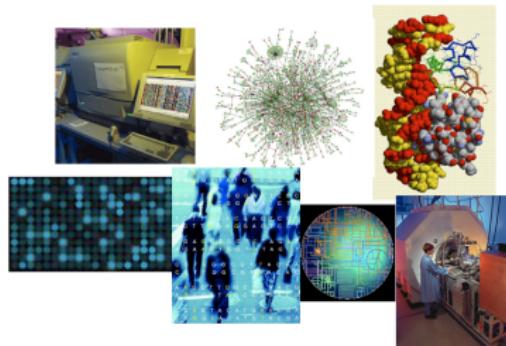
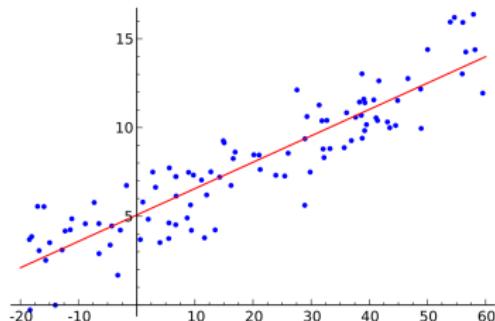


Machine Learning with Kernel Methods

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

Lectures:

- Day 1:
 - Lecture 1: Positive definite kernels
 - Lecture 2: Kernel tricks and applications: 2h
 - Tutorial 1: Applications: Kernel ridge + PCA 2h
- Day 2:
 - Lecture 3: Kernels on graphs + for graphs. 2h30
 - Tutorial 2: Kernels on graphs + 1h
 - Lecture 4: Comparing distribution using the MMD
 - Tutorial 3: MMD + KMM 1h30
- Day 3:
 - Lecture 5: Large scale learning + NTK 2h30
 - Tutorial 4: Applications to NTK, 1h30
 - Conclusion: Take home message

Outline

1 Kernels and RKHS

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

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

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

3 Kernels and Graphs

- Kernels for graphs
- Kernels on graphs

Outline

4 Characterizing probabilities with kernels

- Kernel mean embedding
- The Maximum Mean Discrepancy
- Characteristic kernels

Outline

4 Characterizing probabilities with kernels

- Kernel mean embedding
- The Maximum Mean Discrepancy
- Characteristic kernels

5 Open Problems and Research Topics

- Large-scale learning with kernels
- Foundations of deep learning from a kernel point of view

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

1 Kernels and RKHS

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

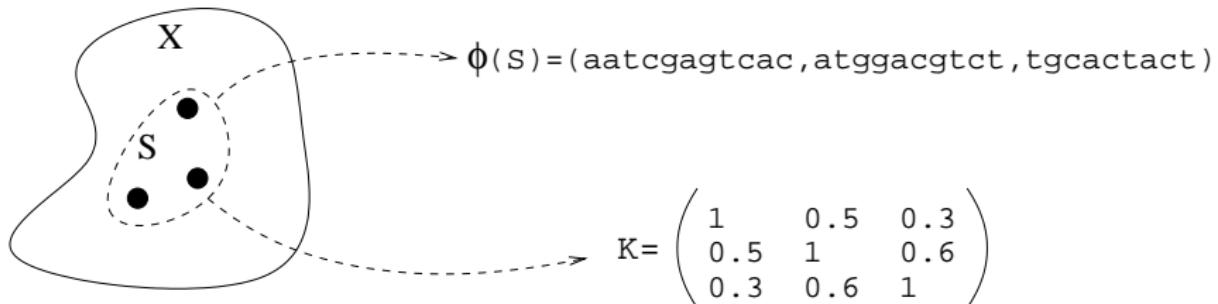
2 Kernel tricks and applications

3 Kernels and Graphs

4 Characterizing probabilities with kernels

5 Open Problems and Research Topics

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 $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ by the $n \times n$ matrix:

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

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 $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ that is **symmetric**:

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

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

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

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 $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) \in \mathcal{X}^N$, 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:

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

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:

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

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:

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

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:

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

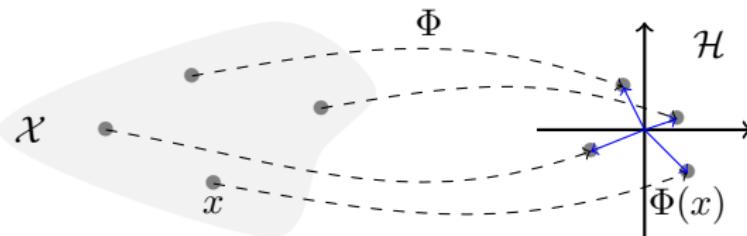
is p.d. (it is often called the **linear kernel**).

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

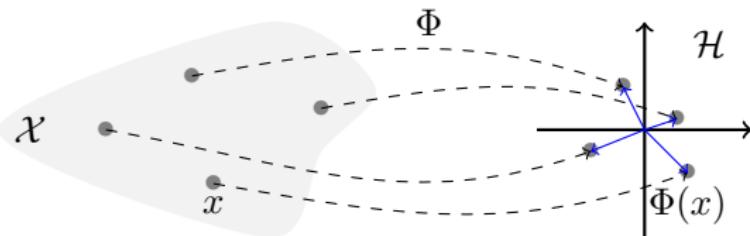


Lemma

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.:

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

A more ambitious p.d. kernel



Lemma

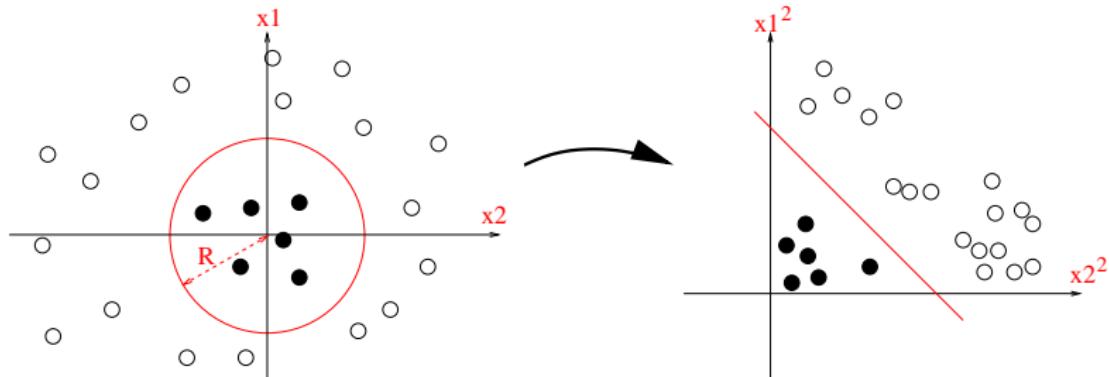
Let \mathcal{X} be any set, and $\Phi : \mathcal{X} \mapsto \mathbb{R}^d$. Then, the function $K : \mathcal{X}^2 \mapsto \mathbb{R}$ defined as follows is p.d.:

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

Proof:

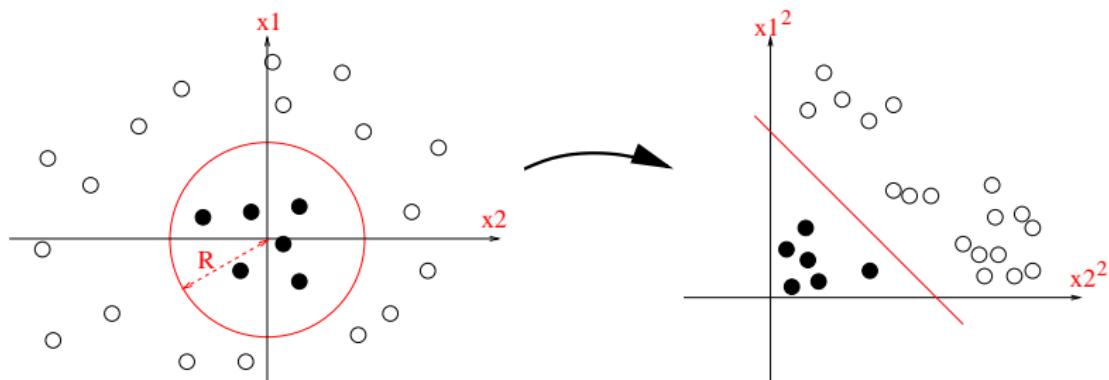
- $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathbb{R}^d} = \langle \Phi(\mathbf{x}'), \Phi(\mathbf{x}) \rangle_{\mathbb{R}^d}$
- $\sum_{i=1}^N \sum_{j=1}^N a_i a_j \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathbb{R}^d} = \| \sum_{i=1}^N a_i \Phi(\mathbf{x}_i) \|_{\mathbb{R}^d}^2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

Example: polynomial kernel



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

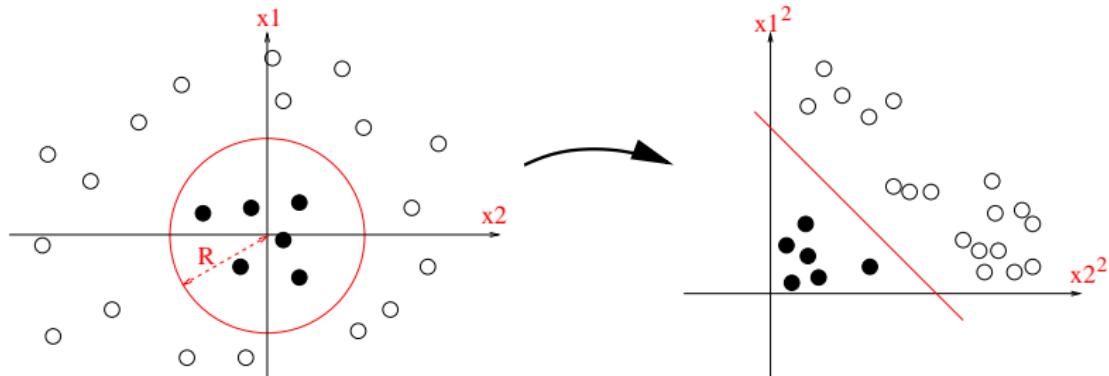
Example: polynomial kernel



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

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

Example: polynomial kernel



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

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

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

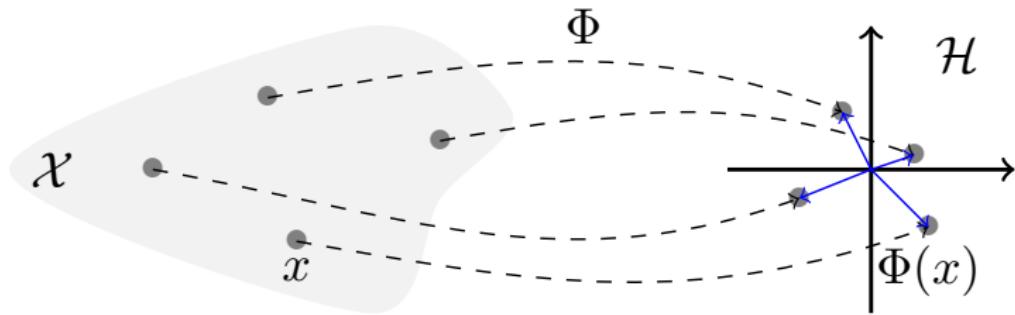
Theorem (Aronszajn, 1950)

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} :

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



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 $f \in \mathcal{H} \setminus \{0\}$.
- 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:

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

- A **Hilbert space** is a pre-Hilbert space **complete** for the norm $\|\cdot\|_{\mathcal{H}}$. 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 $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ is finite of size N .
- Any p.d. kernel $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ 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 $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N)$, with non-negative eigenvalues $0 \leq \lambda_1 \leq \dots \leq \lambda_N$, 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

1 Kernels and RKHS

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

2 Kernel tricks and applications

3 Kernels and Graphs

4 Characterizing probabilities with kernels

5 Open Problems and Research Topics

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 x in \mathcal{X} will be represented by a **function** $\Phi(x) = K_x$ 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?

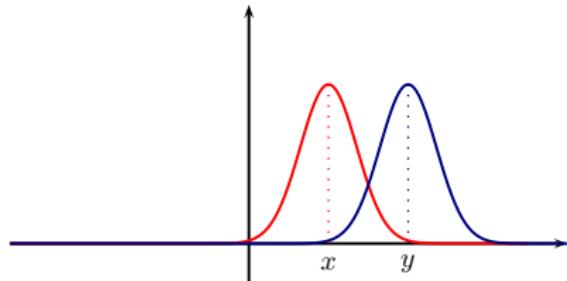
Functional spaces for machine learning

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 $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. 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 $\Phi : \mathcal{X} \rightarrow \mathcal{H}$, 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 $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ 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 $N \in \mathbb{N}, (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) \in \mathcal{X}^N$, and $(a_1, a_2, \dots, a_N) \in \mathbb{R}^N$:

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

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 $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ 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:

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

In particular Cauchy-Schwarz is valid with $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$.

- By Cauchy-Schwarz, we deduce that $\forall \mathbf{x} \in \mathcal{X}$:

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

therefore $\|f\|_{\mathcal{H}_0} = 0 \implies f = 0$.

- \mathcal{H}_0 is therefore a **pre-Hilbert space** endowed with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$.

Proof

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

- For any Cauchy sequence $(f_n)_{n \geq 0}$ in $(\mathcal{H}_0, \langle \cdot, \cdot \rangle_{\mathcal{H}_0})$, 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 Aronszajn's theorem

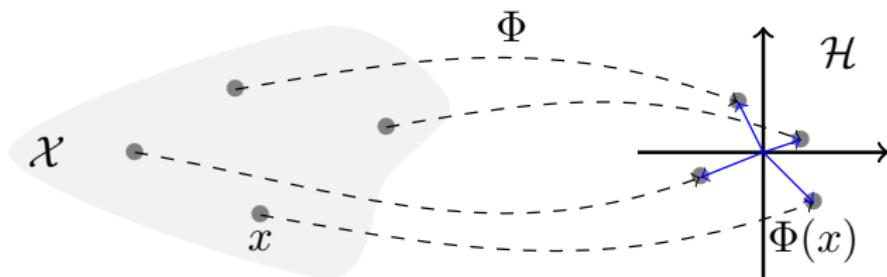
Theorem (Aronszajn, 1950)

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} :

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



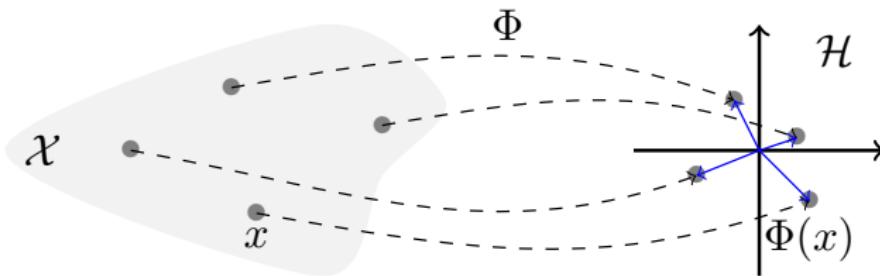
Proof of Aronzsajn's theorem

- 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 $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ 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:

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

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:

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

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 $(\mathbf{x}, f) \in \mathcal{X} \times \mathcal{H}$:

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

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. \square

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

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

1 Kernels and RKHS

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

2 Kernel tricks and applications

3 Kernels and Graphs

4 Characterizing probabilities with kernels

5 Open Problems and Research Topics

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

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

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:

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

The polynomial kernel

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

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

First step: Look for an inner-product.

$$\begin{aligned} K(\mathbf{x}, \mathbf{y}) &= \text{trace} \left(\mathbf{x}^\top \mathbf{y} \mathbf{x}^\top \mathbf{y} \right) \\ &= \text{trace} \left(\mathbf{y}^\top \mathbf{x} \mathbf{x}^\top \mathbf{y} \right) \\ &= \text{trace} \left(\mathbf{x} \mathbf{x}^\top \mathbf{y} \mathbf{y}^\top \right) \\ &= \left\langle \mathbf{x} \mathbf{x}^\top, \mathbf{y} \mathbf{y}^\top \right\rangle_F, \end{aligned}$$

where F is the Frobenius 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

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

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

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

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

¹Why 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 $K_{\mathbf{x}} : \mathbf{t} \mapsto K(\mathbf{x}, \mathbf{t}) = \langle \mathbf{x}\mathbf{x}^T, \mathbf{t}\mathbf{t}^T \rangle_F$.
- ② For all $f_{\mathbf{S}}$ in \mathcal{H} and \mathbf{x} in \mathcal{X} ,

