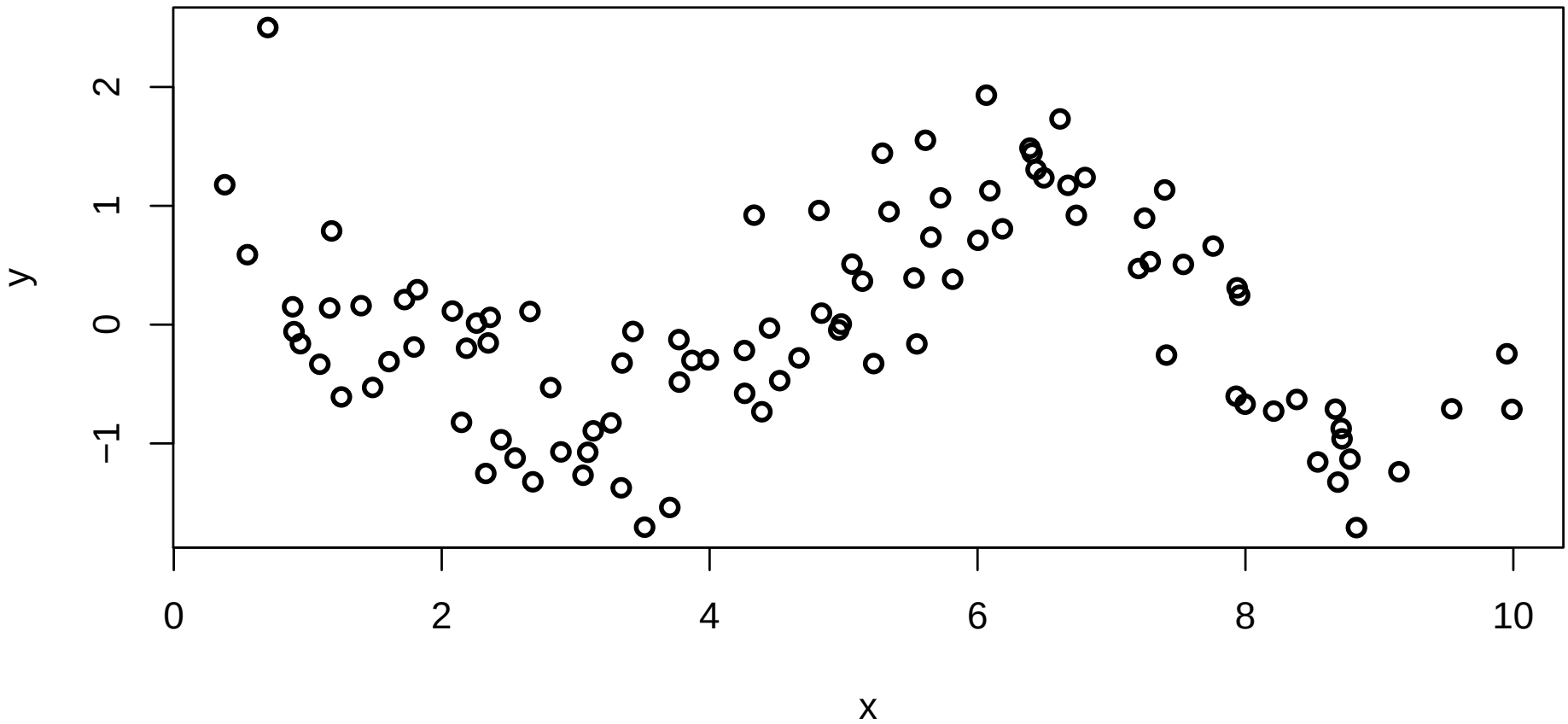
- This is a convex and differentiable function of $\boldsymbol{\alpha}$. Its minimum can therefore be found by setting the gradient in $\boldsymbol{\alpha}$ to zero:

$$\begin{aligned} 0 &= \frac{2}{n}\mathbf{K}\left(\mathbf{K}\boldsymbol{\alpha}-\mathbf{y}\right)+2\lambda\mathbf{K}\boldsymbol{\alpha} \\ &= \mathbf{K}\left[\left(\mathbf{K}+\lambda n\mathbf{I}\right)\boldsymbol{\alpha}-\mathbf{y}\right] \end{aligned}$$

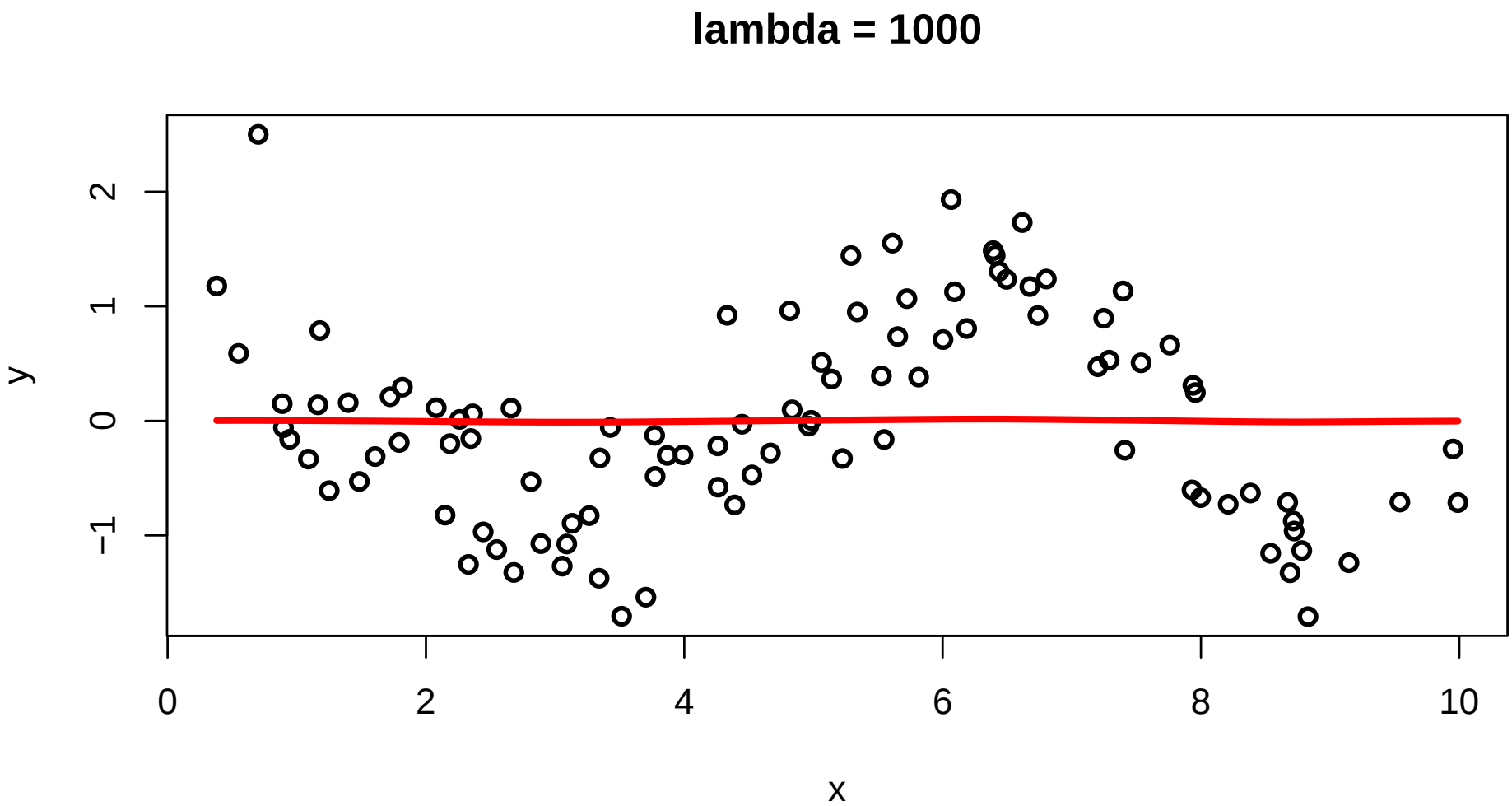
- For $\lambda > 0$, $\mathbf{K}+\lambda n\mathbf{I}$ is invertible (because $\mathbf{K}$ is positive semidefinite) so one solution is to take:

$$\boldsymbol{\alpha} = \left(\mathbf{K}+\lambda n\mathbf{I}\right)^{-1}\mathbf{y}.$$

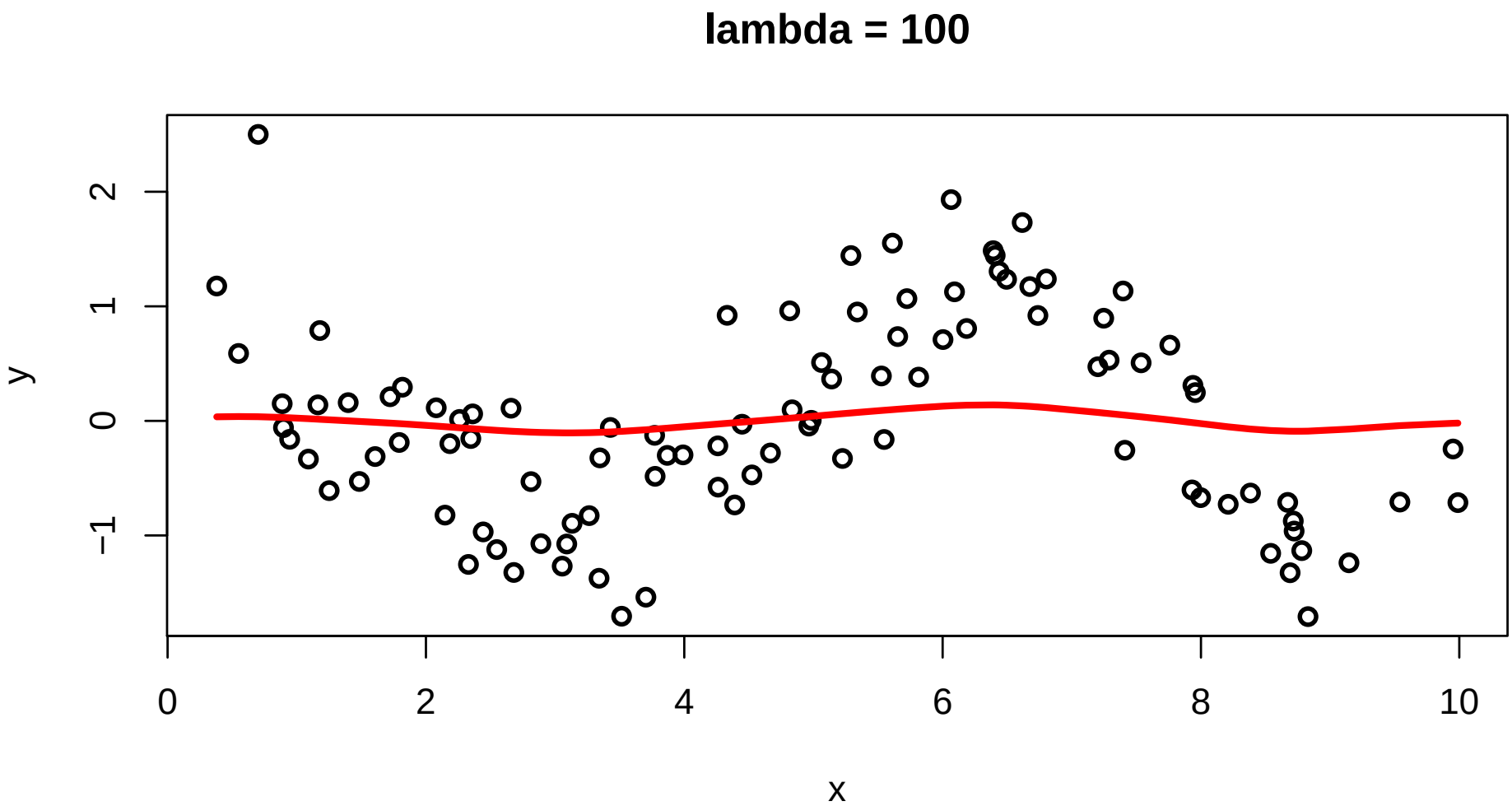
Example (KRR with Gaussian RBF kernel)