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

□

Remark

All points \mathbf{x} in \mathcal{X} are mapped to a rank-one matrix $\mathbf{x}\mathbf{x}^T$, hence to a function $K_{\mathbf{x}} = f_{\mathbf{x}\mathbf{x}^T}$ 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 :

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

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,

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

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:

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

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?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Outline

1 Kernels and RKHS

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

2 Kernel tricks and applications

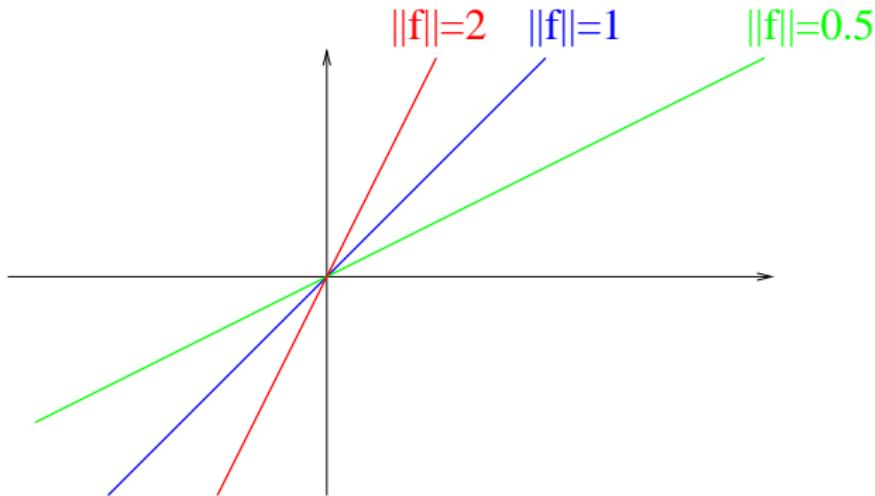
3 Kernels and Graphs

4 Characterizing probabilities with kernels

5 Open Problems and Research Topics

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 $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$:

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

- 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 \implies 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

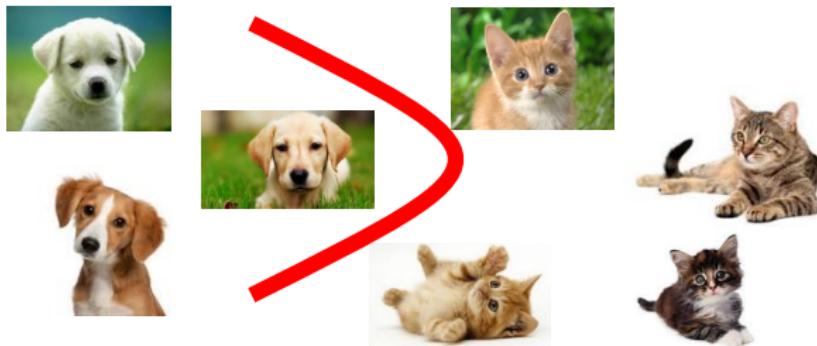
Two theoretical results underpin a family of powerful algorithms for data analysis using p.d. kernels, collectively known as **kernel methods**:

- 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** $f : \mathcal{X} \rightarrow \mathcal{Y}$ 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** $f : \mathcal{X} \rightarrow \mathcal{Y}$ 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** $f : \mathcal{X} \rightarrow \mathcal{Y}$ 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.

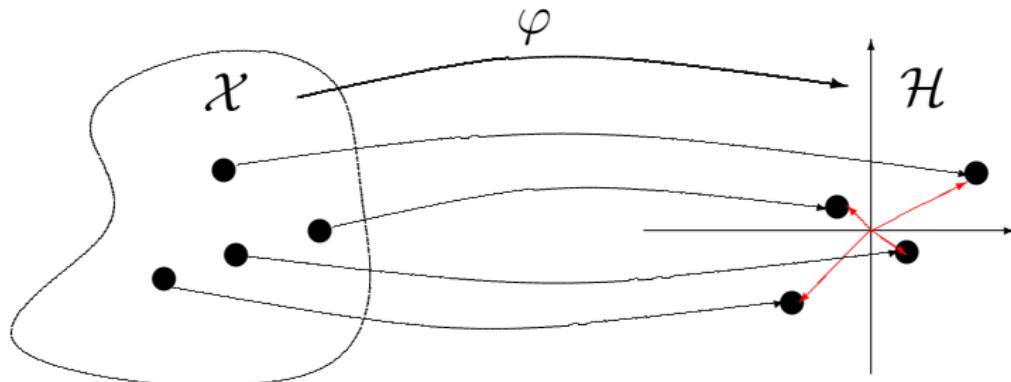
- assume there exists a linear relation between y and features \mathbf{x} in \mathbb{R}^p .
- $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ is parametrized by \mathbf{w}, b in \mathbb{R}^{p+1} ;
- L is often a **convex** loss function;
- $\Omega(f)$ is often the squared ℓ_2 -norm $\|\mathbf{w}\|^2$.

Motivation from supervised learning

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

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

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



Motivation from supervised learning

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

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

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

First purpose: embed data in a vectorial space where

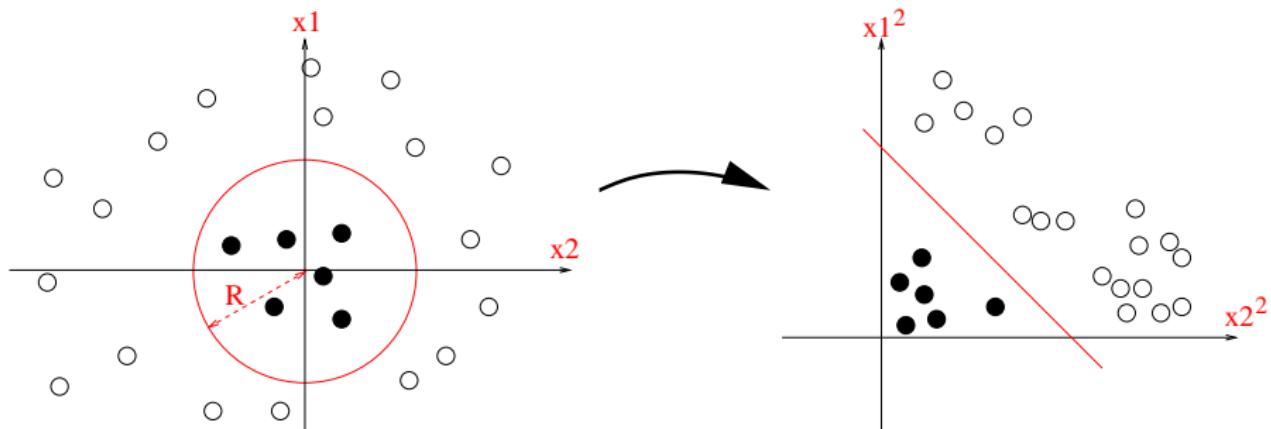
- many **geometrical operations** exist (angle computation, projection on linear subspaces, definition of barycenters....).
- one may learn potentially **rich infinite-dimensional models**.
- **regularization** is natural and theoretically grounded.

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

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
 - The kernel trick
 - The representer theorem
 - Kernel ridge regression
 - Kernel logistic regression
 - Kernel PCA
- 3 Kernels and Graphs
- 4 Characterizing probabilities with kernels
- 5 Open Problems and Research Topics

The kernel trick

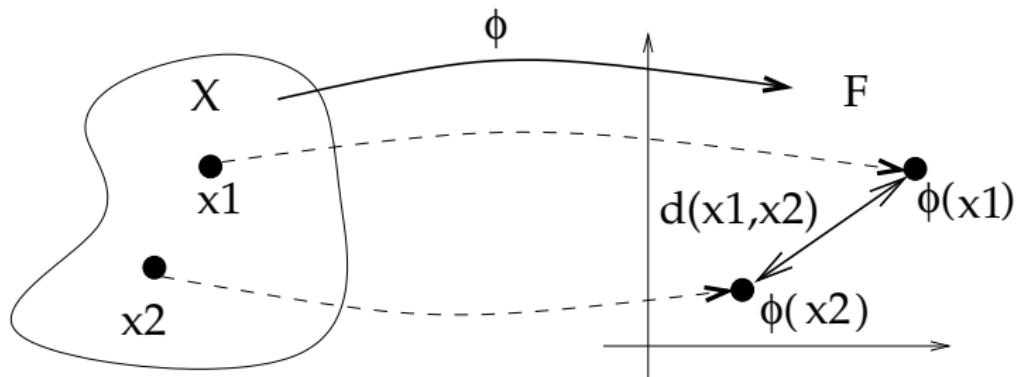
Proposition

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



$$\begin{aligned} d_K(x_1, x_2)^2 &= \| \Phi(x_1) - \Phi(x_2) \|_{\mathcal{H}}^2 \\ &= \langle \Phi(x_1) - \Phi(x_2), \Phi(x_1) - \Phi(x_2) \rangle_{\mathcal{H}} \\ &= \langle \Phi(x_1), \Phi(x_1) \rangle_{\mathcal{H}} + \langle \Phi(x_2), \Phi(x_2) \rangle_{\mathcal{H}} - 2 \langle \Phi(x_1), \Phi(x_2) \rangle_{\mathcal{H}} \\ d_K(x_1, x_2)^2 &= K(x_1, x_1) + K(x_2, x_2) - 2K(x_1, x_2) \end{aligned}$$

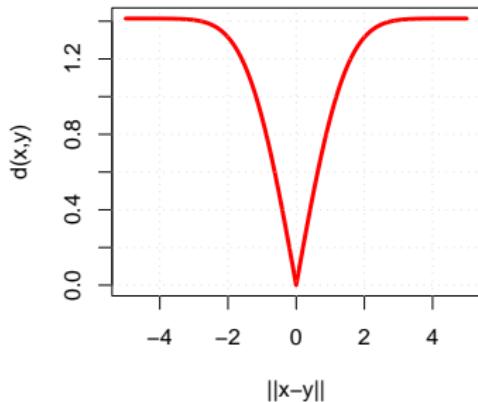
Distance for the Gaussian kernel

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

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

- $K(\mathbf{x}, \mathbf{x}) = 1 = \|\Phi(\mathbf{x})\|_{\mathcal{H}}^2$, 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 $\mathcal{S} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ 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 $\mathcal{S} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ 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:

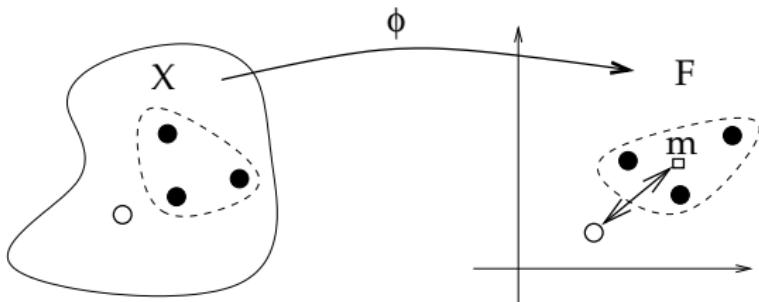
- Map all points to the feature space.
- Summarize \mathcal{S} by the **barycenter** of the points:

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

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

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

Computation



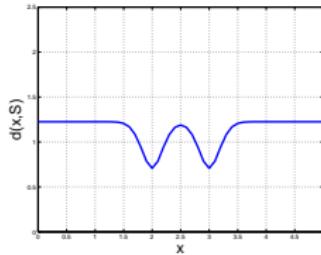
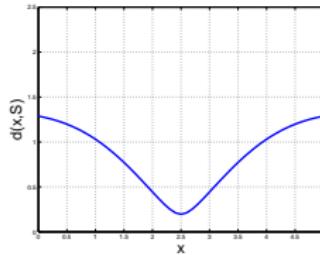
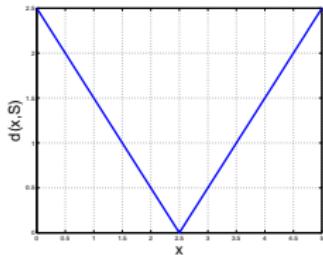
$$\begin{aligned} d_K(x, \mathcal{S}) &= \left\| \phi(x) - \frac{1}{n} \sum_{i=1}^n \phi(x_i) \right\|_{\mathcal{H}} \\ &= \sqrt{K(x, x) - \frac{2}{n} \sum_{i=1}^n K(x, x_i) + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n K(x_i, x_j)}. \end{aligned}$$

Remark

The barycentre μ only exists in the feature space in general: it does not necessarily have a pre-image x_μ such that $\Phi(x_\mu) = \mu$.

1D illustration

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



$$K(x, y) = xy.$$

(linear)

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

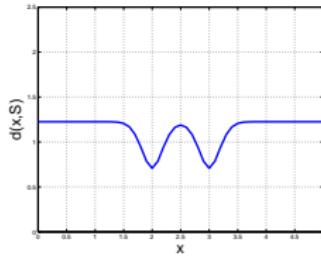
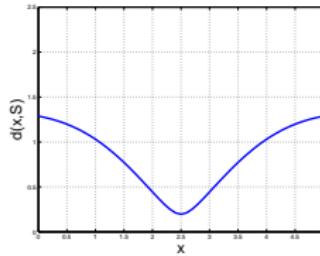
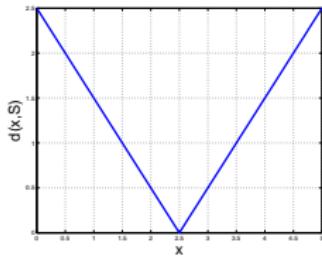
with $\sigma = 1$.

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

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 $\sigma = 1$.

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

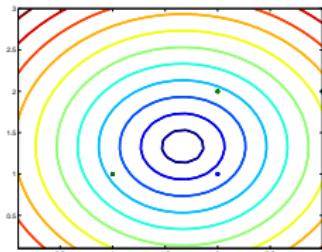
with $\sigma = 0.2$.

Remarks

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

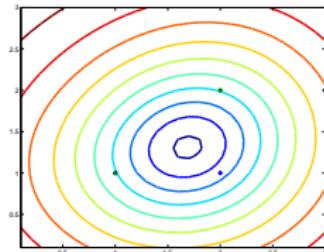
2D illustration

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



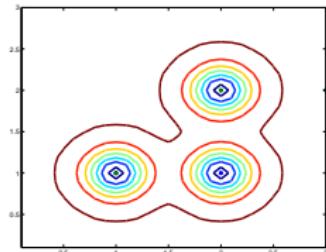
$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}\mathbf{y}.$$

(linear)



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

with $\sigma = 1$.

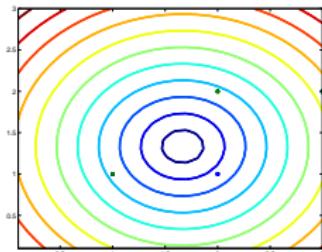


$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$$

with $\sigma = 0.2$.

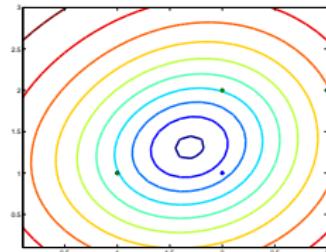
2D illustration

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



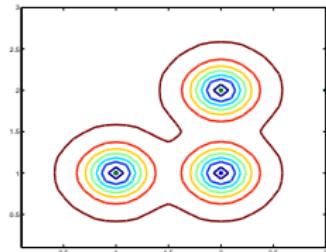
$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}\mathbf{y}.$$

(linear)



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

with $\sigma = 1$.



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

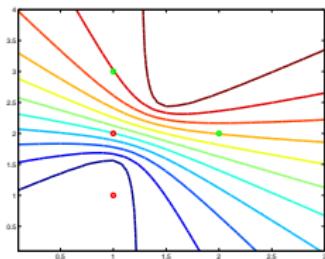
with $\sigma = 0.2$.

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 $f(\mathbf{x}) = d(\mathbf{x}, \mathcal{S}_1)^2 - d(\mathbf{x}, \mathcal{S}_2)^2$

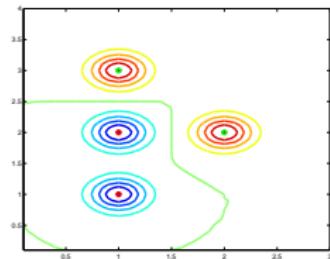


$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}\mathbf{y}.$$

(linear)

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

with $\sigma = 1$.



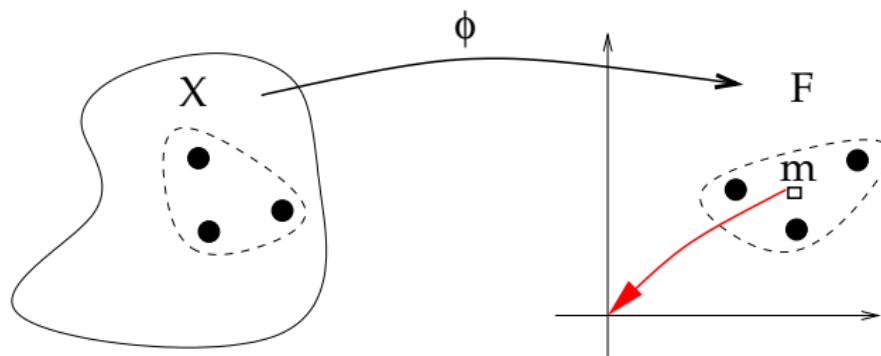
$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$$

with $\sigma = 0.2$.

Example 3: Centering data in the feature space

Problem

- Let $\mathcal{S} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ 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 $i = 1, \dots, n$ 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 $0 \leq i, j \leq n$:

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

- 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

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
 - The kernel trick
 - The representer theorem
 - Kernel ridge regression
 - Kernel logistic regression
 - Kernel PCA
- 3 Kernels and Graphs
- 4 Characterizing probabilities with kernels
- 5 Open Problems and Research Topics

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 $f : \mathcal{X} \rightarrow \mathbb{R}$ is to solve something like:

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

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 $\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subseteq \mathcal{X}$ a finite set of points in \mathcal{X} .
- Let $\Psi : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ be a function of $n + 1$ variables, strictly increasing with respect to the last variable.
- Then, any solution to the optimization problem:

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

admits a representation of the form:

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

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

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

Proof (1/2)

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

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

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

$$f = f_S + f_{\perp},$$

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

Proof (2/2)

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

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

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

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

- 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:

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

where $c(\cdot)$ 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 $j = 1, \dots, n$, 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

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

is equivalent to the following n -dimensional optimization problem:

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

- 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.

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 $h : \mathcal{X} \rightarrow \mathcal{Y}$ 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 $h : \mathcal{X} \rightarrow \mathcal{Y}$ to **predict** the output for any future input.

Depending on the nature of the output, this covers:

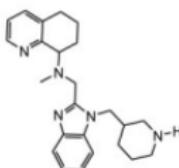
- **Regression** when $\mathcal{Y} = \mathbb{R}$;
- **Classification** when $\mathcal{Y} = \{-1, 1\}$ or any set of two labels;
- **Structured output** regression or classification when \mathcal{Y} is more general.

Example: regression

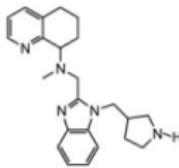
Task: predict the capacity of a small molecule to inhibit a drug target

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

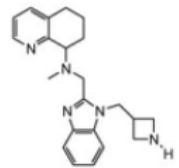
$\mathcal{Y} = \mathbb{R}$



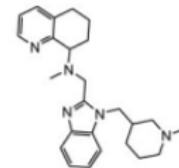
$IC_{50} = 51 \pm 2 \text{nM}$



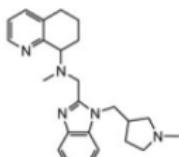
$IC_{50} = 24 \pm 1 \text{nM}$



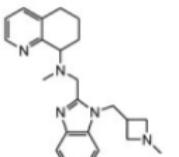
$IC_{50} = 84 \pm 7 \text{nM}$



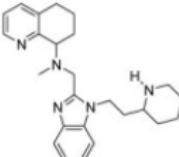
$IC_{50} = 19 \pm 2 \text{nM}$



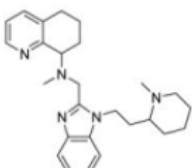
$IC_{50} = 57 \pm 2 \text{nM}$



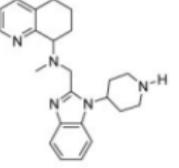
$IC_{50} = 66 \pm 5 \text{nM}$



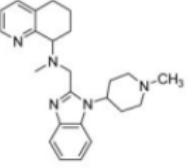
$IC_{50} = 64 \pm 4 \text{nM}$



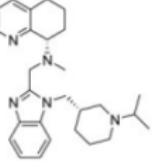
$IC_{50} = 33 \pm 1 \text{nM}$



$IC_{50} = 99 \pm 6 \text{nM}$



$IC_{50} = 95 \pm 8 \text{nM}$



$IC_{50} = 2.0 \pm 1 \text{nM}$

Example: classification

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

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

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



Example: classification

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

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

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



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

The screenshot shows a web browser window for translate.google.fr. The interface includes a top navigation bar with tabs, a search bar, and various icons. Below the bar, the Google logo is visible. The main area is titled "Traduction". On the left, there is a text input field containing the Japanese sentence "猿も木から落ちる" (Sarumokikaraochiru). On the right, the corresponding French translation "Même les singes tombent des arbres" is displayed. The bottom of the page features standard footer links for Google Translate and other Google services.

translate.google.fr

Google

Traduction

Désactiver la traduction instantanée

Anglais Français Arabe Japonais - détecté

Traduire

猿も木から落ちる

Même les singes tombent des arbres

Sarumokikaraochiru

À propos de Google Traduction Communauté Mobile G+ B

À propos de Google Confidentialité et conditions d'utilisation Aide Envoyer des commentaires

83 / 341

Supervised learning with kernels: general principles

- ① Express $h : \mathcal{X} \rightarrow \mathcal{Y}$ using a real-valued function $f : \mathcal{Z} \rightarrow \mathbb{R}$:

- regression $\mathcal{Y} = \mathbb{R}$:

$$h(\mathbf{x}) = f(\mathbf{x}) \quad \text{with} \quad f : \mathcal{X} \rightarrow \mathbb{R} \quad (\mathcal{Z} = \mathcal{X})$$

- classification $\mathcal{Y} = \{-1, 1\}$:

$$h(\mathbf{x}) = \text{sign}(f(\mathbf{x})) \quad \text{with} \quad f : \mathcal{X} \rightarrow \mathbb{R} \quad (\mathcal{Z} = \mathcal{X})$$

- structured output:

$$h(\mathbf{x}) = \arg \max_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y}) \quad \text{with} \quad f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R} \quad (\mathcal{Z} = \mathcal{X} \times \mathcal{Y})$$

- 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:

$$R_n(f) := \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i), \mathbf{y}_i)$$

- 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

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

- Regularization is important, particularly in high dimension, to prevent **overfitting**
- When $\mathcal{Z} = \mathbb{R}^d$ and K is the linear kernel, $f = f_w$ is a linear model and the regularization is $\|w\|^2$
- 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

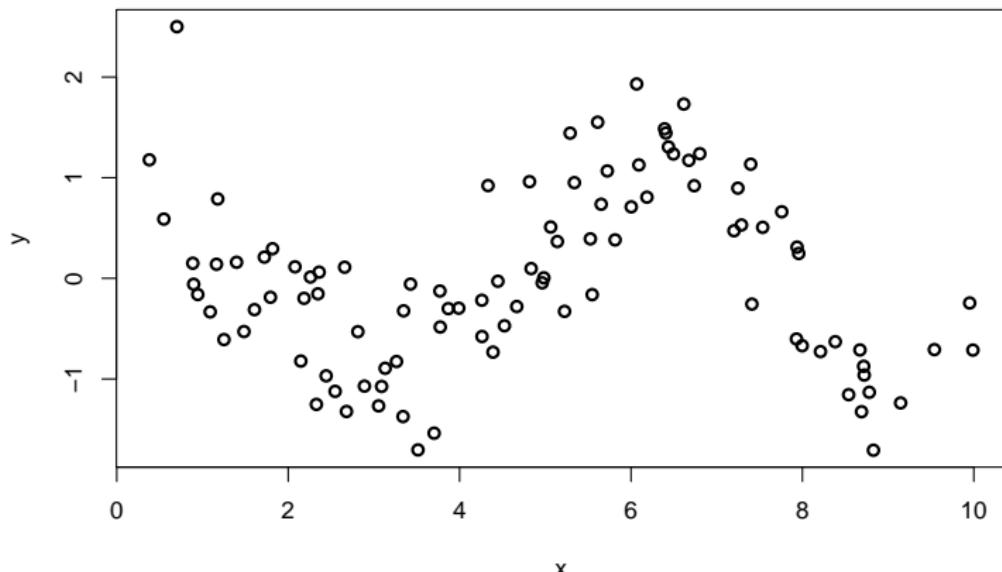
Outline

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
 - The kernel trick
 - The representer theorem
 - Kernel ridge regression
 - Kernel logistic regression
 - Kernel PCA
- 3 Kernels and Graphs
- 4 Characterizing probabilities with kernels
- 5 Open Problems and Research Topics

Regression

Setup

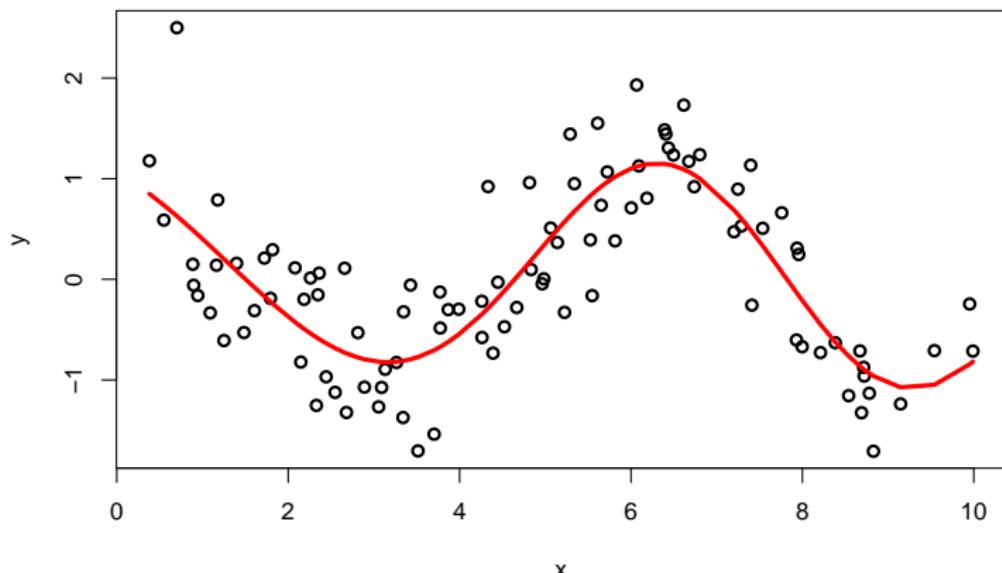
- \mathcal{X} set of inputs
- $\mathcal{Y} = \mathbb{R}$ real-valued outputs
- $\mathcal{S}_n = (\mathbf{x}_i, y_i)_{i=1, \dots, n} \in (\mathcal{X} \times \mathbb{R})^n$ a training set of n pairs
- Goal = find a function $f : \mathcal{X} \rightarrow \mathbb{R}$ to predict y by $f(\mathbf{x})$



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 $f : \mathcal{X} \rightarrow \mathbb{R}$ 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:

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

- 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:

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

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

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

- 2nd effect = simplifying the solution.*

Solving KRR

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

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

- 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

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

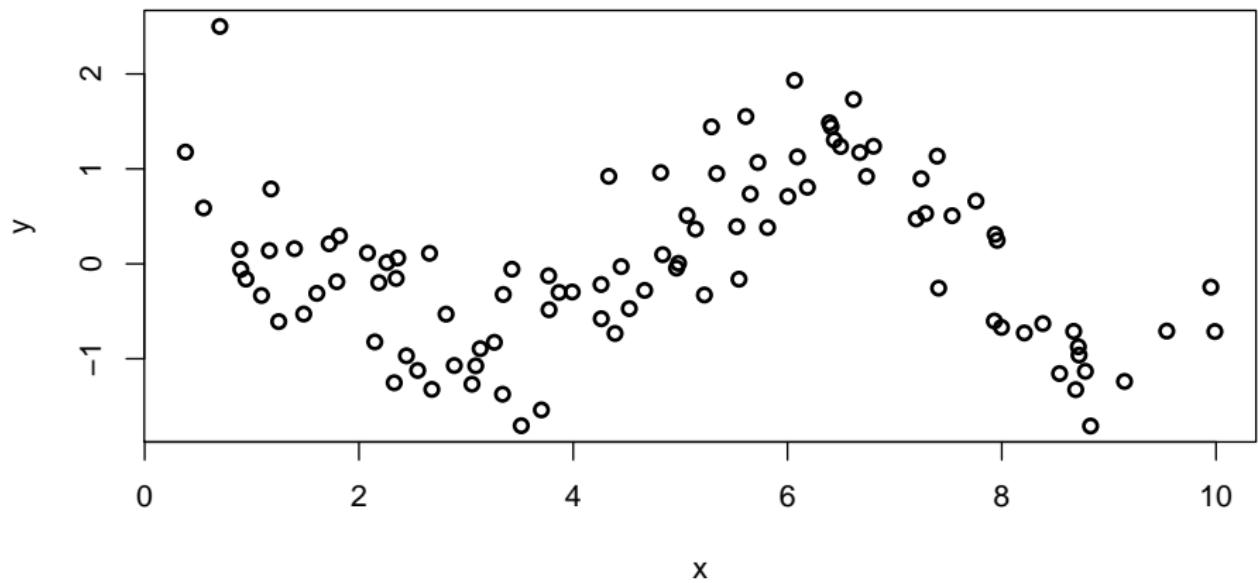
- This is a convex and differentiable function of α . Its minimum can therefore be found by setting the gradient in α to zero:

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

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

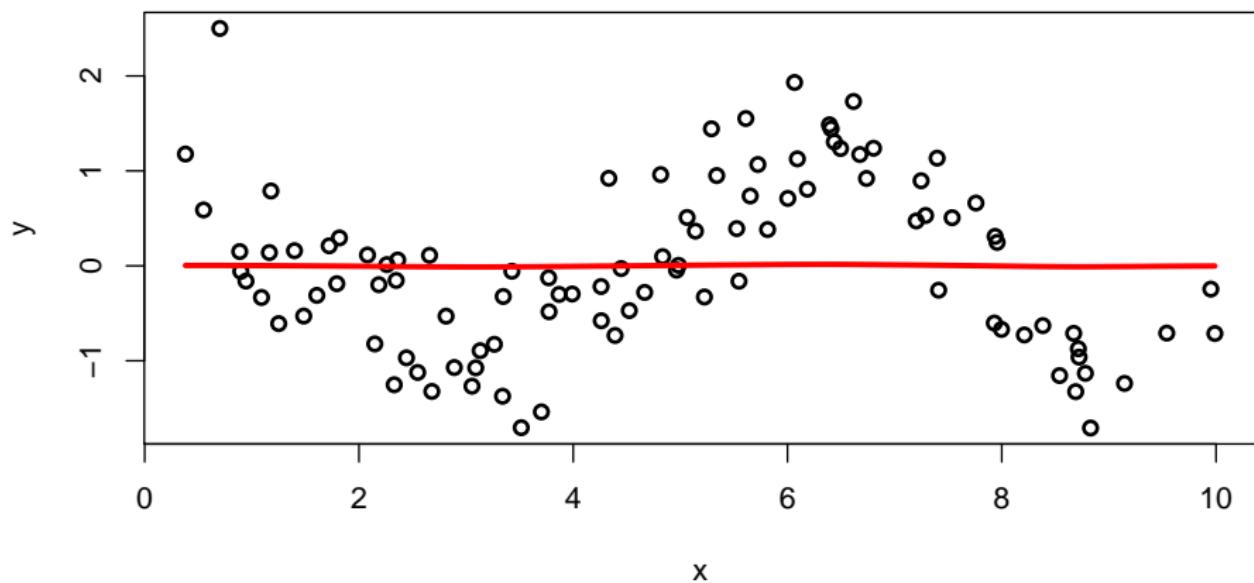
$$\alpha = (\mathbf{K} + \lambda n \mathbf{I})^{-1} \mathbf{y}.$$

Example (KRR with Gaussian RBF kernel)