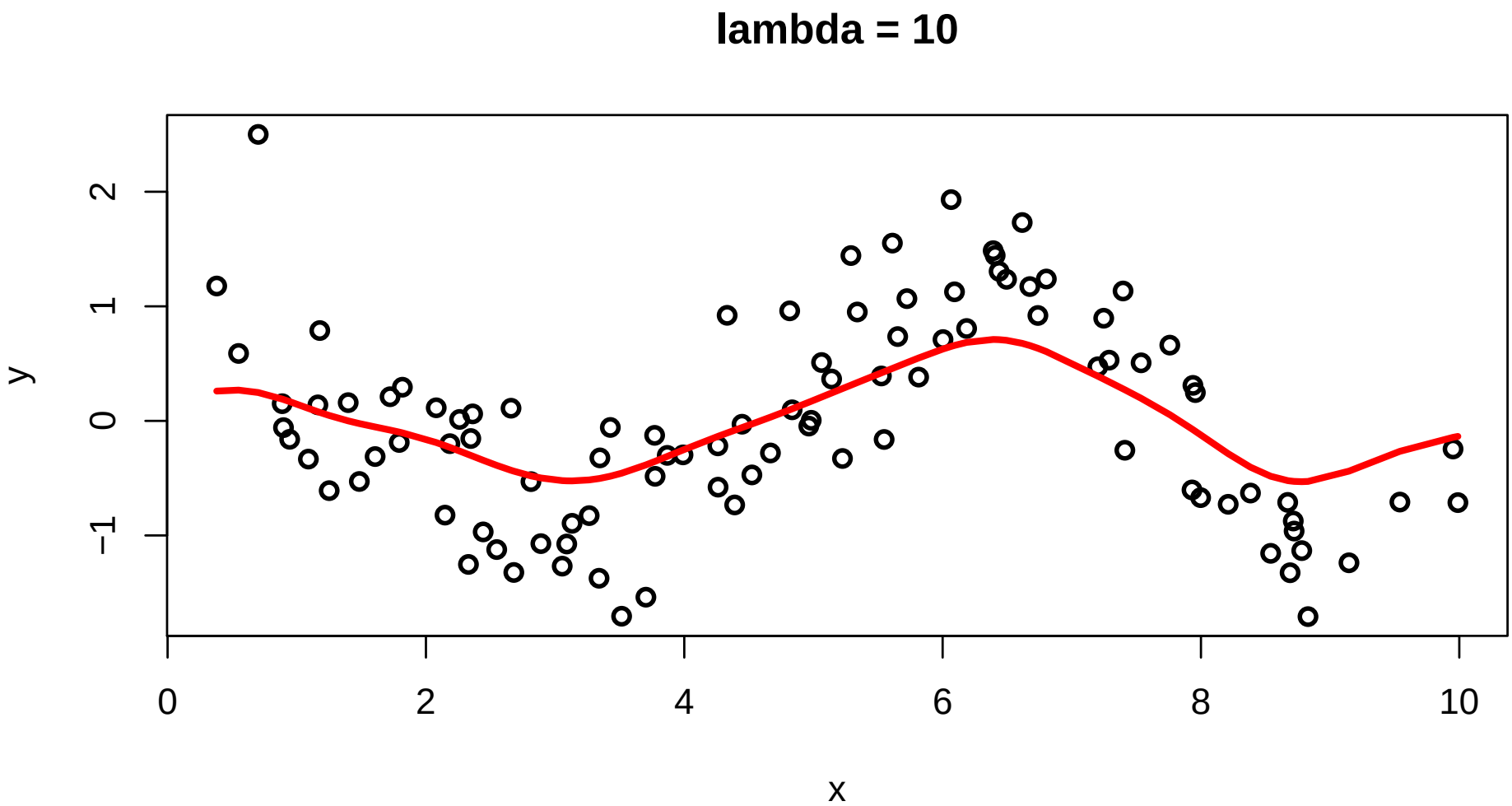
Example (KRR with Gaussian RBF kernel)



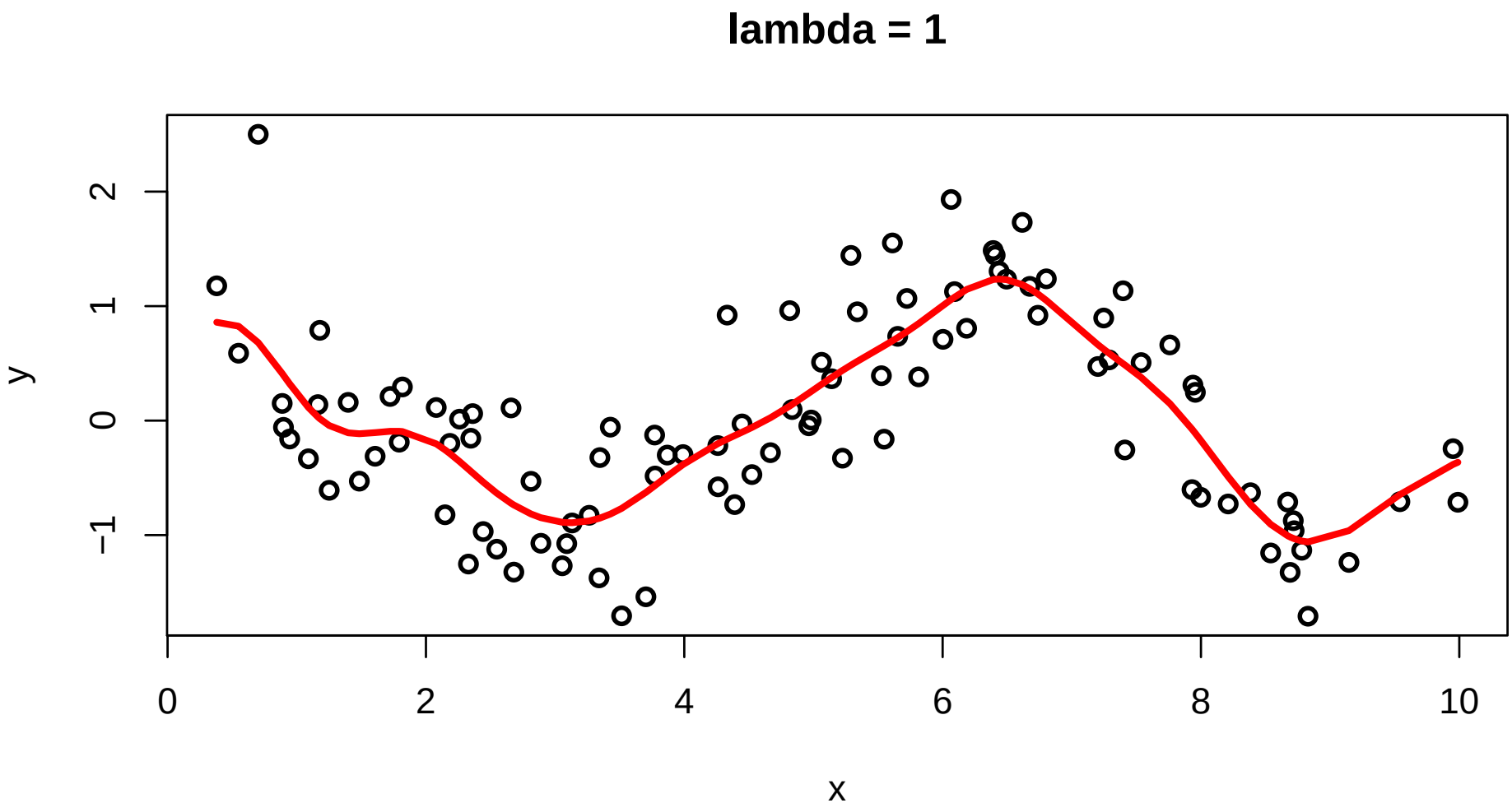
Example (KRR with Gaussian RBF kernel)