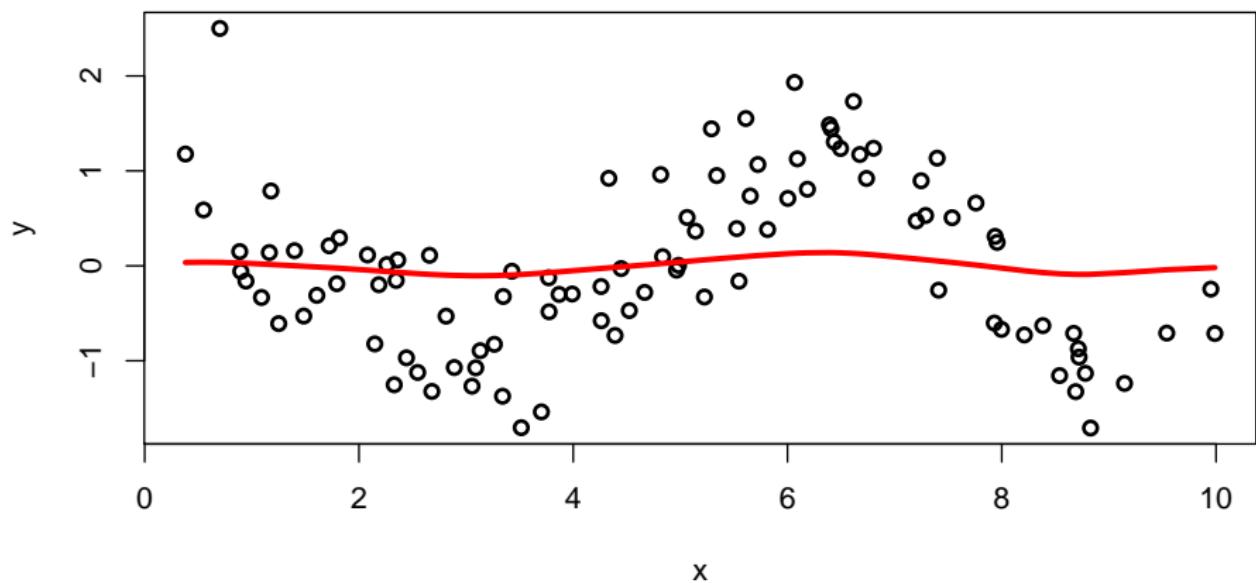
Example (KRR with Gaussian RBF kernel)

lambda = 1000



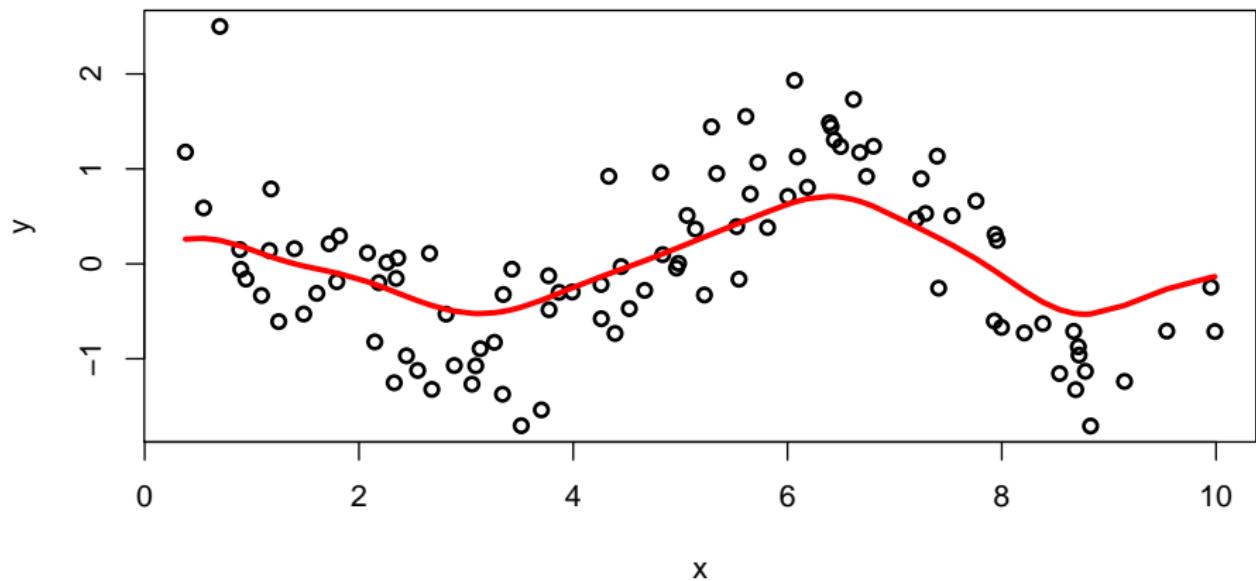
Example (KRR with Gaussian RBF kernel)

lambda = 100



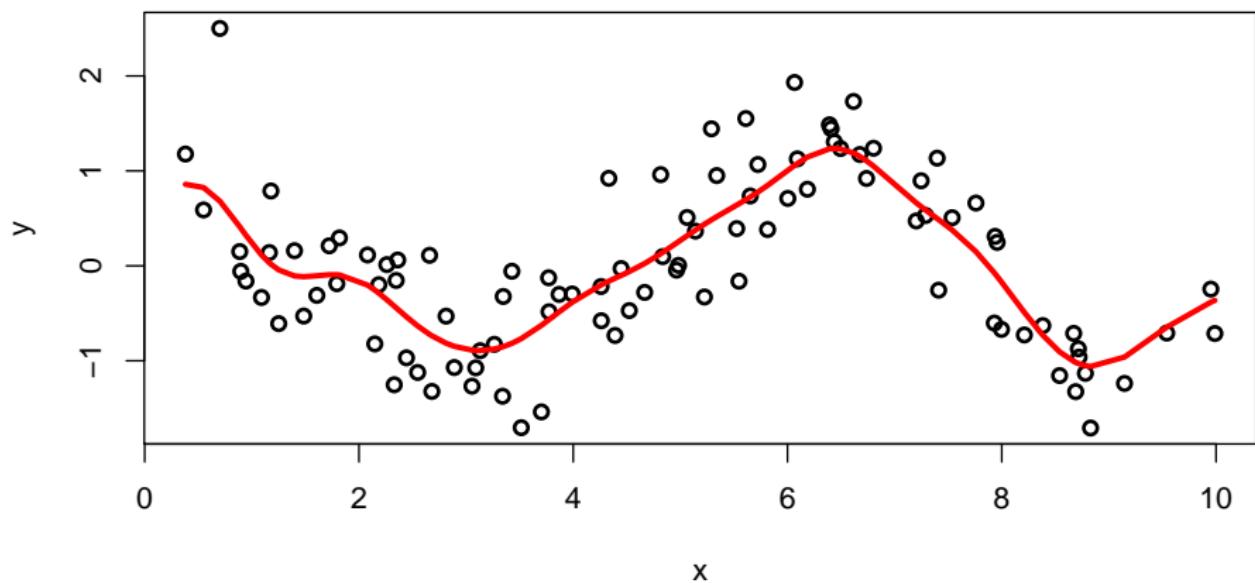
Example (KRR with Gaussian RBF kernel)

lambda = 10



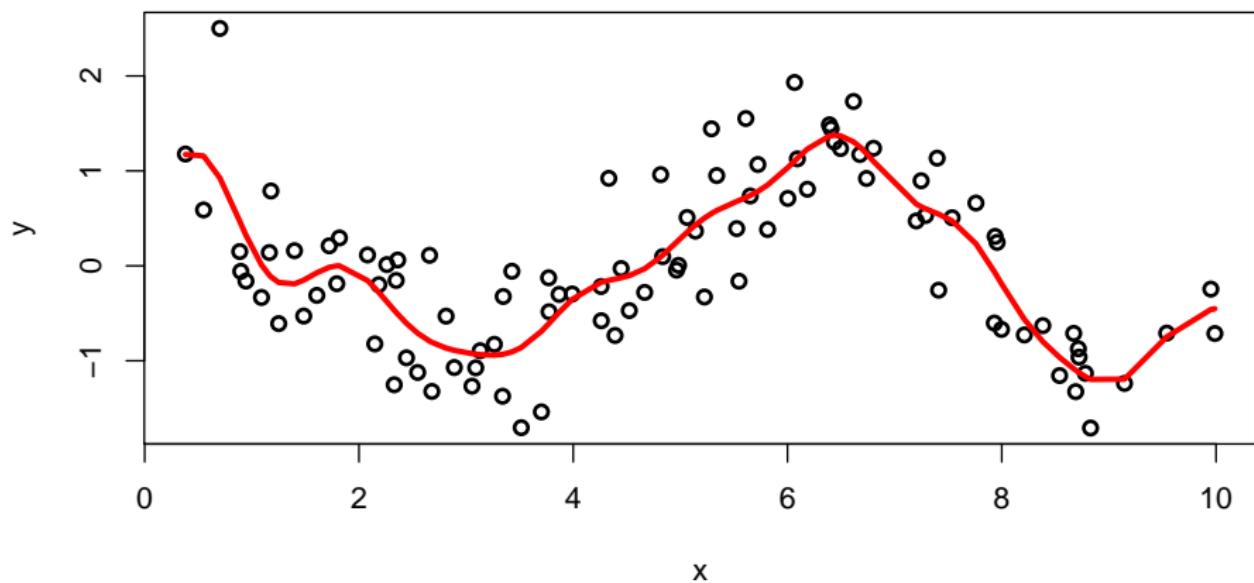
Example (KRR with Gaussian RBF kernel)

lambda = 1



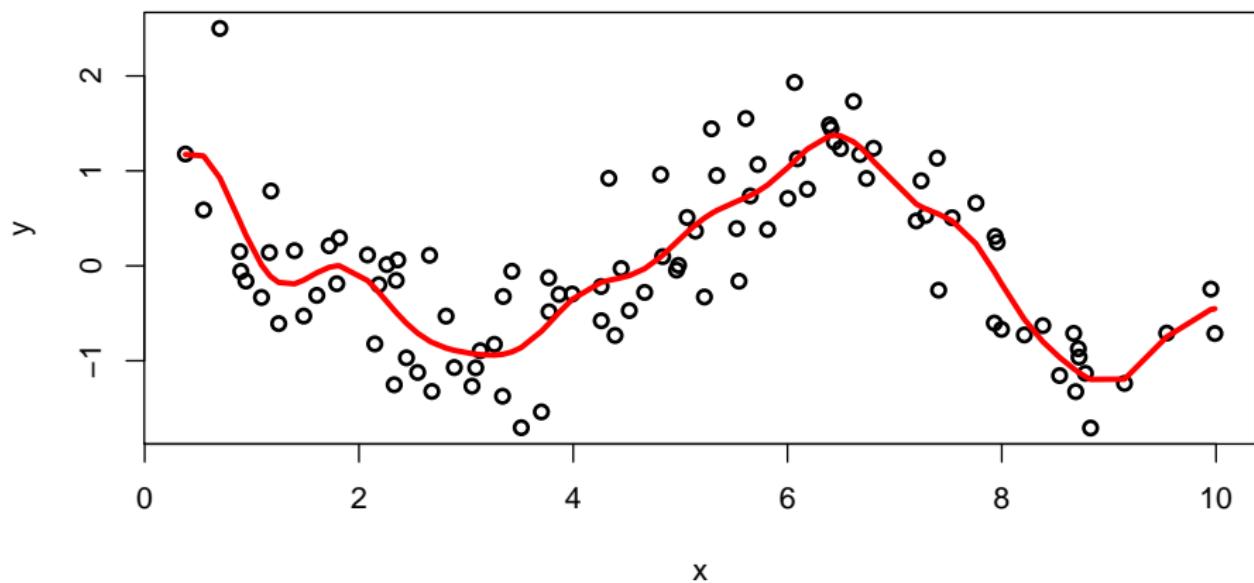
Example (KRR with Gaussian RBF kernel)

lambda = 0.1



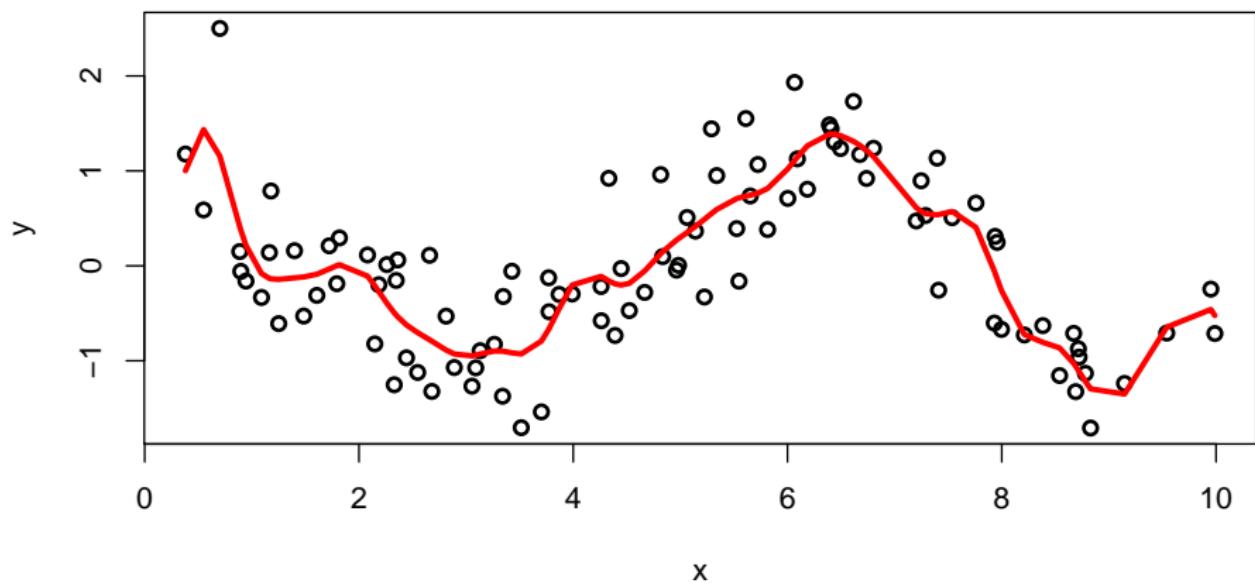
Example (KRR with Gaussian RBF kernel)

lambda = 0.01



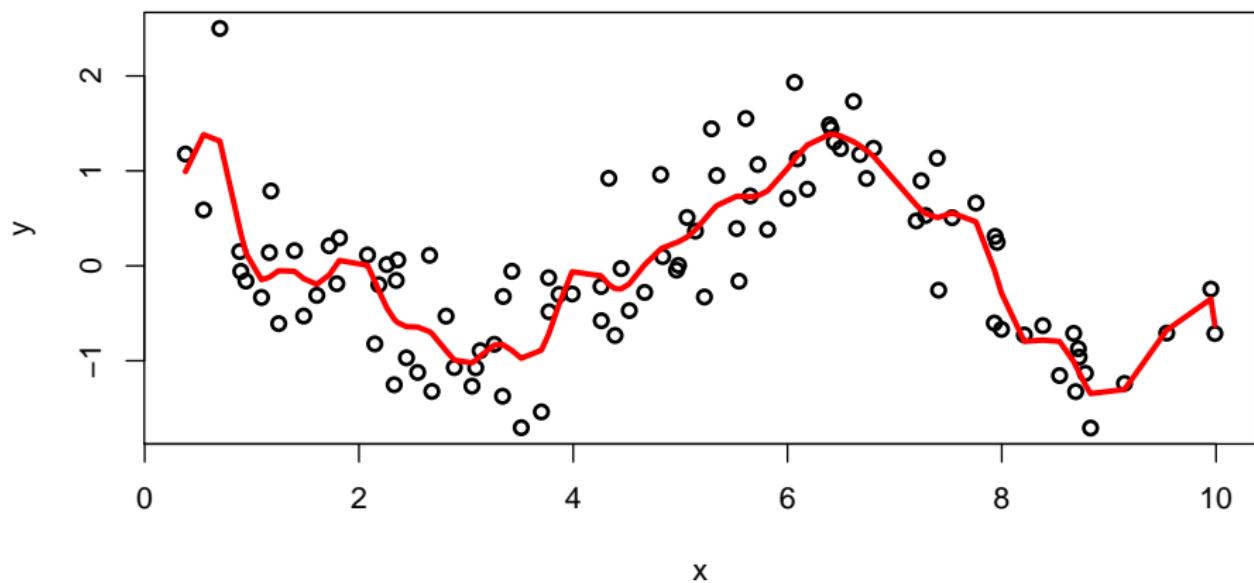
Example (KRR with Gaussian RBF kernel)

lambda = 0.001



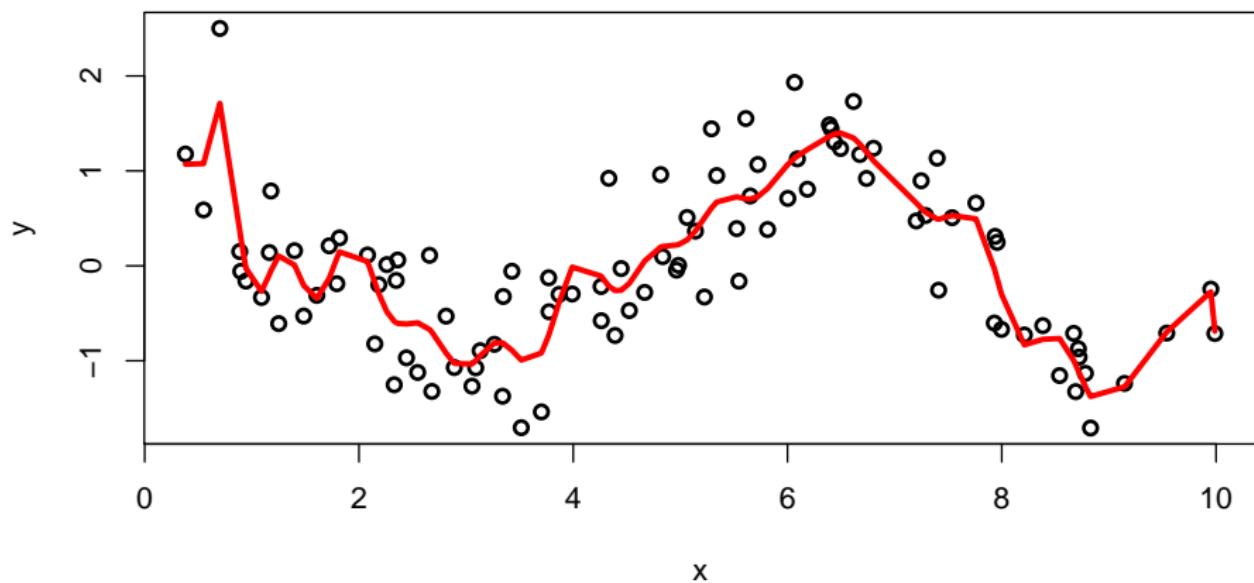
Example (KRR with Gaussian RBF kernel)

lambda = 0.0001



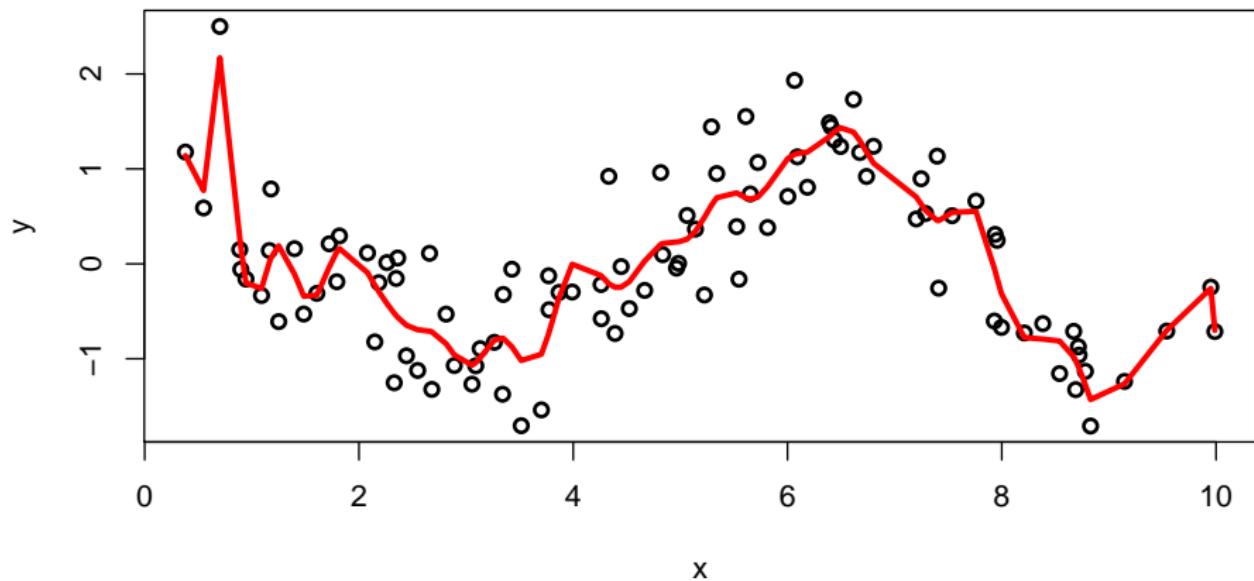
Example (KRR with Gaussian RBF kernel)

lambda = 0.00001



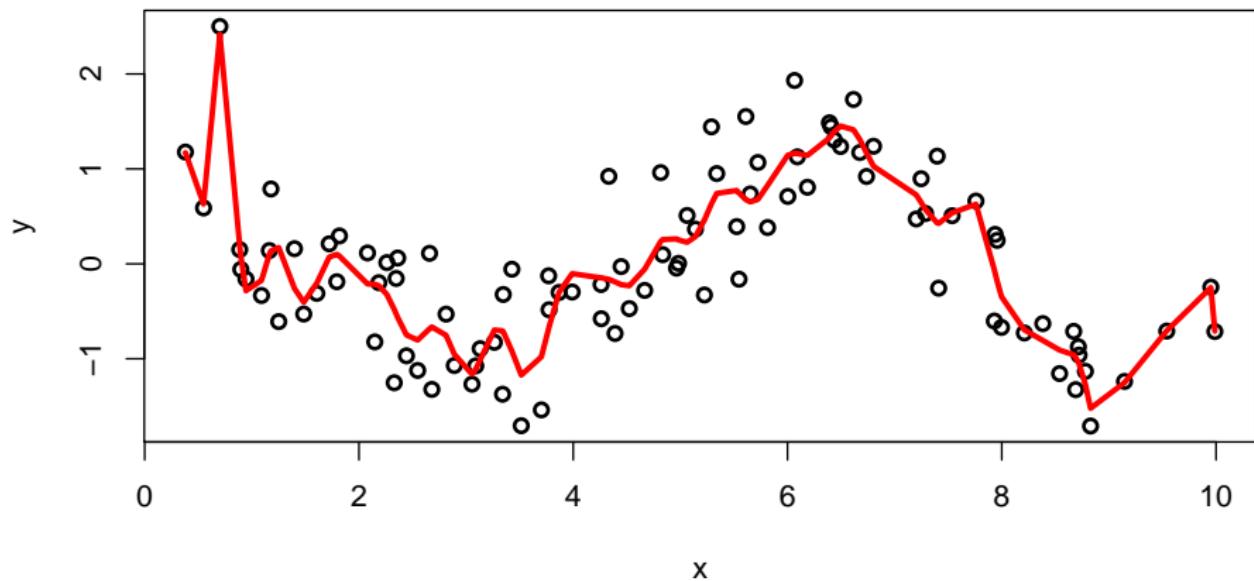
Example (KRR with Gaussian RBF kernel)

lambda = 0.000001



Example (KRR with Gaussian RBF kernel)

lambda = 0.0000001



Remark: uniqueness of the solution

Let us find *all* α 's that solve

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

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

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

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

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

therefore $f = f'$. KRR has a unique solution $f \in \mathcal{H}$, which can possibly be expressed by several α '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 $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$ 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

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

- Setting the gradient to 0 gives the solution:

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

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

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):

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

We deduce that (of course...):

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

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):

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

We deduce that (of course...):

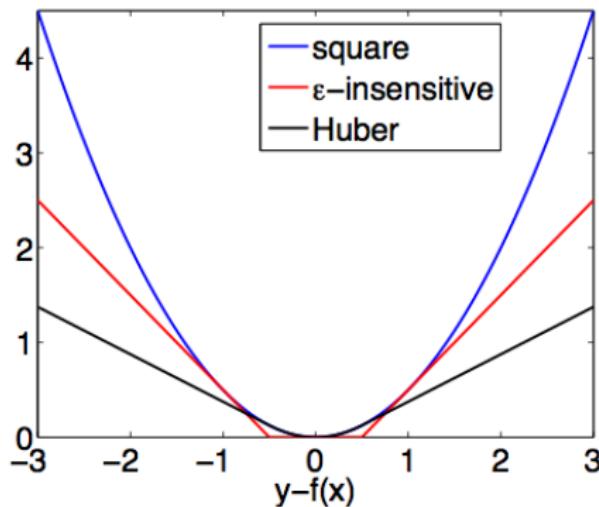
$$\mathbf{w}_{RR} = \underbrace{\left(\mathbf{X}^\top \mathbf{X} + \lambda n I\right)^{-1}}_{d \times d} \mathbf{X}^\top \mathbf{y} = \mathbf{X}^\top \underbrace{\left(\mathbf{X} \mathbf{X}^\top + \lambda n I\right)^{-1}}_{n \times n} \mathbf{y} = \mathbf{w}_{KRR}$$

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

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

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

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

Weighted regression

- Setting the gradient to zero gives

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

- 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

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
 - The kernel trick
 - The representer theorem
 - Kernel ridge regression
 - **Kernel logistic regression**
 - Kernel PCA
- 3 Kernels and Graphs
- 4 Characterizing probabilities with kernels
- 5 Open Problems and Research Topics

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 $f : \mathcal{X} \rightarrow \mathbb{R}$ to predict y by $\text{sign}(f(\mathbf{x}))$



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 $f : \mathcal{X} \rightarrow \mathbb{R}$ to predict y by $\text{sign}(f(\mathbf{x}))$



The 0/1 loss

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

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

- It is then tempting to learn f by solving:

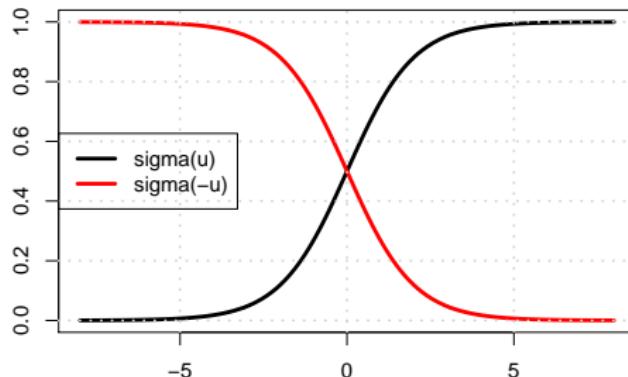
$$\min_{f \in \mathcal{H}} \underbrace{\frac{1}{n} \sum_{i=1}^n \ell_{0/1}(f(\mathbf{x}_i), y_i)}_{\text{misclassification rate}} + \underbrace{\lambda \|f\|_{\mathcal{H}}^2}_{\text{regularization}}$$

- 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.:

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



- The **logistic loss** is the negative conditional likelihood:

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

Kernel logistic regression (KLR)

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

- 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

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

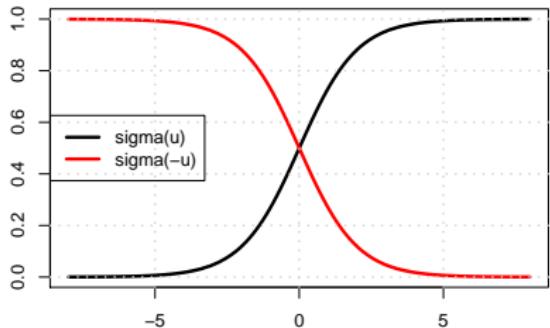
and as always we have:

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

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

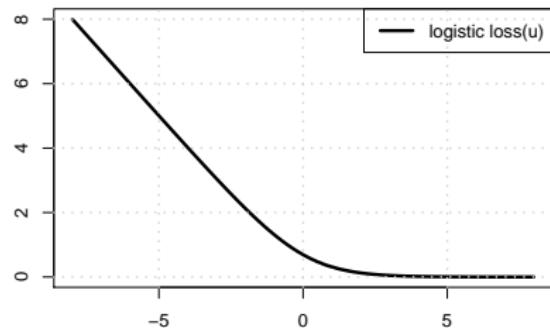
$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ln \left(1 + e^{-y_i [\mathbf{K}\boldsymbol{\alpha}]_i} \right) + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

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:

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

Back to KLR

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

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 α_0 is the function:

$$J_q(\alpha) = J(\alpha_0) + (\alpha - \alpha_0)^\top \nabla J(\alpha_0) + \frac{1}{2} (\alpha - \alpha_0)^\top \nabla^2 J(\alpha_0) (\alpha - \alpha_0)$$

Let us compute the different terms...

Computing the quadratic approximation

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

therefore

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

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

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

therefore

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

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

Computing the quadratic approximation

$$J_q(\alpha) = J(\alpha_0) + (\alpha - \alpha_0)^\top \nabla J(\alpha_0) + \frac{1}{2} (\alpha - \alpha_0)^\top \nabla^2 J(\alpha_0) (\alpha - \alpha_0)$$

Terms that depend on α , with $\mathbf{P} = \mathbf{P}(\alpha_0)$ and $\mathbf{W} = \mathbf{W}(\alpha_0)$:

- $\alpha^\top \nabla J(\alpha_0) = \frac{1}{n} \alpha^\top \mathbf{K} \mathbf{P} \mathbf{y} + \lambda \alpha^\top \mathbf{K} \alpha_0$
- $\frac{1}{2} \alpha^\top \nabla^2 J(\alpha_0) \alpha = \frac{1}{2n} \alpha^\top \mathbf{K} \mathbf{W} \mathbf{K} \alpha + \frac{\lambda}{2} \alpha^\top \mathbf{K} \alpha$
- $-\alpha^\top \nabla^2 J(\alpha_0) \alpha_0 = -\frac{1}{n} \alpha^\top \mathbf{K} \mathbf{W} \mathbf{K} \alpha_0 - \lambda \alpha^\top \mathbf{K} \alpha_0$

Putting it all together:

$$\begin{aligned} 2J_q(\alpha) &= -\frac{2}{n} \alpha^\top \mathbf{K} \mathbf{W} \underbrace{(\mathbf{K} \alpha_0 - \mathbf{W}^{-1} \mathbf{P} \mathbf{y})}_{:= \mathbf{z}} + \frac{1}{n} \alpha^\top \mathbf{K} \mathbf{W} \mathbf{K} \alpha + \lambda \alpha^\top \mathbf{K} \alpha + C \\ &= \frac{1}{n} (\mathbf{K} \alpha - \mathbf{z})^\top \mathbf{W} (\mathbf{K} \alpha - \mathbf{z}) + \lambda \alpha^\top \mathbf{K} \alpha + C \end{aligned}$$

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:

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

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

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

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

Kernel Methods

Unsupervised Learning

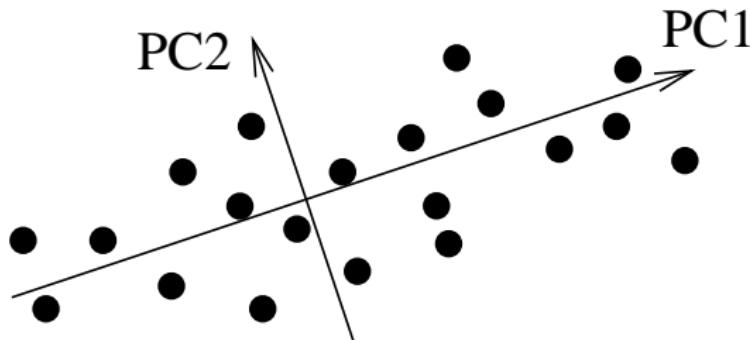
Outline

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
 - The kernel trick
 - The representer theorem
 - Kernel ridge regression
 - Kernel logistic regression
 - Kernel PCA
- 3 Kernels and Graphs
- 4 Characterizing probabilities with kernels
- 5 Open Problems and Research Topics

Principal Component Analysis (PCA)

Classical setting

- Let $\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ 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 $h_{\mathbf{w}} : \mathbb{R}^d \rightarrow \mathbb{R}$ 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:

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

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

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

Principal Component Analysis (PCA)

Solution

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

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

- The solutions of:

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

Principal Component Analysis (PCA)

Solution

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

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

- The solutions of:

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

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

Kernel Principal Component Analysis (PCA)

Let $\mathbf{x}_1, \dots, \mathbf{x}_n$ be a set of data points in \mathcal{X} ; let $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ 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.:

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

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

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

Kernel Principal Component Analysis (PCA)

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

Formalization

- The empirical variance captured by h_f is:

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

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

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

Kernel Principal Component Analysis (PCA)

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

Formalization

- The empirical variance captured by h_f is:

$$\hat{\text{var}}(h_w) = \frac{1}{n} \sum_{i=1}^n \frac{(\mathbf{x}_i^\top \mathbf{w})^2}{\|\mathbf{w}\|^2} \implies \hat{\text{var}}(h_f) := \frac{1}{n} \sum_{i=1}^n \frac{f(\mathbf{x}_i)^2}{\|f\|_{\mathcal{H}}^2}.$$

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

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

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:

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

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

Kernel Principal Component Analysis (PCA)

Solution

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

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

- We can apply the representer theorem (*exercise: check that is is also valid in this case*): for $i = 1, \dots, d$, 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 $\boldsymbol{\alpha}_i = (\alpha_{i,1}, \dots, \alpha_{i,n})^\top \in \mathbb{R}^n$.