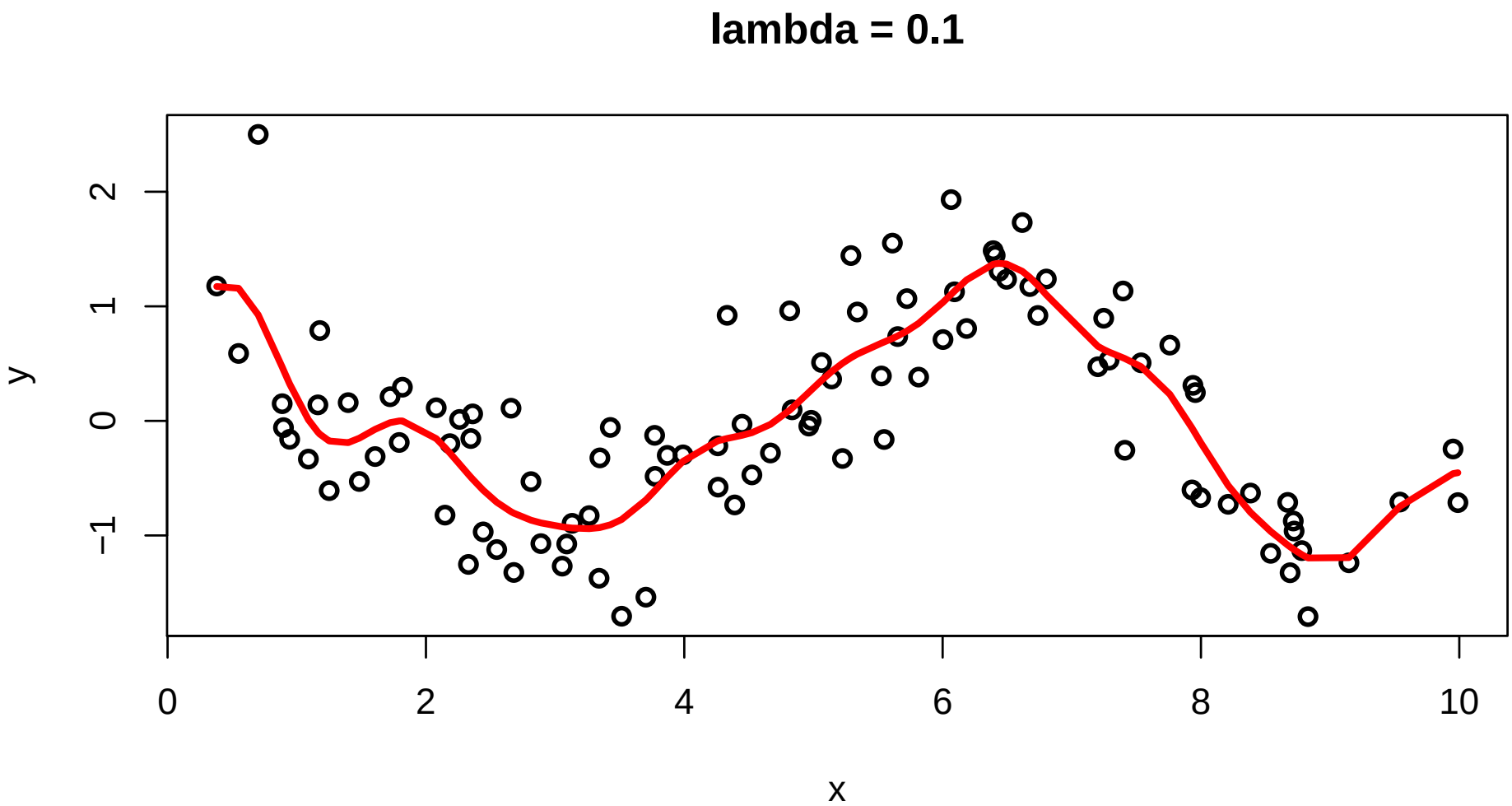
Example (KRR with Gaussian RBF kernel)



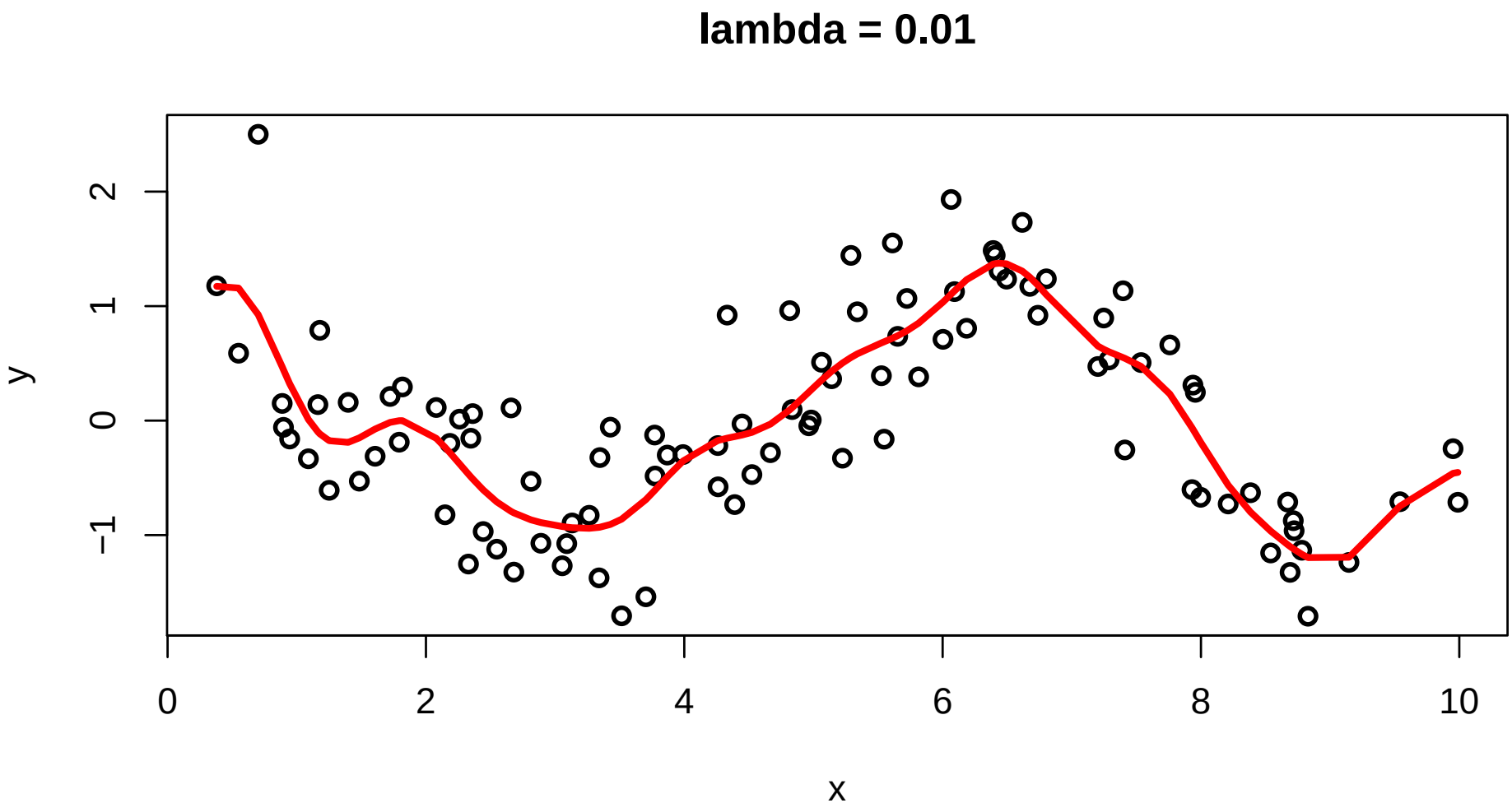
Example (KRR with Gaussian RBF kernel)



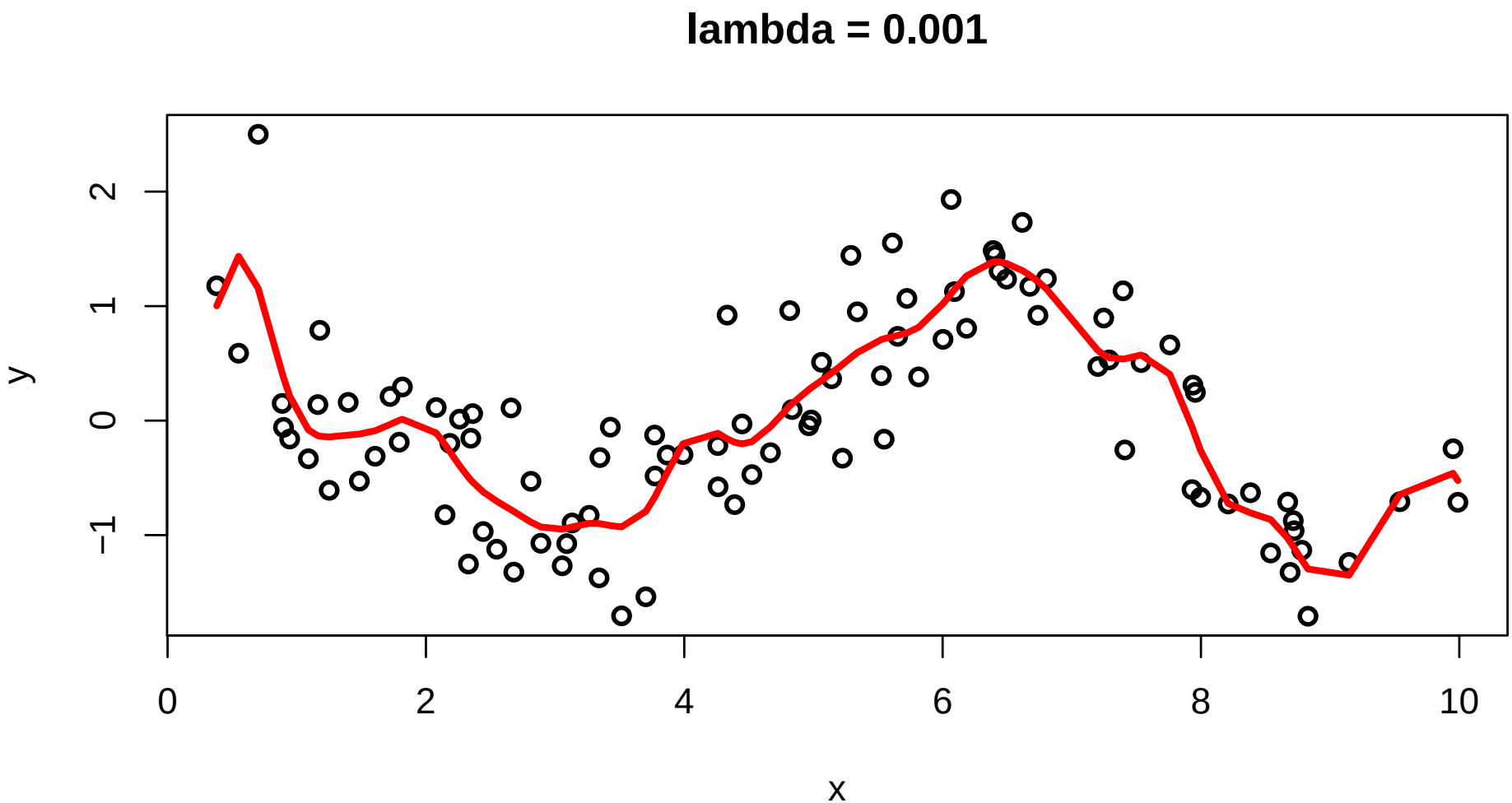
Example (KRR with Gaussian RBF kernel)