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 α the function:

$$\alpha_i = \arg \max_{\alpha \in \mathbb{R}^n} \alpha^\top \mathbf{K}^2 \alpha,$$

under the constraints:

$$\begin{cases} \alpha_i^\top \mathbf{K} \alpha_j &= 0 \quad \text{for } j = 1, \dots, i-1. \\ \alpha_i^\top \mathbf{K} \alpha_i &= 1 \end{cases}$$

Kernel Principal Component Analysis (PCA)

Solution

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

$$\beta_i = \arg \max_{\beta \in \mathbb{R}^n} \beta^\top \mathbf{K} \beta,$$

under the constraints:

$$\begin{cases} \beta_i^\top \beta_j = 0 & \text{for } j = 1, \dots, i-1. \\ \beta_i^\top \beta_i = 1 \end{cases}$$

- Thus, $\beta_i = \mathbf{u}_i$ (i -th eigenvector) is a solution!
- Finally, $\alpha_i = \frac{1}{\sqrt{\Delta_i}} \mathbf{u}_i$.

Kernel Principal Component Analysis (PCA)

Summary

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

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...

Kernels for graphs and Kernels on Graphs

Outline

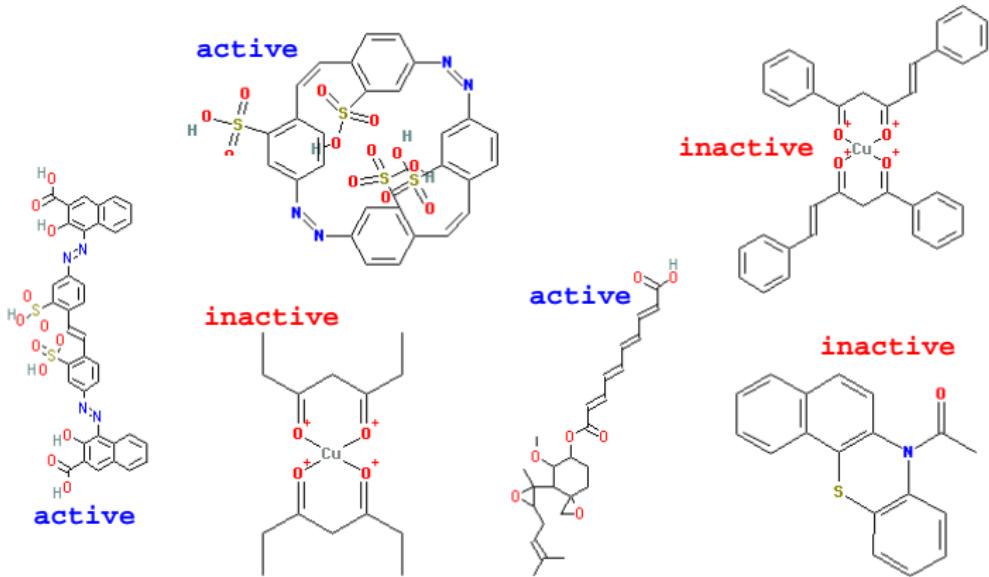
- 1 Kernels and RKHS
- 2 Kernel tricks and applications
- 3 Kernels and Graphs
 - Kernels for graphs
 - Kernels on graphs
- 4 Characterizing probabilities with kernels
- 5 Open Problems and Research Topics

Outline

3 Kernels and Graphs

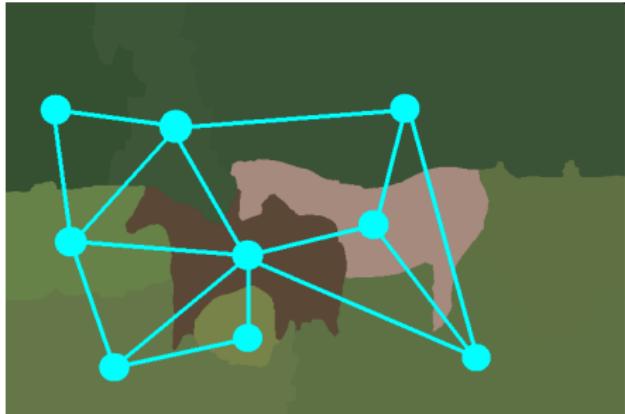
- Kernels for graphs
 - Motivation
 - Explicit enumeration of features
 - Implicit enumeration of features
 - Walk-based kernels
- Kernels on graphs

Virtual screening for drug discovery



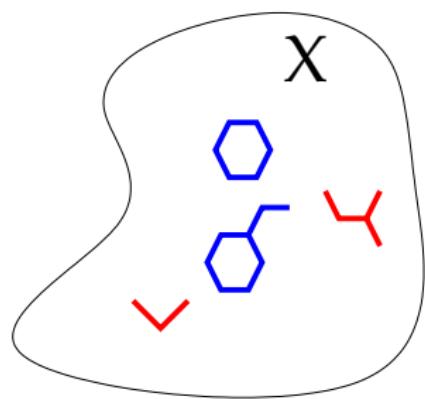
NCI AIDS screen results (from <http://cactus.nci.nih.gov>).

Image retrieval and classification



From Harchaoui and Bach (2007).

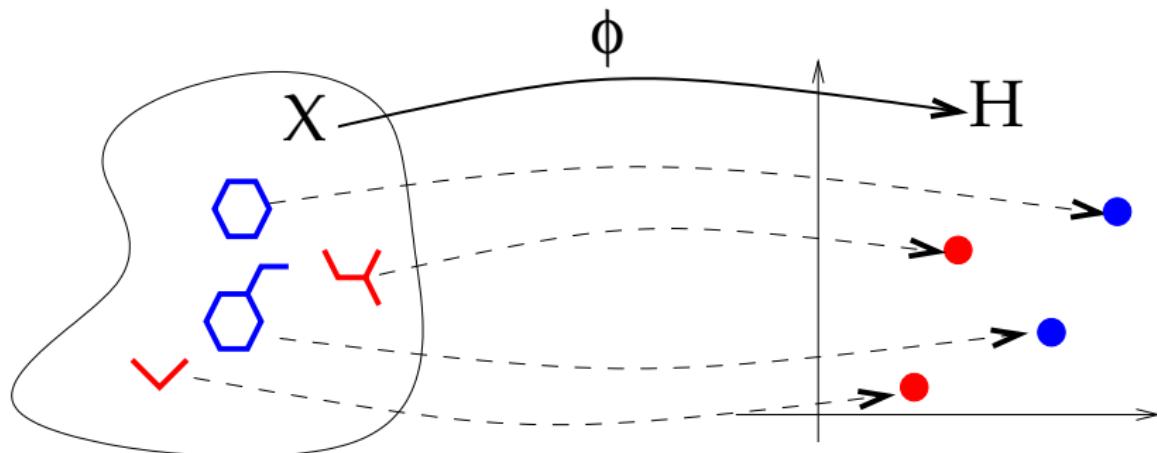
Our approach



Our approach

- ① Represent each graph \mathbf{x} in \mathcal{X} by a vector $\Phi(\mathbf{x}) \in \mathcal{H}$, either **explicitly** or **implicitly** through the kernel

$$K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^\top \Phi(\mathbf{x}').$$

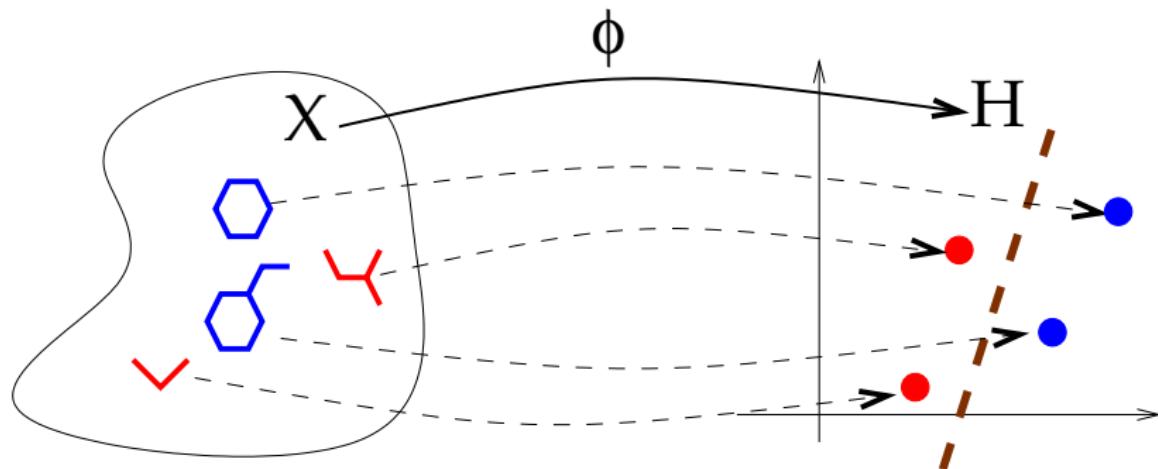


Our approach

- ① Represent each graph \mathbf{x} in \mathcal{X} by a vector $\Phi(\mathbf{x}) \in \mathcal{H}$, either **explicitly** or **implicitly** through the kernel

$$K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^\top \Phi(\mathbf{x}').$$

- ② Use a linear method for classification in \mathcal{H} .



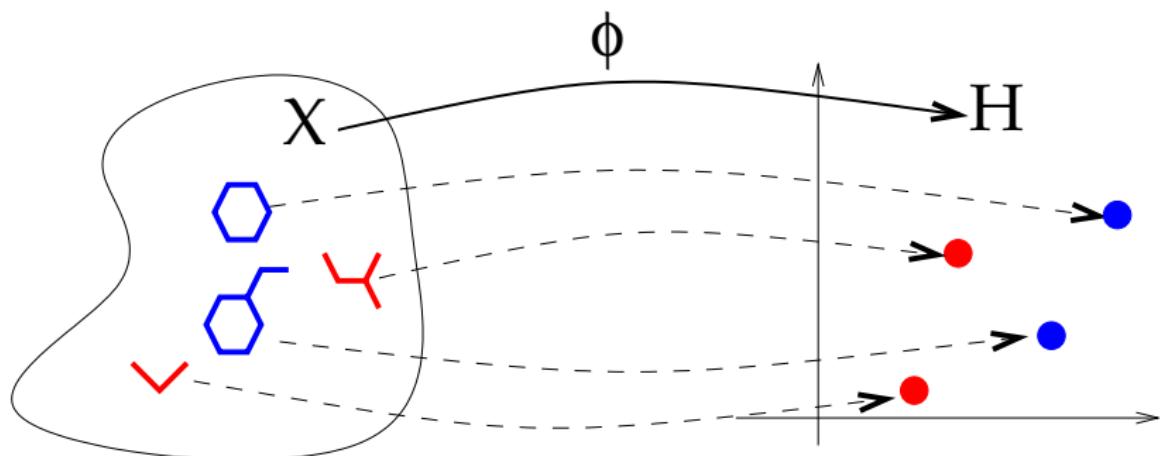
Outline

3 Kernels and Graphs

- Kernels for graphs
 - Motivation
 - Explicit enumeration of features
 - Implicit enumeration of features
 - Walk-based kernels
- Kernels on graphs

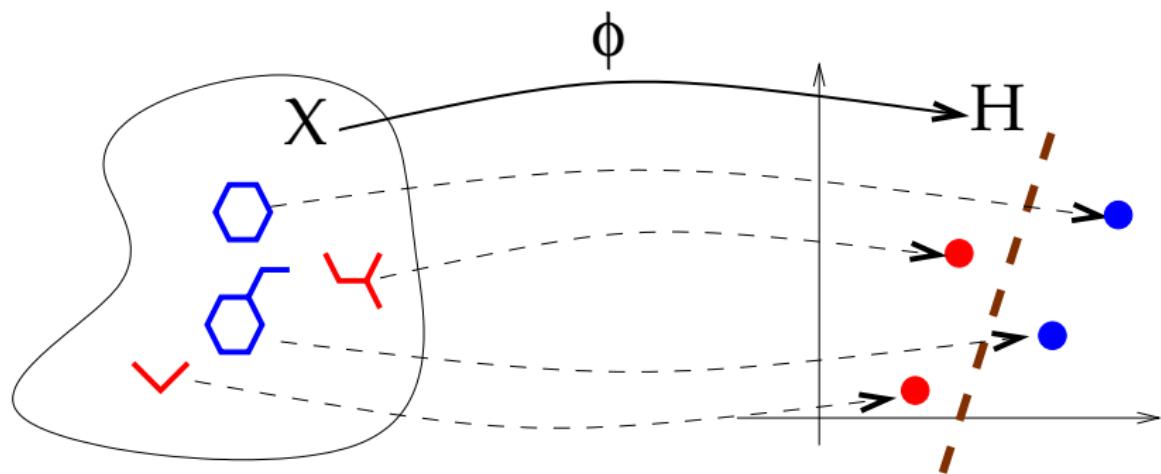
The approach

- ① Represent explicitly each graph x by a **vector of fixed dimension** $\Phi(x) \in \mathbb{R}^p$.



The approach

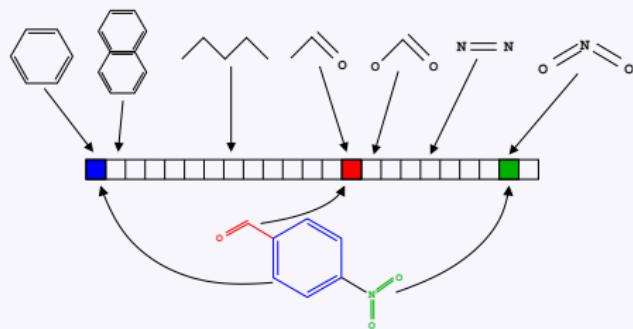
- ① Represent explicitly each graph x by a vector of fixed dimension $\Phi(x) \in \mathbb{R}^p$.
- ② Use an algorithm for regression or pattern recognition in \mathbb{R}^p .



Example

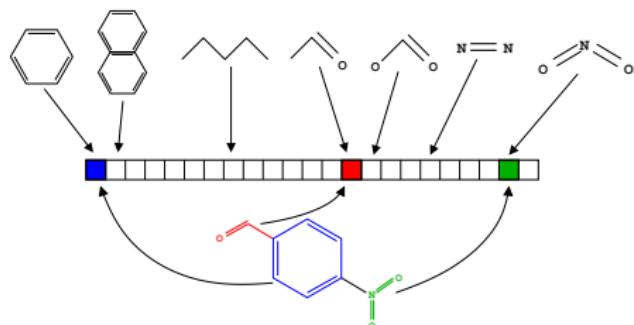
2D structural keys in chemoinformatics

- Index a molecule by a binary fingerprint defined by a limited set of **predefined** structures



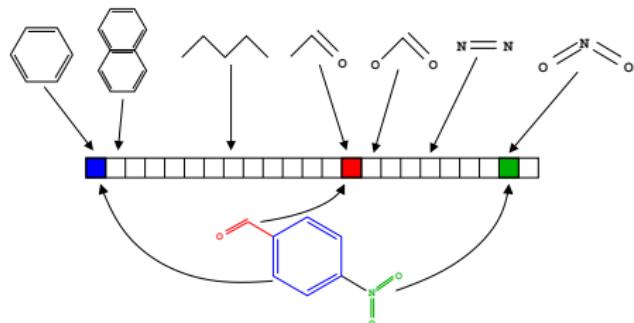
- Use a machine learning algorithm such as SVM, k NN, PLS, decision tree, etc.

Challenge: which descriptors (patterns)?