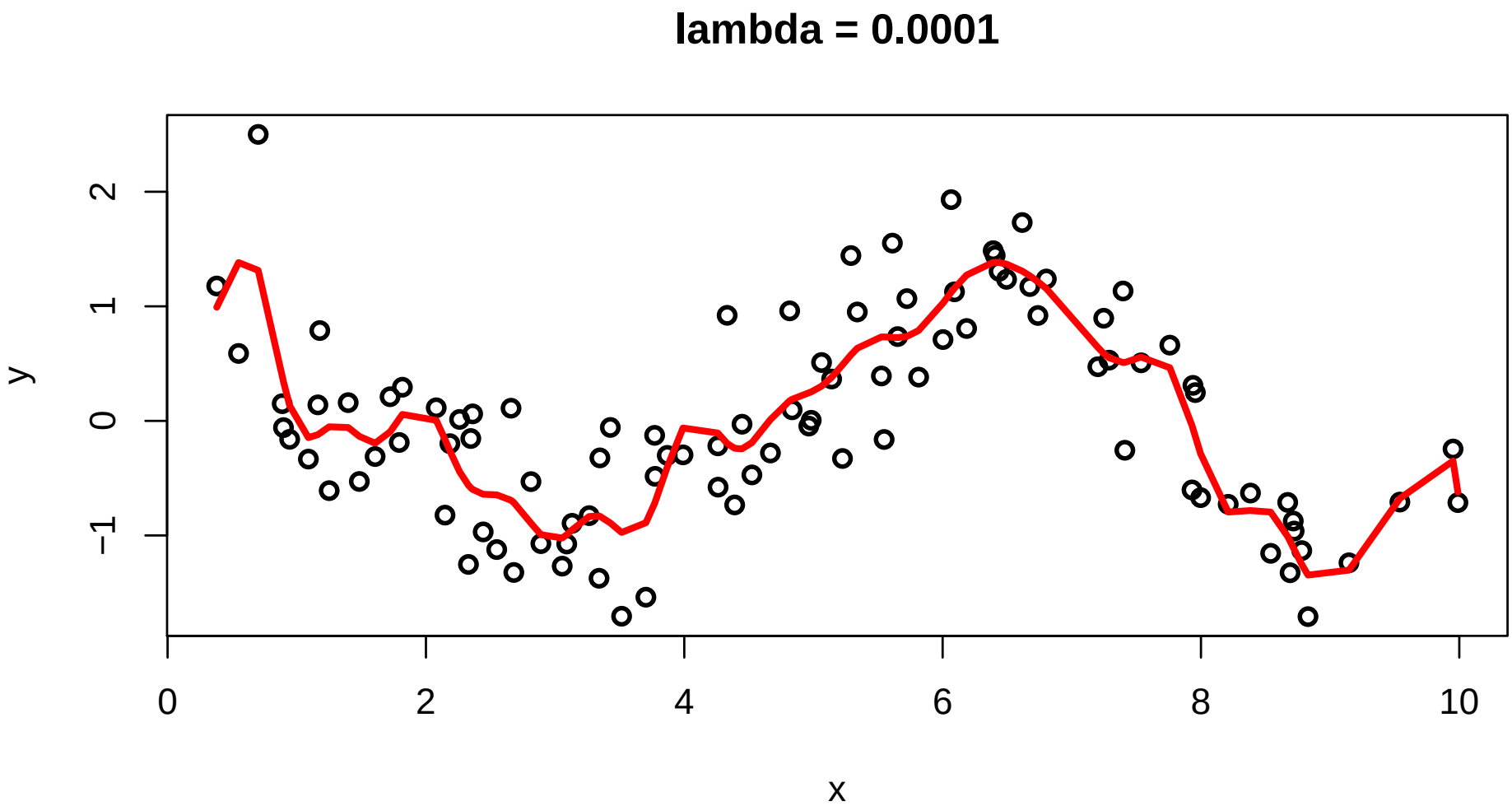
Example (KRR with Gaussian RBF kernel)



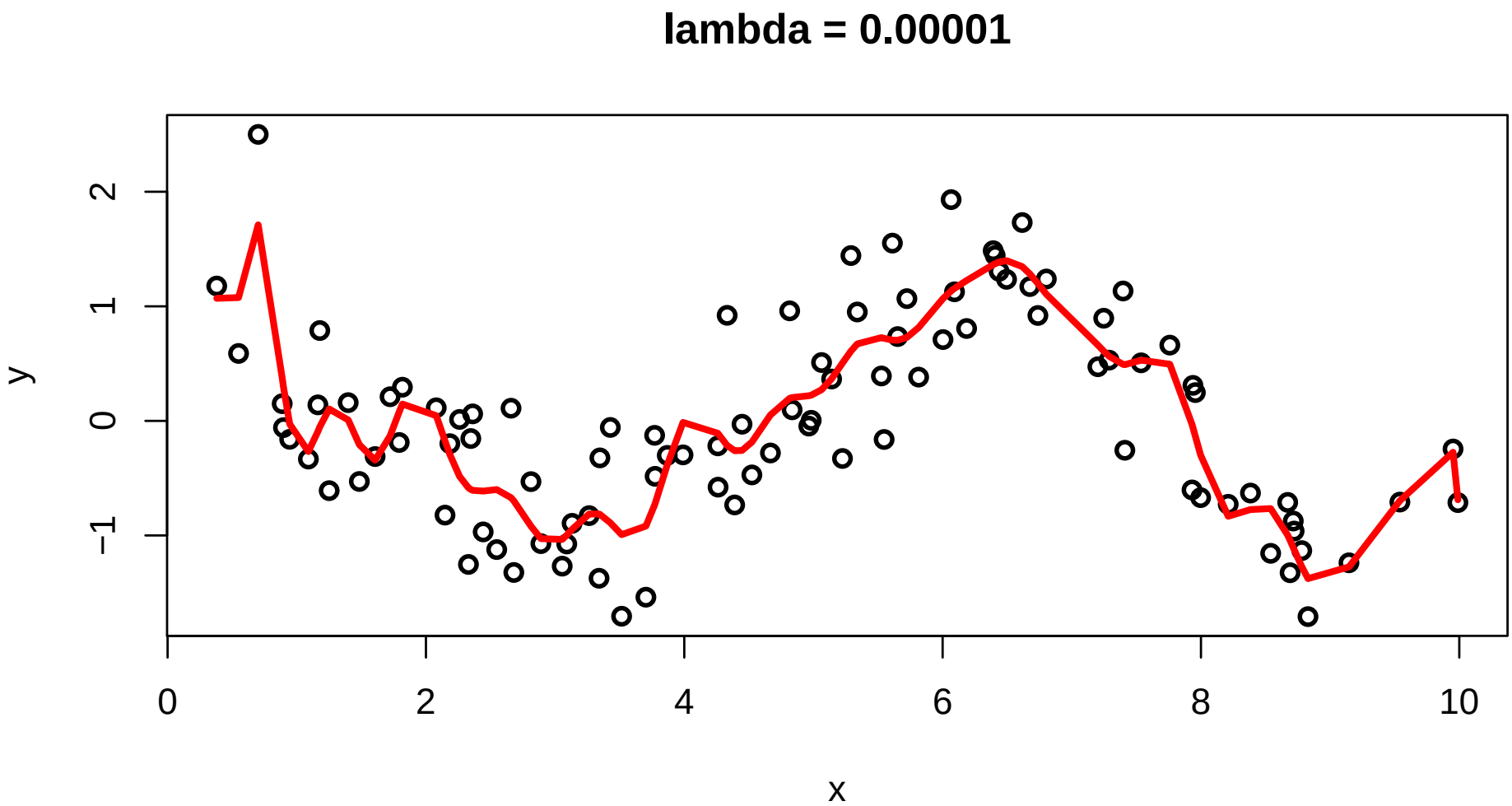
Example (KRR with Gaussian RBF kernel)



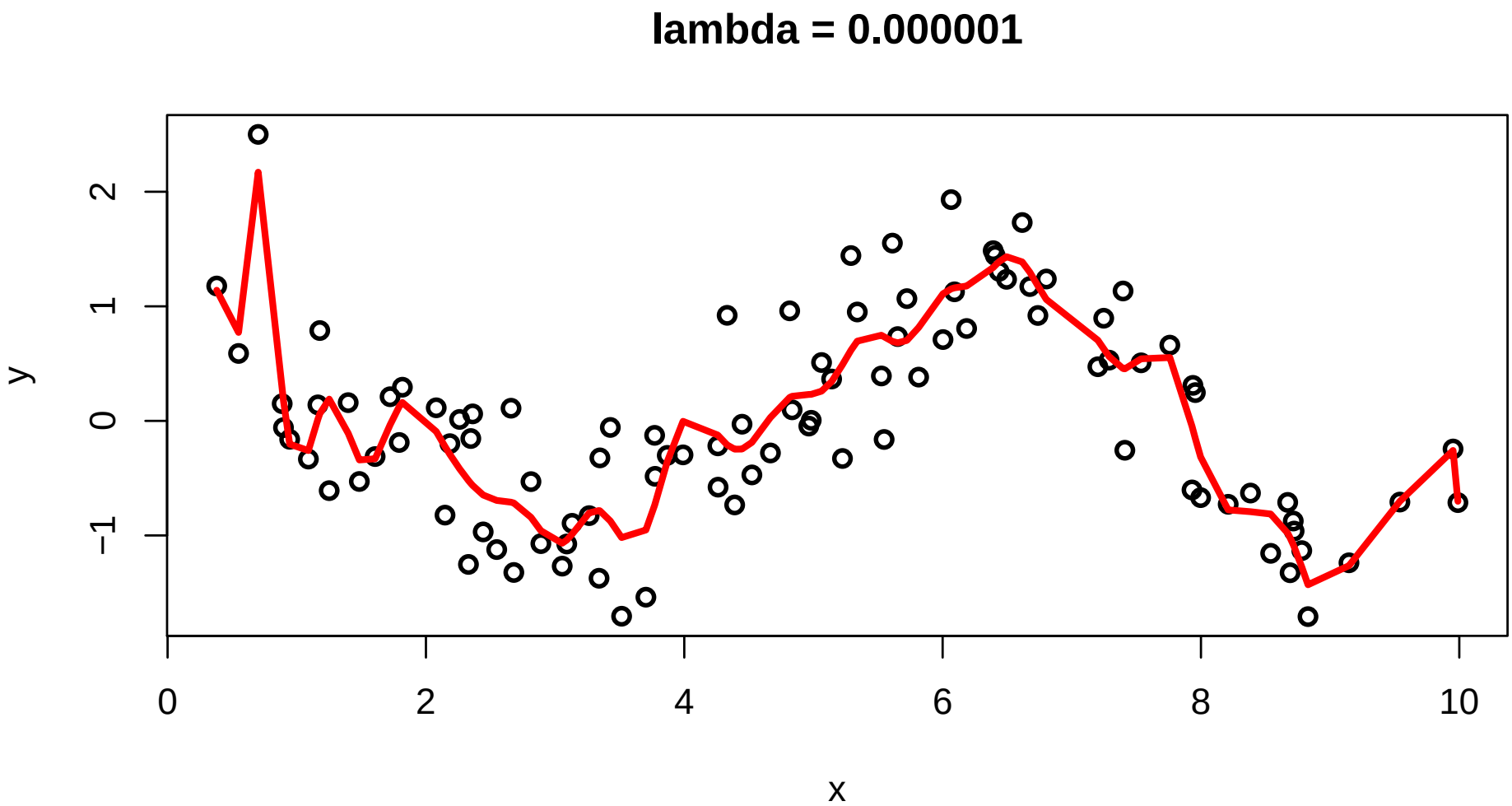
Example (KRR with Gaussian RBF kernel)



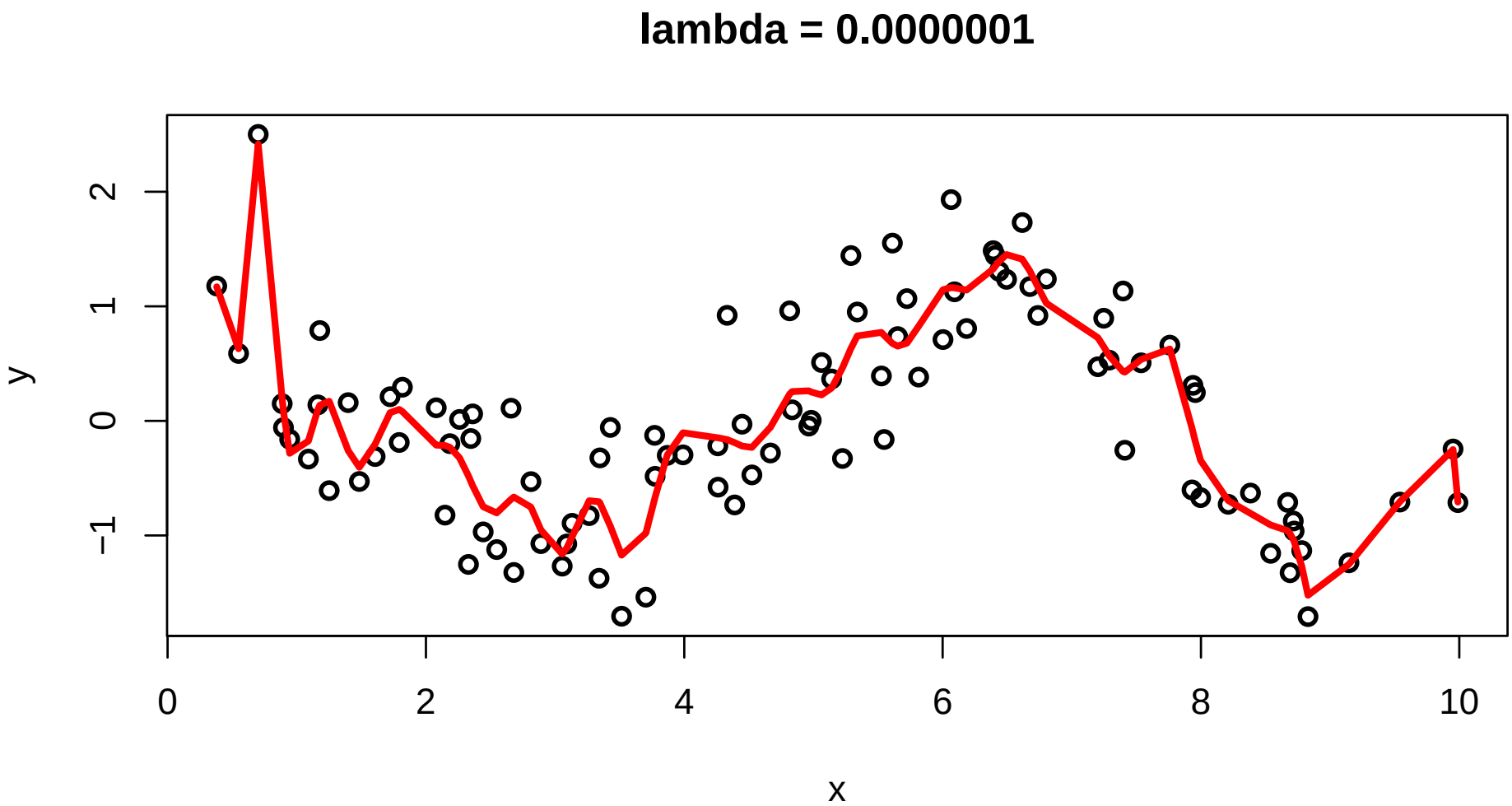
Example (KRR with Gaussian RBF kernel)



Example (KRR with Gaussian RBF kernel)



Example (KRR with Gaussian RBF kernel)



Remark: uniqueness of the solution

Let us find *all* $\boldsymbol{\alpha}$'s that solve

$$\mathbf{K}\left[\left(\mathbf{K}+\lambda n\mathbf{I}\right)\boldsymbol{\alpha}-\mathbf{y}\right]=0$$

- $\mathbf{K}$ being a symmetric matrix, it can be diagonalized in an orthonormal basis and $Ker(\mathbf{K}) \perp Im(\mathbf{K})$.
- In this basis we see that $\left(\mathbf{K}+\lambda n\mathbf{I}\right)^{-1}$ leaves $Im(\mathbf{K})$ and $Ker(\mathbf{K})$ invariant.
- The problem is therefore equivalent to:

$$\begin{aligned}
&\left(\mathbf{K}+\lambda n\mathbf{I}\right)\boldsymbol{\alpha}-\mathbf{y}\in Ker(\mathbf{K})\\
\Leftrightarrow&\boldsymbol{\alpha}-\left(\mathbf{K}+\lambda n\mathbf{I}\right)^{-1}\mathbf{y}\in Ker(\mathbf{K})\\
\Leftrightarrow&\boldsymbol{\alpha}=\left(\mathbf{K}+\lambda n\mathbf{I}\right)^{-1}\mathbf{y}+\boldsymbol{\epsilon},\ \text{with } \mathbf{K}\boldsymbol{\epsilon}=0.
\end{aligned}$$

- However, if $\boldsymbol{\alpha}'=\boldsymbol{\alpha}+\boldsymbol{\epsilon}$ with $\mathbf{K}\boldsymbol{\epsilon}=0$, then:

$$\|f-f'\|_{\mathcal{H}}^2=\left(\boldsymbol{\alpha}-\boldsymbol{\alpha}'\right)^\top\mathbf{K}\left(\boldsymbol{\alpha}-\boldsymbol{\alpha}'\right)=0,$$

therefore $f=f'$. KRR has a unique solution $f\in\mathcal{H}$, which can possibly be expressed by several $\boldsymbol{\alpha}$'s if K is singular.

Remark: link with "standard" ridge regression

- Take $\mathcal{X} = \mathbb{R}^d$ and the linear kernel $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$
- Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ the $n \times d$ data matrix
- The kernel matrix is then $\mathbf{K} = \mathbf{X}\mathbf{X}^\top$
- The function learned by KRR in that case is linear:

$$f_{KRR}(\mathbf{x}) = \mathbf{w}_{KRR}^\top \mathbf{x}$$

with

$$\mathbf{w}_{KRR} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i = \mathbf{X}^\top \boldsymbol{\alpha} = \mathbf{X}^\top \left(\mathbf{X}\mathbf{X}^\top + \lambda n \mathbf{I}\right)^{-1} \mathbf{y}$$

Remark: link with "standard" ridge regression

- On the other hand, the RKHS is the set of linear functions $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ and the RKHS norm is $\| f \|_{\mathcal{H}} = \| \mathbf{w} \|$
- We can therefore directly rewrite the original KRR problem (2) as

$$\arg \min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \mathbf{w}^\top \mathbf{x}_i \right)^2 + \lambda \| \mathbf{w} \|^2$$

$$= \arg \min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^\top \mathbf{w}$$

- Setting the gradient to 0 gives the solution:

$$\mathbf{w}_{RR} = \left(\mathbf{X}^\top \mathbf{X} + \lambda n \mathbf{I} \right)^{-1} \mathbf{X}^\top \mathbf{y}$$

- Oups, looks different from $\mathbf{w}_{KRR} = \mathbf{X}^\top \left(\mathbf{X}\mathbf{X}^\top + \lambda n \mathbf{I} \right)^{-1} \mathbf{y}$ ..?

Remark: link with "standard" ridge regression

Matrix inversion lemma

For any matrices B and C, and $\gamma > 0$ the following holds (when it makes sense):

$$B\left(CB + \gamma \mathbf{I}\right)^{-1} = \left(BC + \gamma \mathbf{I}\right)^{-1} B$$

We deduce that (of course...):

$$\mathbf{w}_{RR} = \underbrace{\left(\mathbf{X}^\top \mathbf{X} + \lambda n \mathbf{I}\right)^{-1}}_{d \times d} \mathbf{X}^\top \mathbf{y} = \mathbf{X}^\top \underbrace{\left(\mathbf{X}\mathbf{X}^\top + \lambda n \mathbf{I}\right)^{-1}}_{n \times n} \mathbf{y} = \mathbf{w}_{KRR}$$

Remark: link with "standard" ridge regression

Matrix inversion lemma

For any matrices B and C, and $\gamma > 0$ the following holds (when it makes sense):

$$B\left(CB + \gamma \mathbf{I}\right)^{-1} = \left(BC + \gamma \mathbf{I}\right)^{-1} B$$

We deduce that (of course...):

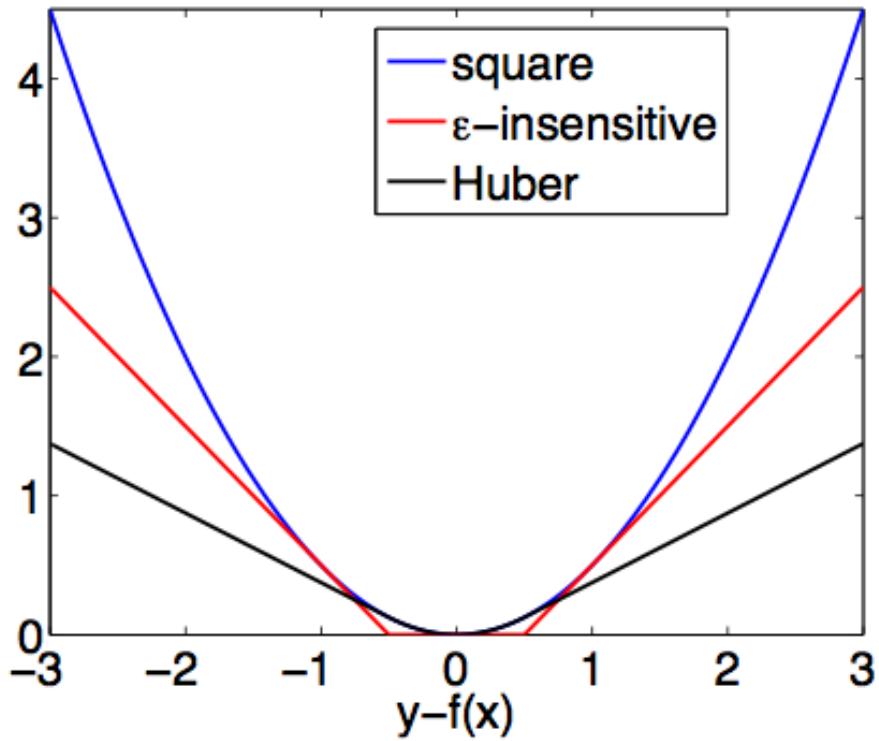
$$\mathbf{w}_{RR} = \underbrace{\left(\mathbf{X}^\top \mathbf{X} + \lambda n \mathbf{I}\right)^{-1}}_{d \times d} \mathbf{X}^\top \mathbf{y} = \mathbf{X}^\top \underbrace{\left(\mathbf{X}\mathbf{X}^\top + \lambda n \mathbf{I}\right)^{-1}}_{n \times n} \mathbf{y} = \mathbf{w}_{KRR}$$

Computationally, inverting the matrix is the expensive part, which suggest to implement:

- KRR when $d > n$ (high dimension)
- RR when $d < n$ (many points)

Robust regression

- The squared error $\ell(t, y) = (t - y)^2$ is arbitrary and sensitive to outliers
- Many other loss functions exist for regression, e.g.:



- Any loss function leads to a valid kernel method, which is usually solved by numerical optimization as there is usually no analytical solution beyond the squared error.