- **Expressiveness:** they should retain as much information as possible from the graph
- **Computation:** they should be fast to compute
- **Large dimension** of the vector representation: memory storage, speed, statistical issues

Indexing by substructures

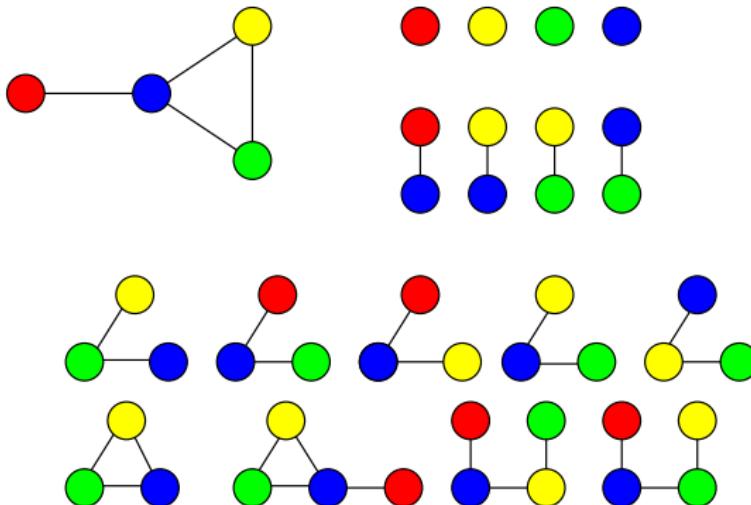


- Often we believe that **the presence or absence of particular substructures** may be important predictive patterns
- Hence it makes sense to represent a graph by **features** that indicate the presence (or the number of occurrences) of these substructures
- However, detecting the presence of particular substructures may be **computationally challenging...**

Subgraphs

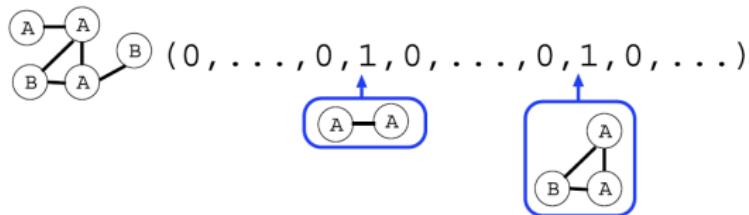
Definition

A **subgraph** of a graph (V, E) is a graph (V', E') with $V' \subset V$ and $E' \subset E$.

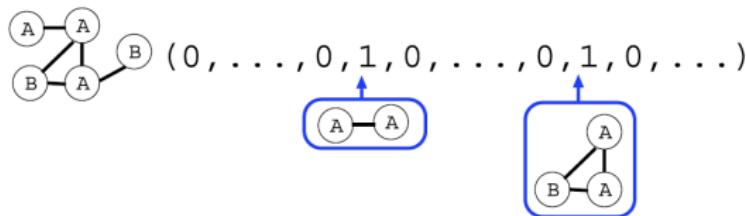


A graph and all its connected subgraphs.

Indexing by all subgraphs?



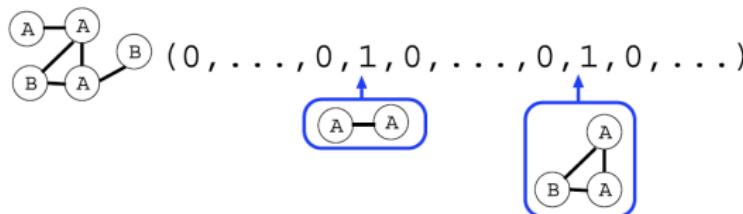
Indexing by all subgraphs?



Theorem

Computing all subgraph occurrences is NP-hard.

Indexing by all subgraphs?



Theorem

Computing all subgraph occurrences is NP-hard.

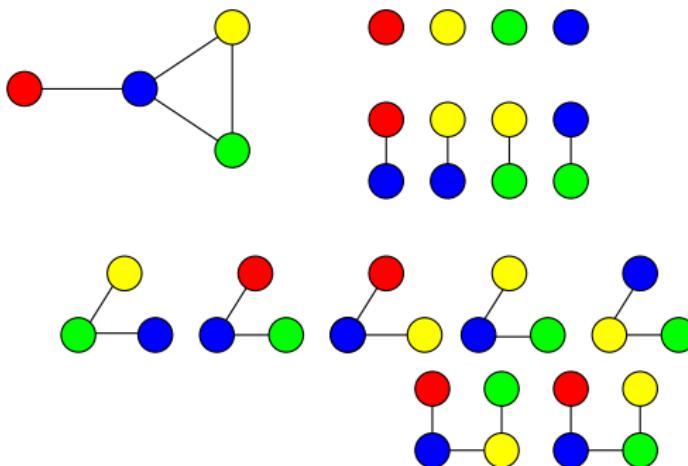
Proof

- The linear graph of size n is a subgraph of a graph X with n vertices iff X has a Hamiltonian path;
- The decision problem whether a graph has a Hamiltonian path is NP-complete.

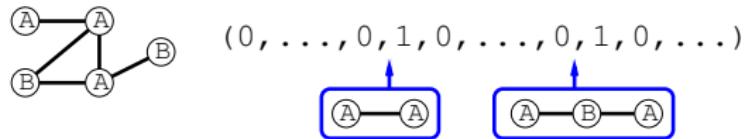
Paths

Definition

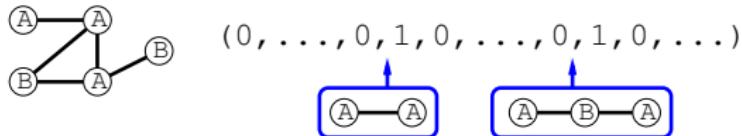
- A **path** of a graph (V, E) is a sequence of **distinct vertices** $v_1, \dots, v_n \in V$ ($i \neq j \implies v_i \neq v_j$) such that $(v_i, v_{i+1}) \in E$ for $i = 1, \dots, n - 1$.
- Equivalently the paths are the **linear subgraphs**.



Indexing by all paths?



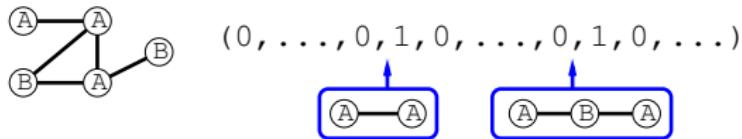
Indexing by all paths?



Theorem

Computing all path occurrences is NP-hard.

Indexing by all paths?



Theorem

Computing all path occurrences is NP-hard.

Proof

Same as for subgraphs.

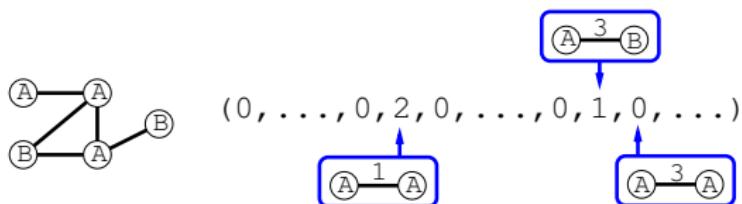
Indexing by what?

Substructure selection

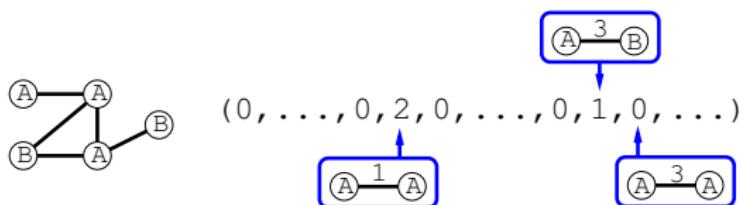
We can imagine more limited sets of substructures that lead to more computationnally efficient indexing (non-exhaustive list)

- substructures selected by domain knowledge (MDL fingerprint)
- all paths up to length k (Openeye fingerprint, Nicholls 2005)
- all shortest path lengths (Borgwardt and Kriegel, 2005)
- all subgraphs up to k vertices (graphlet kernel, Shervashidze et al., 2009)
- all frequent subgraphs in the database (Helma et al., 2004)

Example: Indexing by all shortest path lengths and their endpoint labels



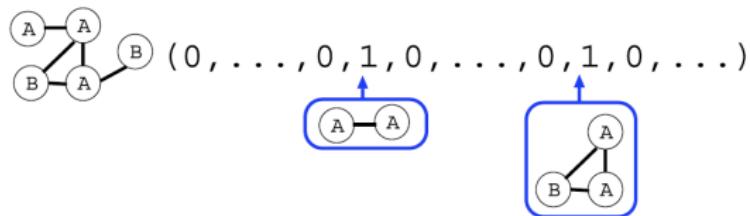
Example: Indexing by all shortest path lengths and their endpoint labels



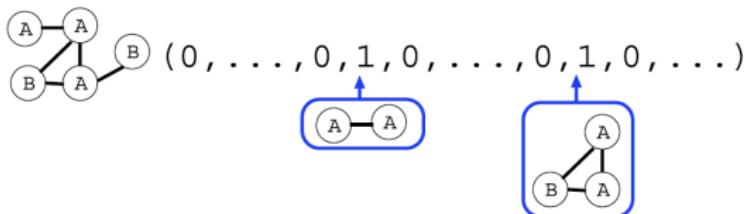
Properties (Borgwardt and Kriegel, 2005)

- There are $O(n^2)$ shortest paths.
- The vector of counts can be computed in $O(n^3)$ with the Floyd-Warshall algorithm.

Example: Indexing by all subgraphs up to k vertices



Example: Indexing by all subgraphs up to k vertices



Properties (Shervashidze et al., 2009)

- Naive enumeration scales as $O(n^k)$.
- Enumeration of connected graphlets in $O(nd^{k-1})$ for graphs with degree $\leq d$ and $k \leq 5$.
- Randomly sample subgraphs if enumeration is infeasible.

Summary

- Explicit computation of substructure occurrences can be **computationnally prohibitive** (subgraphs, paths);
- Several ideas to **reduce** the set of substructures considered;
- In practice, NP-hardness may not be so prohibitive (e.g., graphs with small degrees), the strategy followed should depend on the data considered.

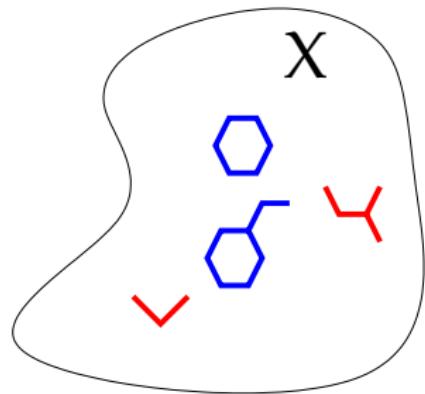
Outline

3 Kernels and Graphs

• Kernels for graphs

- Motivation
- Explicit enumeration of features
- **Implicit enumeration of features**
- Walk-based kernels
- Kernels on graphs

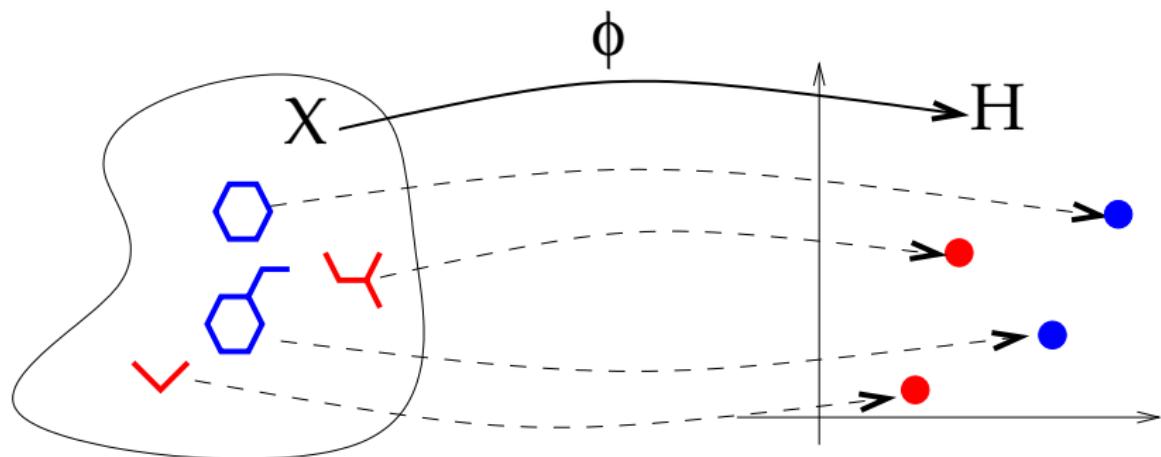
The idea



The idea

- ① Represent **implicitly** each graph \mathbf{x} in \mathcal{X} by a vector $\Phi(\mathbf{x}) \in \mathcal{H}$ through the kernel

$$K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^\top \Phi(\mathbf{x}').$$

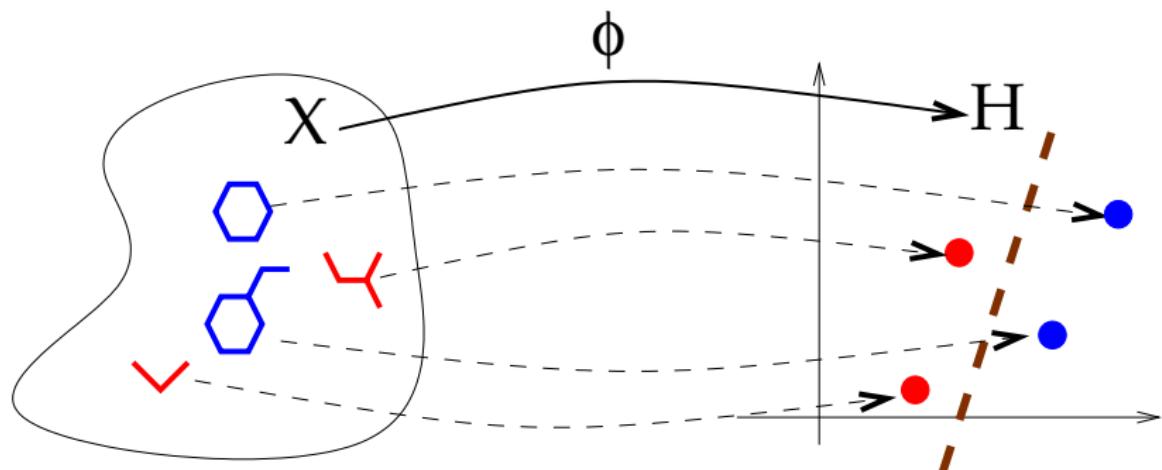


The idea

- ① Represent **implicitly** each graph \mathbf{x} in \mathcal{X} by a vector $\Phi(\mathbf{x}) \in \mathcal{H}$ through the kernel

$$K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^\top \Phi(\mathbf{x}').$$

- ② Use a kernel method for classification in \mathcal{H} .



Expressiveness vs Complexity

Definition: Complete graph kernels

A graph kernel is **complete** if it distinguishes non-isomorphic graphs, i.e.:

$$\forall G_1, G_2 \in \mathcal{X}, \quad d_K(G_1, G_2) = 0 \implies G_1 \simeq G_2.$$

Equivalently, $\Phi(G_1) \neq \Phi(G_2)$ if G_1 and G_2 are not isomorphic.

Expressiveness vs Complexity

Definition: Complete graph kernels

A graph kernel is **complete** if it distinguishes non-isomorphic graphs, i.e.:

$$\forall G_1, G_2 \in \mathcal{X}, \quad d_K(G_1, G_2) = 0 \implies G_1 \simeq G_2.$$

Equivalently, $\Phi(G_1) \neq \Phi(G_2)$ if G_1 and G_2 are not isomorphic.

Expressiveness vs Complexity trade-off

- If a graph kernel is not complete, then there is **no hope** to learn all possible functions over \mathcal{X} : the kernel is not **expressive** enough.
- On the other hand, kernel **computation** must be **tractable**, i.e., no more than polynomial (with small degree) for practical applications.
- Can we define **tractable** and **expressive** graph kernels?

Complexity of complete kernels

Proposition (Gärtner et al., 2003)

Computing **any complete graph kernel** is at least as hard as the graph isomorphism problem.

Complexity of complete kernels

Proposition (Gärtner et al., 2003)

Computing any complete graph kernel is at least as hard as the graph isomorphism problem.