Weighted regression

- Given weights $W_1, \ldots, W_n \in \mathbb{R}$, a variant of ridge regression is to weight differently the error at different points:

$$\arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} W_i \left(y_i - f\left(\mathbf{x}_i\right)\right)^2 + \lambda \| f \|_{\mathcal{H}}^2$$

- By the representer theorem the solution is $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K\left(\mathbf{x}_i, \mathbf{x}\right)$ where $\boldsymbol{\alpha}$ solves, with $\mathbf{W} = \operatorname{diag}\left(W_1, \ldots, W_n\right)$:

$$\arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} \left(\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\right)^\top \mathbf{W} \left(\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\right) + \lambda \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

Weighted regression

- Setting the gradient to zero gives

$$\begin{aligned} 0 &= \frac{2}{n}\left(\mathbf{KWK}\boldsymbol{\alpha} - \mathbf{KWy}\right) + 2\lambda\mathbf{K}\boldsymbol{\alpha} \\ &= \frac{2}{n}\mathbf{K}\mathbf{W}^{\frac{1}{2}}\left[\left(\mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}} + n\lambda\mathbf{I}\right)\mathbf{W}^{-\frac{1}{2}}\boldsymbol{\alpha} - \mathbf{W}^{\frac{1}{2}}\mathbf{y}\right] \end{aligned}$$

- A solution is therefore given by

$$\left(\mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}} + n\lambda\mathbf{I}\right)\mathbf{W}^{-\frac{1}{2}}\boldsymbol{\alpha} - \mathbf{W}^{\frac{1}{2}}\mathbf{y} = 0$$

therefore

$$\boldsymbol{\alpha} = \mathbf{W}^{\frac{1}{2}}\left(\mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}} + n\lambda\mathbf{I}\right)^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{Y}$$

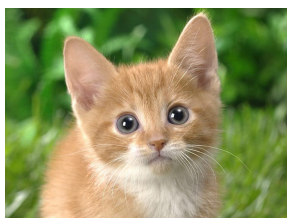
Outline

Binary classification

Setup

- $\mathcal{X}$ set of inputs
- $\mathcal{Y} = \{-1, 1\}$ binary outputs
- $\mathcal{S}_n = (\mathbf{x}_i, y_i)_{i=1,\dots,n} \in (\mathcal{X} \times \mathcal{Y})^n$ a training set of n pairs
- Goal = find a function $f : \mathcal{X} \to \mathbb{R}$ to predict y by $sign(f(\mathbf{x}))$



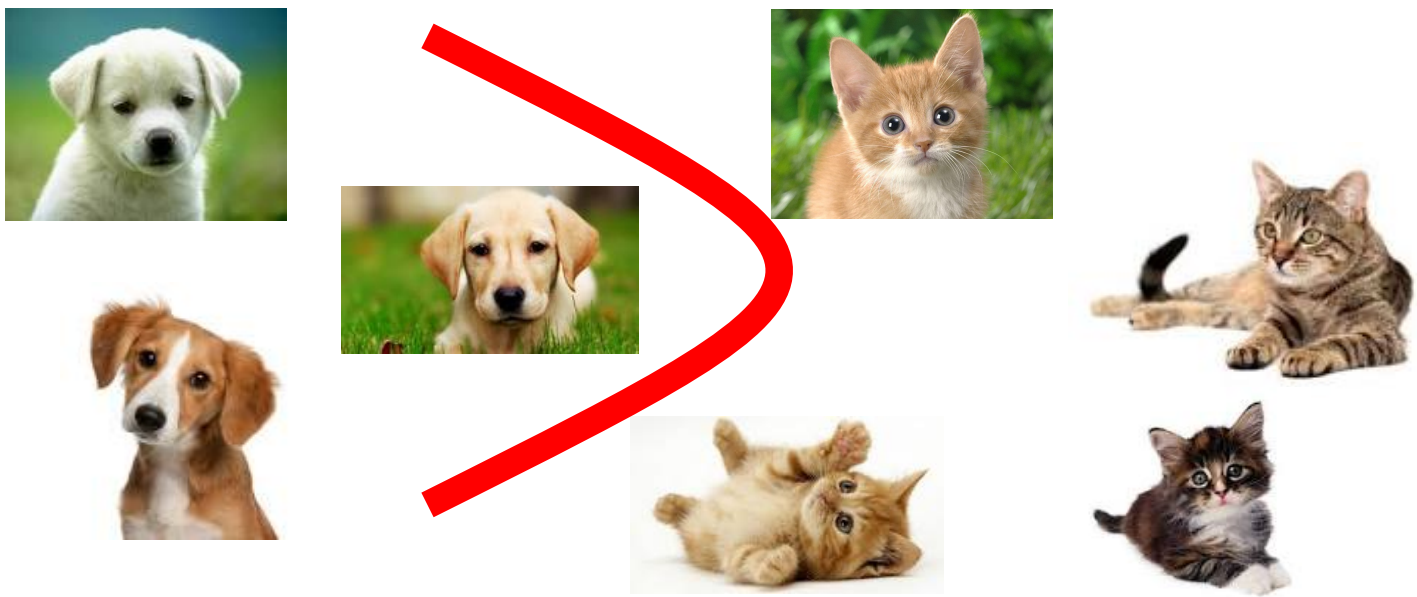




Binary classification

Setup

- $\mathcal{X}$ set of inputs
- $\mathcal{Y} = \{-1, 1\}$ binary outputs
- $\mathcal{S}_n = (\mathbf{x}_i, y_i)_{i=1,\dots,n} \in (\mathcal{X} \times \mathcal{Y})^n$ a training set of n pairs
- Goal = find a function $f : \mathcal{X} \to \mathbb{R}$ to predict y by $sign(f(\mathbf{x}))$



The 0/1 loss

- The 0/1 loss measures if a prediction is correct or not:

$$\ell_{0/1}\left(f(\mathbf{x}), y\right)) = \mathbf{1}\left(yf(\mathbf{x}) < 0\right) = \begin{cases} 0 & \text{if } y = sign\left(f(\mathbf{x})\right) \\ 1 & \text{otherwise.} \end{cases}$$

- It is then tempting to learn f by solving:

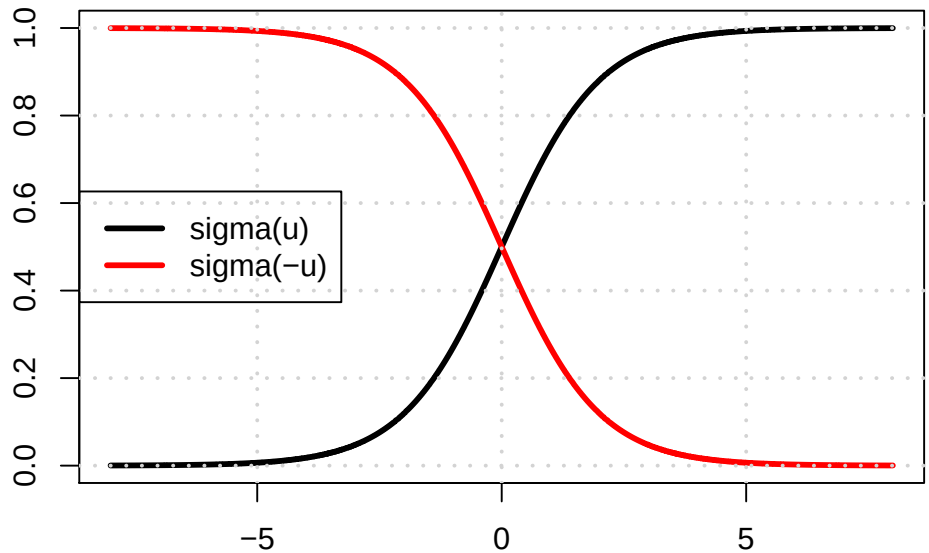
$$\min_{f \in \mathcal{H}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell_{0/1}\left(f\left(\mathbf{x}_i\right), y_i\right)}_{\text{misclassification rate}} + \underbrace{\lambda \| f \|_{\mathcal{H}}^2}_{\text{regularization}}$$

- However:
  - The problem is non-smooth, and typically NP-hard to solve
  - The regularization has no effect since the 0/1 loss is invariant by scaling of f
  - In fact, no function achieves the minimum when $\lambda > 0$ (*why?*)

The logistic loss

- An alternative is to define a probabilistic model of y parametrized by $f(\mathbf{x})$, e.g.:

$$\forall \mathbf{y} \in \{-1, 1\}, \quad p(y \mid f(\mathbf{x})) = \frac{1}{1 + e^{-yf(\mathbf{x})}} = \sigma(yf(\mathbf{x}))$$



- The logistic loss is the negative conditional likelihood:

$$\ell_{logistic}(f(\mathbf{x}), y) = -\ln p(y \mid f(\mathbf{x})) = \ln\left(1 + e^{-yf(\mathbf{x})}\right)$$

Kernel logistic regression (KLR)

$$\begin{aligned}\hat{f} &= \arg\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell_{logistic}\left(f(\mathbf{x}_i), y_i\right) + \frac{\lambda}{2}\| f \|_{\mathcal{H}}^2 \\ &= \arg\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ln\left(1 + e^{-y_i f(\mathbf{x}_i)}\right) + \frac{\lambda}{2}\| f \|_{\mathcal{H}}^2\end{aligned}$$

- Can be interpreted as a regularized conditional maximum likelihood estimator
- No explicit solution, but smooth convex optimization problem that can be solved numerically

Solving KLR

- By the representer theorem, any solution of KLR can be expanded as

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

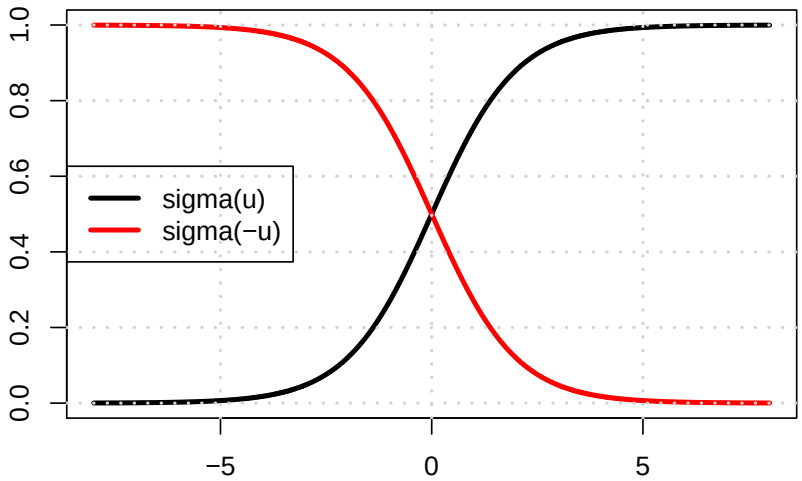
and as always we have:

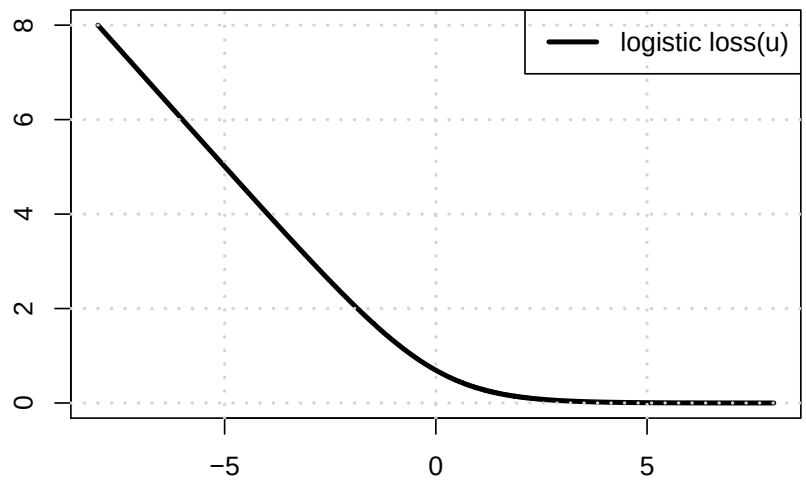
$$\left(\hat{f}(\mathbf{x}_1), \ldots, \hat{f}(\mathbf{x}_n)\right)^\top = \mathbf{K}\boldsymbol{\alpha} \quad \text{and} \quad \| \hat{f} \|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

- To find $\boldsymbol{\alpha}$ we therefore need to solve:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-y_i [\mathbf{K}\boldsymbol{\alpha}]_i}\right) + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

Technical facts





Sigmoid:

- $\sigma(u) = \frac{1}{1+e^{-u}}$
- $\sigma(-u) = 1 - \sigma(u)$
- $\sigma'(u) = \sigma(u)\sigma(-u) \geq 0$

Logistic loss:

- $\ell_{logistic}(u) = \ln\left(1 + e^{-u}\right)$
- $\ell'_{logistic}(u) = -\sigma(-u)$
- $\ell''_{logistic}(u) = \sigma(u)\sigma(-u) \geq 0$

Back to KLR

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} J(\boldsymbol{\alpha}) = \frac{1}{n}\sum_{i=1}^{n} \ell_{logistic}\left(y_i[\mathbf{K}\boldsymbol{\alpha}]_i\right) + \frac{\lambda}{2}\boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}$$

This is a smooth convex optimization problem, that can be solved by many numerical methods. Let us explicit one of them, Newton's method, which iteratively approximates J by a quadratic function and solves the quadratic problem.
The quadratic approximation near a point $\boldsymbol{\alpha}_0$ is the function:

$$J_q(\boldsymbol{\alpha}) = J(\boldsymbol{\alpha}_0) + (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)^\top \nabla J(\boldsymbol{\alpha}_0) + \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)^\top \nabla^2 J(\boldsymbol{\alpha}_0)(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)$$

Let us compute the different terms...

Computing the quadratic approximation

$$\frac{\partial J}{\partial \alpha_j} = \frac{1}{n} \sum_{i=1}^{n} \underbrace{\ell'_{logistic}\left(y_i [\mathbf{K}\boldsymbol{\alpha}]_i\right)}_{P_i(\boldsymbol{\alpha})} y_i \mathbf{K}_{ij} + \lambda [\mathbf{K}\boldsymbol{\alpha}]_j$$

therefore

$$\nabla J(\boldsymbol{\alpha}) = \frac{1}{n} \mathbf{K} \mathbf{P}(\boldsymbol{\alpha}) \mathbf{y} + \lambda \mathbf{K} \boldsymbol{\alpha}$$

where $\mathbf{P}(\boldsymbol{\alpha}) = \operatorname{diag}(P_1(\boldsymbol{\alpha}), \ldots, P_n(\boldsymbol{\alpha}))$.

$$\frac{\partial^2 J}{\partial \alpha_j \partial \alpha_l} = \frac{1}{n} \sum_{i=1}^{n} \underbrace{\ell''_{logistic}\left(y_i [\mathbf{K}\boldsymbol{\alpha}]_i\right)}_{W_i(\boldsymbol{\alpha})} y_i \mathbf{K}_{ij} y_i \mathbf{K}_{il} + \lambda \mathbf{K}_{jl}$$

therefore

$$\nabla^2 J(\boldsymbol{\alpha}) = \frac{1}{n} \mathbf{K} \mathbf{W}(\boldsymbol{\alpha}) \mathbf{K} + \lambda \mathbf{K}$$

where $\mathbf{W}(\boldsymbol{\alpha}) = \operatorname{diag}(W_1(\boldsymbol{\alpha}), \ldots, W_n(\boldsymbol{\alpha}))$.

Computing the quadratic approximation

$$J_q(\boldsymbol{\alpha}) = J(\boldsymbol{\alpha}_0) + (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)^\top \nabla J(\boldsymbol{\alpha}_0) + \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)^\top \nabla^2 J(\boldsymbol{\alpha}_0)(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)$$

Terms that depend on $\boldsymbol{\alpha}$, with $\mathbf{P} = \mathbf{P}(\boldsymbol{\alpha}_0)$ and $\mathbf{W} = \mathbf{W}(\boldsymbol{\alpha}_0)$:

- $\boldsymbol{\alpha}^\top \nabla J(\boldsymbol{\alpha}_0) = \frac{1}{n}\boldsymbol{\alpha}^\top \mathbf{K}\mathbf{P}\mathbf{y} + \lambda \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}_0$
- $\frac{1}{2}\boldsymbol{\alpha}^\top \nabla^2 J(\boldsymbol{\alpha}_0)\boldsymbol{\alpha} = \frac{1}{2n}\boldsymbol{\alpha}^\top \mathbf{K}\mathbf{W}\mathbf{K}\boldsymbol{\alpha} + \frac{\lambda}{2}\boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}$
- $-\boldsymbol{\alpha}^\top \nabla^2 J(\boldsymbol{\alpha}_0)\boldsymbol{\alpha}_0 = -\frac{1}{n}\boldsymbol{\alpha}^\top \mathbf{K}\mathbf{W}\mathbf{K}\boldsymbol{\alpha}_0 - \lambda \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}_0$

Putting it all together:

$$2J_q(\boldsymbol{\alpha}) = -\frac{2}{n}\boldsymbol{\alpha}^\top \mathbf{K}\mathbf{W}\underbrace{\left(\mathbf{K}\boldsymbol{\alpha}_0 - \mathbf{W}^{-1}\mathbf{P}\mathbf{y}\right)}_{:=\mathbf{z}} + \frac{1}{n}\boldsymbol{\alpha}^\top \mathbf{K}\mathbf{W}\mathbf{K}\boldsymbol{\alpha} + \lambda \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha} + C$$

$$= \frac{1}{n}(\mathbf{K}\boldsymbol{\alpha} - \mathbf{z})^\top \mathbf{W}(\mathbf{K}\boldsymbol{\alpha} - \mathbf{z}) + \lambda \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha} + C$$

This is a standard weighted kernel ridge regression (WKRR) problem!

Solving KLR by IRLS

In summary, one way to solve KLR is to iteratively solve a WKRR problem until convergence:

$$\boldsymbol{\alpha}^{t+1} \leftarrow \text{solveWKRR}(\mathbf{K}, \mathbf{W}^t, \mathbf{z}^t)$$

where we update $\mathbf{W}^t$ and $\mathbf{z}^t$ from $\boldsymbol{\alpha}^t$ as follows (for $i = 1, \ldots, n$):

- $m_i \leftarrow [\mathbf{K}\boldsymbol{\alpha}^t]_i$
- $P_i^t \leftarrow \ell'_{logistic}(y_i m_i) = -\sigma(-y_i m_i)$
- $W_i^t \leftarrow \ell''_{logistic}(y_i m_i) = \sigma(m_i)\sigma(-m_i)$
- $z_i^t \leftarrow m_i - P_i^t y_i / W_i^t = m_i + y_i / \sigma(y_i m_i)$

This is the kernelized version of the famous *iteratively reweighted least-square* (IRLS) method to solve the standard linear logistic regression.

Part 4

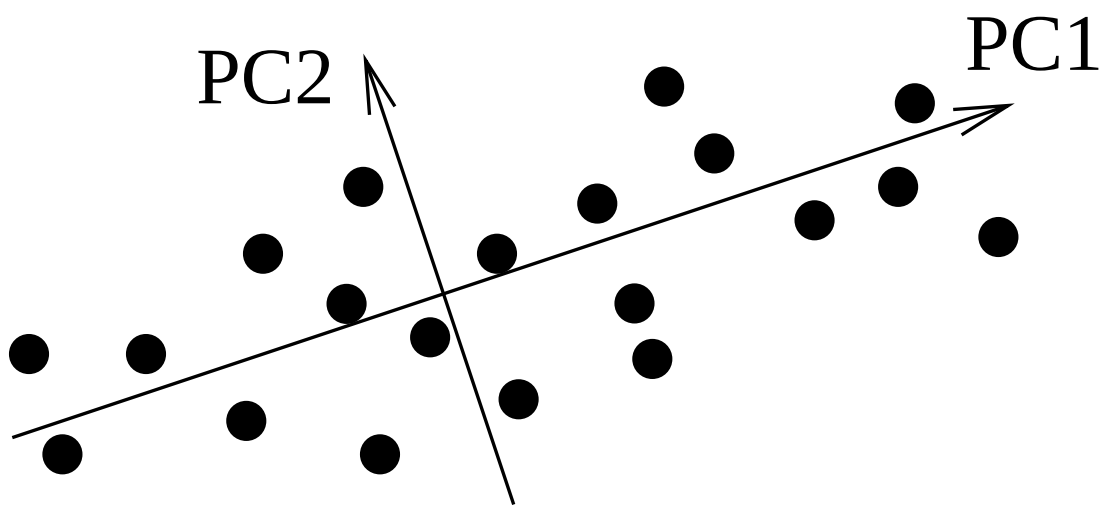
Kernel Methods
Unsupervised Learning

Outline

Principal Component Analysis (PCA)

Classical setting

- Let $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of vectors ($\mathbf{x}_i \in \mathbb{R}^d$)
- PCA is a classical algorithm in multivariate statistics to define a set of orthogonal directions that capture the maximum variance
- Applications: low-dimensional representation of high-dimensional points, visualization



Principal Component Analysis (PCA)

Formalization

- Assume that the data are centered (otherwise center them as preprocessing), i.e.:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i = 0.$$

- The orthogonal projection onto a direction $\mathbf{w} \in \mathbb{R}^d$ is the function $h_{\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}$ defined by:

$$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^\top \frac{\mathbf{w}}{\| \mathbf{w} \|}.$$

Principal Component Analysis (PCA)

Formalization

- The empirical variance captured by $h_{\mathbf{w}}$ is:

$$\hat{var}\left(h_{\mathbf{w}}\right) := \frac{1}{n}\sum_{i=1}^{n} h_{\mathbf{w}}\left(\mathbf{x}_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n} \frac{\left(\mathbf{x}_i^\top \mathbf{w}\right)^2}{\| \mathbf{w} \|^2}.$$

- The i-th principal direction $\mathbf{w}_i$ $(i = 1, \ldots, d)$ is defined by:

$$\mathbf{w}_i = \underset{\mathbf{w} \perp \{\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}\}}{\arg\max} \; \hat{var}\left(h_{\mathbf{w}}\right) \quad \text{s.t.} \quad \|\mathbf{w}\| = 1.$$

Principal Component Analysis (PCA)

Solution

- Let $\mathbf{X}$ be the $n \times d$ data matrix whose rows are the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$. We can then write:

$$\hat{var}\left(h_{\mathbf{w}}\right) = \frac{1}{n} \sum_{i=1}^{n} \frac{\left(\mathbf{x}_i^\top \mathbf{w}\right)^2}{\| \mathbf{w} \|^2} = \frac{1}{n} \frac{\mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}}{\mathbf{w}^\top \mathbf{w}}.$$

- The solutions of:

$$\mathbf{w}_i = \underset{\mathbf{w} \perp \{\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}\}}{\arg\max} \; \mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w} \quad \text{s.t.} \quad \|\mathbf{w}\| = 1$$

Principal Component Analysis (PCA)

Solution

- Let $\mathbf{X}$ be the $n \times d$ data matrix whose rows are the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$. We can then write:

$$\hat{var}\left(h_{\mathbf{w}}\right) = \frac{1}{n} \sum_{i=1}^{n} \frac{\left(\mathbf{x}_i^\top \mathbf{w}\right)^2}{\| \mathbf{w} \|^2} = \frac{1}{n} \frac{\mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}}{\mathbf{w}^\top \mathbf{w}}.$$

- The solutions of:

$$\mathbf{w}_i = \underset{\mathbf{w} \perp \{\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}\}}{\arg\max} \; \mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w} \quad \text{s.t.} \quad \|\mathbf{w}\| = 1$$

  are the successive eigenvectors of $\mathbf{X}^\top \mathbf{X}$, ranked by decreasing eigenvalues.

Kernel Principal Component Analysis (PCA)

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a set of data points in $\mathcal{X}$; let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel and $\mathcal{H}$ be its RKHS.

Formalization

- Assume that the data are centered (otherwise center by manipulating the kernel matrix), i.e.:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \quad \Longrightarrow \quad \frac{1}{n}\sum_{i=1}^{n}\varphi(\mathbf{x}_i) = 0.$$

- The orthogonal projection onto a direction $f \in \mathcal{H}$ is the function $h_f : \mathcal{X} \to \mathbb{R}$ defined by:

$$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^\top \frac{\mathbf{w}}{\|\mathbf{w}\|} \quad \Longrightarrow \quad h_f(\mathbf{x}) = \left\langle \varphi(\mathbf{x}), \frac{f}{\|f\|_{\mathcal{H}}} \right\rangle_{\mathcal{H}}.$$

Kernel Principal Component Analysis (PCA)

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a set of data points in $\mathcal{X}$; let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel and $\mathcal{H}$ be its RKHS.

Formalization

- The empirical variance captured by h_f is:

$$\hat{var}\left(h_{\mathbf{w}}\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(\mathbf{x}_i^\top \mathbf{w}\right)^2}{\| \mathbf{w} \|^2} \quad \Longrightarrow \quad \hat{var}\left(h_f\right) := \frac{1}{n}\sum_{i=1}^{n}\frac{\langle \varphi(\mathbf{x}_i), f\rangle_{\mathcal{H}}^2}{\| f \|_{\mathcal{H}}^2}.$$

- The i-th principal direction f_i $(i = 1, \ldots, d)$ is defined by:

$$f_i = \underset{f \perp \{f_1, \ldots, f_{i-1}\}}{\arg\max} \ \hat{var}\left(h_f\right) \ \text{ s.t. } \ \|f\|_{\mathcal{H}} = 1.$$

Kernel Principal Component Analysis (PCA)

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a set of data points in $\mathcal{X}$; let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel and $\mathcal{H}$ be its RKHS.

Formalization

- The empirical variance captured by h_f is:

$$\hat{var}\left(h_{\mathbf{w}}\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(\mathbf{x}_i^\top \mathbf{w}\right)^2}{\| \mathbf{w} \|^2} \quad \Longrightarrow \quad \hat{var}\left(h_f\right) := \frac{1}{n}\sum_{i=1}^{n}\frac{f(\mathbf{x}_i)^2}{\| f \|_{\mathcal{H}}^2}.$$

- The i-th principal direction f_i $(i = 1, \ldots, d)$ is defined by:

$$f_i = \underset{f \perp \{f_1, \ldots, f_{i-1}\}}{\arg\max} \sum_{i=1}^{n} f(\mathbf{x}_i)^2 \quad \text{s.t.} \quad \|f\|_{\mathcal{H}} = 1.$$

Sanity check: kernel PCA with linear kernel = PCA

- Let $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$ be the linear kernel.
- The associated RKHS $\mathcal{H}$ is the set of linear functions:

$$f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x},$$

  endowed with the norm $\| f_{\mathbf{w}} \|_{\mathcal{H}} = \| \mathbf{w} \|_{\mathbb{R}^d}$.
- Therefore we can write:

$$\hat{var}(h_{\mathbf{w}}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\left(\mathbf{x}_i^\top \mathbf{w}\right)^2}{\| \mathbf{w} \|^2} = \frac{1}{n \| f_{\mathbf{w}} \|^2} \sum_{i=1}^{n} f_{\mathbf{w}}(\mathbf{x}_i)^2.$$

- Moreover, $\mathbf{w} \perp \mathbf{w}' \Leftrightarrow f_{\mathbf{w}} \perp f_{\mathbf{w}'}$.

Kernel Principal Component Analysis (PCA)

Solution

- Kernel PCA solves, for $i = 1, \ldots, d$:

$$f_i = \underset{f \perp \{f_1, \ldots, f_{i-1}\}}{\arg\max} \sum_{i=1}^{n} f(\mathbf{x}_i)^2 \quad \text{s.t.} \quad \|f\|_{\mathcal{H}} = 1.$$

- We can apply the representer theorem (*exercise: check that is is also valid in this case*): for $i = 1, \ldots, d$, we have:

$$\forall \mathbf{x} \in \mathcal{X}, \quad f_i(\mathbf{x}) = \sum_{j=1}^{n} \alpha_{i,j} K(\mathbf{x}_j, \mathbf{x}),$$

with $\boldsymbol{\alpha}_i = (\alpha_{i,1}, \ldots, \alpha_{i,n})^\top \in \mathbb{R}^n$.

Kernel Principal Component Analysis (PCA)

- Therefore we have:

$$\| f_i \|_{\mathcal{H}}^2 = \sum_{k,l=1}^{n} \alpha_{i,k}\alpha_{i,l}K(\mathbf{x}_k, \mathbf{x}_l) = \boldsymbol{\alpha}_i^\top \mathbf{K}\boldsymbol{\alpha}_i,$$

- Similarly:

$$\sum_{k=1}^{n} f_i(\mathbf{x}_k)^2 = \boldsymbol{\alpha}_i^\top \mathbf{K}^2\boldsymbol{\alpha}_i.$$

- and

$$\langle f_i, f_j \rangle_{\mathcal{H}} = \boldsymbol{\alpha}_i^\top \mathbf{K}\boldsymbol{\alpha}_j.$$

Kernel Principal Component Analysis (PCA)

Solution

Kernel PCA maximizes in $\boldsymbol{\alpha}$ the function:

$$\boldsymbol{\alpha}_i = \underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\arg\max}\ \boldsymbol{\alpha}^\top \mathbf{K}^2 \boldsymbol{\alpha},$$

under the constraints:

$$\begin{cases} \boldsymbol{\alpha}_i^\top \mathbf{K} \boldsymbol{\alpha}_j &= 0 \quad \text{for } j = 1, \ldots, i-1\,. \\ \boldsymbol{\alpha}_i^\top \mathbf{K} \boldsymbol{\alpha}_i &= 1 \end{cases}$$

Kernel Principal Component Analysis (PCA)

Solution

- Compute the eigenvalue decomposition of the kernel matrix $\mathbf{K} = \mathbf{U}\mathbf{\Delta}\mathbf{U}^\top$, with eigenvalues $\Delta_1 \geq \ldots \geq \Delta_n \geq 0$.
- After a change of variable $\boldsymbol{\beta} = \mathbf{K}^{1/2}\boldsymbol{\alpha}$ (with $\mathbf{K}^{1/2} = \mathbf{U}\mathbf{\Delta}^{1/2}\mathbf{U}^\top$),

$$\boldsymbol{\beta}_i = \underset{\boldsymbol{\beta} \in \mathbb{R}^n}{\arg\max}\ \boldsymbol{\beta}^\top \mathbf{K} \boldsymbol{\beta},$$

under the constraints:

$$\begin{cases} \boldsymbol{\beta}_i^\top \boldsymbol{\beta}_j &= 0 \quad \text{for } j = 1, \ldots, i-1\,. \\ \boldsymbol{\beta}_i^\top \boldsymbol{\beta}_i &= 1 \end{cases}$$

- Thus, $\boldsymbol{\beta}_i = \mathbf{u}_i$ (i-th eigenvector) is a solution!
- Finally, $\boldsymbol{\alpha}_i = \frac{1}{\sqrt{\mathbf{\Delta}_i}}\mathbf{u}_i$.

Kernel Principal Component Analysis (PCA)

Summary

1. Center the Gram matrix
2. Compute the first eigenvectors $(\mathbf{u}_i, \Delta_i)$
3. Normalize the eigenvectors $\boldsymbol{\alpha}_i = \mathbf{u}_i / \sqrt{\Delta_i}$
4. The projections of the points onto the i-th eigenvector is given by $\mathbf{K}\boldsymbol{\alpha}_i$

Kernel Principal Component Analysis (PCA)

Remarks

- In this formulation, we must diagonalize the centered kernel Gram matrix, instead of the covariance matrix in the classical setting
- *Exercise: check that* $\mathbf{X}^\top \mathbf{X}$ *and* $\mathbf{X}\mathbf{X}^\top$ *have the same spectrum (up to* 0 *eigenvalues) and that the eigenvectors are related by a simple relationship.*
- This formulation remains valid for any p.d. kernel: this is kernel PCA
- Applications: nonlinear PCA with nonlinear kernels for vectors, PCA of non-vector objects (strings, graphs..) with specific kernels...

References I

N. Aronszajn. Theory of reproducing kernels. *Trans. Am. Math. Soc.*, 68:337 – 404, 1950. URL `http://www.jstor.org/stable/1990404`.

V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0387945598. URL `http://portal.acm.org/citation.cfm?id=211359`.