Proof

- For any kernel K the complexity of computing d_K is the same as the complexity of computing K , because:

$$d_K(G_1, G_2)^2 = K(G_1, G_1) + K(G_2, G_2) - 2K(G_1, G_2).$$

- If K is a complete graph kernel, then computing d_K solves the graph isomorphism problem ($d_K(G_1, G_2) = 0$ iff $G_1 \simeq G_2$). \square

Subgraph kernel

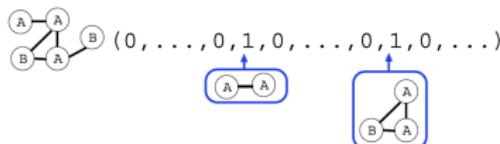
Definition

- Let $(\lambda_G)_{G \in \mathcal{X}}$ be a set or **nonnegative** real-valued weights
- For any graph $G \in \mathcal{X}$ and any connected graph $H \in \mathcal{X}$, let

$$\Phi_H(G) = |\{G' \text{ is a subgraph of } G : G' \simeq H\}|.$$

- The **subgraph kernel** between any two graphs G_1 and $G_2 \in \mathcal{X}$ is defined by:

$$K_{\text{subgraph}}(G_1, G_2) = \sum_{\substack{H \in \mathcal{X} \\ H \text{ connected}}} \lambda_H \Phi_H(G_1) \Phi_H(G_2).$$



Subgraph kernel complexity

Proposition (Gärtner et al., 2003)

Computing the subgraph kernel is **NP-hard**.

Subgraph kernel complexity

Proposition (Gärtner et al., 2003)

Computing the subgraph kernel is **NP-hard**.

Proof (1/2)

- Let P_n be the path graph with n vertices.
- Subgraphs of P_n are path graphs:

$$\Phi(P_n) = ne_{P_1} + (n - 1)e_{P_2} + \dots + e_{P_n}.$$

- The vectors $\Phi(P_1), \dots, \Phi(P_n)$ are linearly independent, therefore:

$$e_{P_n} = \sum_{i=1}^n \alpha_i \Phi(P_i),$$

where the coefficients α_i can be found in polynomial time (solving an $n \times n$ triangular system).

Subgraph kernel complexity

Proposition (Gärtner et al., 2003)

Computing the subgraph kernel is **NP-hard**.

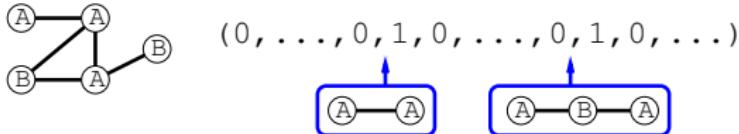
Proof (2/2)

- If G is a graph with n vertices, then it has a path that visits each node exactly once (Hamiltonian path) if and only if $\Phi(G)^\top e_{P_n} > 0$, i.e.,

$$\Phi(G)^\top \left(\sum_{i=1}^n \alpha_i \Phi(P_i) \right) = \sum_{i=1}^n \alpha_i K_{\text{subgraph}}(G, P_i) > 0.$$

- The decision problem whether a graph has a Hamiltonian path is NP-complete. \square

Path kernel



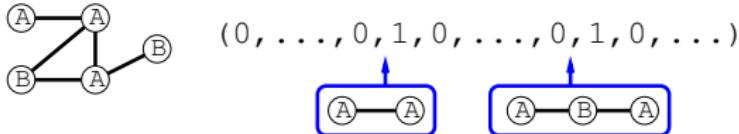
Definition

The **path kernel** is the subgraph kernel restricted to paths, i.e.,

$$K_{\text{path}}(G_1, G_2) = \sum_{H \in \mathcal{P}} \lambda_H \Phi_H(G_1) \Phi_H(G_2),$$

where $\mathcal{P} \subset \mathcal{X}$ is the set of path graphs.

Path kernel



Definition

The **path kernel** is the subgraph kernel restricted to paths, i.e.,

$$K_{\text{path}}(G_1, G_2) = \sum_{H \in \mathcal{P}} \lambda_H \Phi_H(G_1) \Phi_H(G_2),$$

where $\mathcal{P} \subset \mathcal{X}$ is the set of path graphs.

Proposition (Gärtner et al., 2003)

Computing the path kernel is **NP-hard**.

Summary

Expressiveness vs Complexity trade-off

- It is **intractable** to compute **complete graph kernels**.
- It is **intractable** to compute the **subgraph kernels**.
- Restricting subgraphs to be linear does not help: it is also **intractable** to compute the **path kernel**.
- One approach to define polynomial time computable graph kernels is to have the feature space be made up of graphs **homomorphic** to subgraphs, e.g., to consider **walks** instead of paths.

Outline

3 Kernels and Graphs

• Kernels for graphs

- Motivation
- Explicit enumeration of features
- Implicit enumeration of features

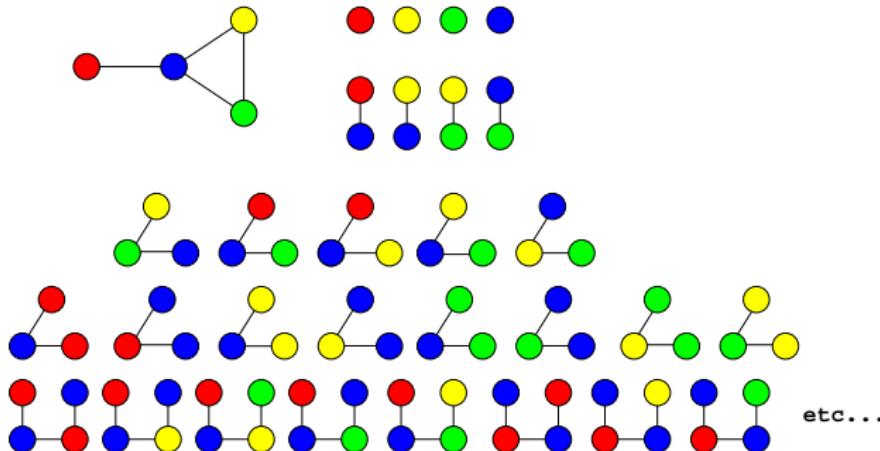
• Walk-based kernels

• Kernels on graphs

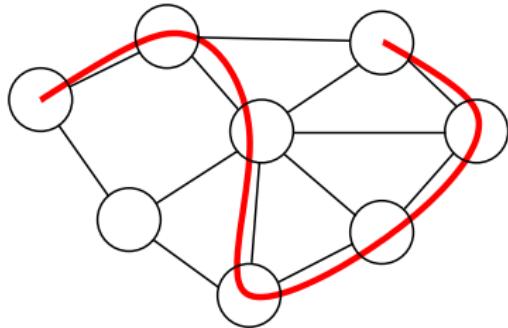
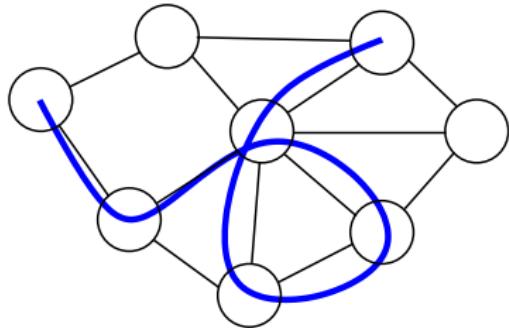
Walks

Definition

- A **walk** of a graph (V, E) is sequence of $v_1, \dots, v_n \in V$ such that $(v_i, v_{i+1}) \in E$ for $i = 1, \dots, n - 1$.
- We note $\mathcal{W}_n(G)$ the set of walks with n vertices of the graph G , and $\mathcal{W}(G)$ the set of all walks.



Walks \neq paths



Walk kernel

Definition

- Let \mathcal{S}_n denote the set of all possible **label sequences** of walks of length n (including vertex and edge labels), and $\mathcal{S} = \cup_{n \geq 1} \mathcal{S}_n$.
- For any graph \mathcal{X} let a **weight** $\lambda_G(w)$ be associated to each walk $w \in \mathcal{W}(G)$.
- Let the feature vector $\Phi(G) = (\Phi_s(G))_{s \in \mathcal{S}}$ be defined by:

$$\Phi_s(G) = \sum_{w \in \mathcal{W}(G)} \lambda_G(w) \mathbf{1} (s \text{ is the label sequence of } w).$$

Walk kernel

Definition

- Let \mathcal{S}_n denote the set of all possible **label sequences** of walks of length n (including vertex and edge labels), and $\mathcal{S} = \cup_{n \geq 1} \mathcal{S}_n$.
- For any graph \mathcal{X} let a **weight** $\lambda_G(w)$ be associated to each walk $w \in \mathcal{W}(G)$.
- Let the feature vector $\Phi(G) = (\Phi_s(G))_{s \in \mathcal{S}}$ be defined by:

$$\Phi_s(G) = \sum_{w \in \mathcal{W}(G)} \lambda_G(w) \mathbf{1} (s \text{ is the label sequence of } w).$$

- A walk kernel is a graph kernel defined by:

$$K_{\text{walk}}(G_1, G_2) = \sum_{s \in \mathcal{S}} \Phi_s(G_1) \Phi_s(G_2).$$

Walk kernel examples

Examples

- The n th-order walk kernel is the walk kernel with $\lambda_G(w) = 1$ if the length of w is n , 0 otherwise. It compares two graphs through their common walks of length n .

Walk kernel examples

Examples

- The *n*th-order walk kernel is the walk kernel with $\lambda_G(w) = 1$ if the length of w is n , 0 otherwise. It compares two graphs through their common walks of length n .
- The random walk kernel is obtained with $\lambda_G(w) = P_G(w)$, where P_G is a Markov random walk on G . In that case we have:

$$K(G_1, G_2) = P(\text{label}(W_1) = \text{label}(W_2)),$$

where W_1 and W_2 are two independent random walks on G_1 and G_2 , respectively (Kashima et al., 2003).

Walk kernel examples

Examples

- The *n*th-order walk kernel is the walk kernel with $\lambda_G(w) = 1$ if the length of w is n , 0 otherwise. It compares two graphs through their common walks of length n .
- The random walk kernel is obtained with $\lambda_G(w) = P_G(w)$, where P_G is a Markov random walk on G . In that case we have:

$$K(G_1, G_2) = P(\text{label}(W_1) = \text{label}(W_2)),$$

where W_1 and W_2 are two independent random walks on G_1 and G_2 , respectively (Kashima et al., 2003).

- The geometric walk kernel is obtained (when it converges) with $\lambda_G(w) = \beta^{\text{length}(w)}$, for $\beta > 0$. In that case the feature space is of infinite dimension (Gärtner et al., 2003).

Computation of walk kernels

Proposition

These three kernels (n th-order, random and geometric walk kernels) can be computed efficiently in **polynomial time**.

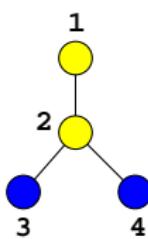
Product graph

Definition

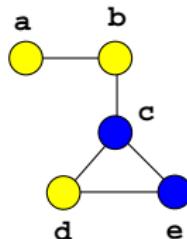
Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs with labeled vertices.

The **product graph** $G = G_1 \times G_2$ is the graph $G = (V, E)$ with:

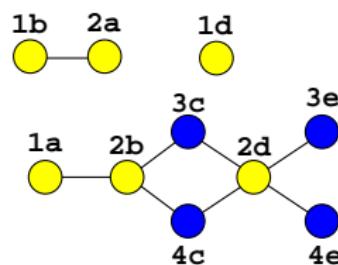
- ① $V = \{(v_1, v_2) \in V_1 \times V_2 : v_1 \text{ and } v_2 \text{ have the same label}\}$,
- ② $E = \{((v_1, v_2), (v'_1, v'_2)) \in V \times V : (v_1, v'_1) \in E_1 \text{ and } (v_2, v'_2) \in E_2\}$.



G_1



G_2



$G_1 \times G_2$

Walk kernel and product graph

Lemma

There is a **bijection** between:

- ① The **pairs of walks** $w_1 \in \mathcal{W}_n(G_1)$ and $w_2 \in \mathcal{W}_n(G_2)$ with the **same label sequences**,
- ② The **walks on the product graph** $w \in \mathcal{W}_n(G_1 \times G_2)$.

Walk kernel and product graph

Lemma

There is a **bijection** between:

- ① The **pairs of walks** $w_1 \in \mathcal{W}_n(G_1)$ and $w_2 \in \mathcal{W}_n(G_2)$ with the **same label sequences**,
- ② The **walks on the product graph** $w \in \mathcal{W}_n(G_1 \times G_2)$.

Corollary

$$\begin{aligned} K_{\text{walk}}(G_1, G_2) &= \sum_{s \in \mathcal{S}} \Phi_s(G_1) \Phi_s(G_2) \\ &= \sum_{(w_1, w_2) \in \mathcal{W}(G_1) \times \mathcal{W}(G_2)} \lambda_{G_1}(w_1) \lambda_{G_2}(w_2) \mathbf{1}(l(w_1) = l(w_2)) \\ &= \sum_{w \in \mathcal{W}(G_1 \times G_2)} \lambda_{G_1 \times G_2}(w). \end{aligned}$$

Computation of the n th-order walk kernel

- For the n th-order walk kernel we have $\lambda_{G_1 \times G_2}(w) = 1$ if the length of w is n , 0 otherwise.
- Therefore:

$$K_{n\text{th-order}}(G_1, G_2) = \sum_{w \in \mathcal{W}_n(G_1 \times G_2)} 1.$$

- Let A be the adjacency matrix of $G_1 \times G_2$. Then we get:

$$K_{n\text{th-order}}(G_1, G_2) = \sum_{i,j} [A^n]_{i,j} = \mathbf{1}^\top A^n \mathbf{1}.$$

- Computation in $O(n|V_1||V_2|d_1 d_2)$, where d_i is the maximum degree of G_i .

Computation of random and geometric walk kernels

- In both cases $\lambda_G(w)$ for a walk $w = v_1 \dots v_n$ can be decomposed as:

$$\lambda_G(v_1 \dots v_n) = \lambda^i(v_1) \prod_{i=2}^n \lambda^t(v_{i-1}, v_i).$$

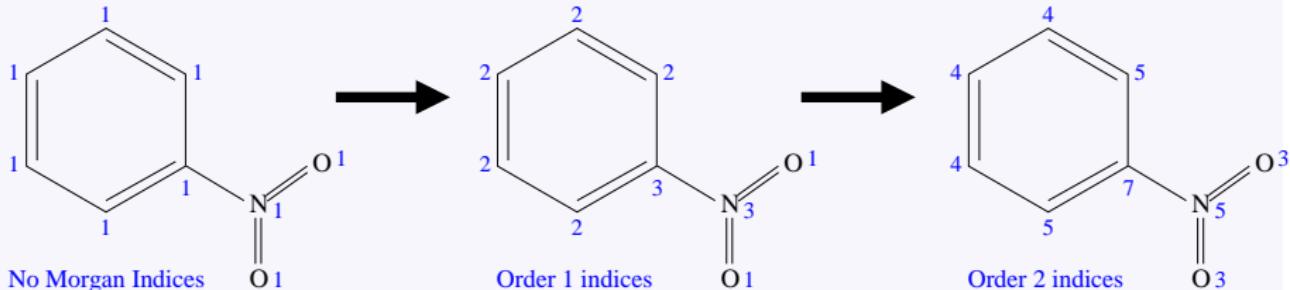
- Let Λ_i be the vector of $\lambda^i(v)$ and Λ_t be the matrix of $\lambda^t(v, v')$:

$$\begin{aligned} K_{\text{walk}}(G_1, G_2) &= \sum_{n=1}^{\infty} \sum_{w \in \mathcal{W}_n(G_1 \times G_2)} \lambda^i(v_1) \prod_{i=2}^n \lambda^t(v_{i-1}, v_i) \\ &= \sum_{n=0}^{\infty} \Lambda_i \Lambda_t^n \mathbf{1} \\ &= \color{red} \Lambda_i (I - \Lambda_t)^{-1} \mathbf{1} \end{aligned}$$

- Computation in $O(|V_1|^3 |V_2|^3)$.

Extensions 1: Label enrichment

Atom relabeling with the Morgan index (Mahé et al., 2004)

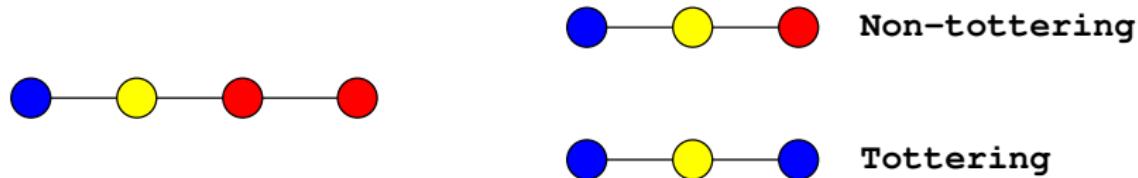


- Compromise between **fingerprints** and **structural keys**.
- Other **relabeling** schemes are possible.
- Faster computation with **more labels** (less matches implies a smaller product graph).

Extension 2: Non-tottering walk kernel

Tottering walks

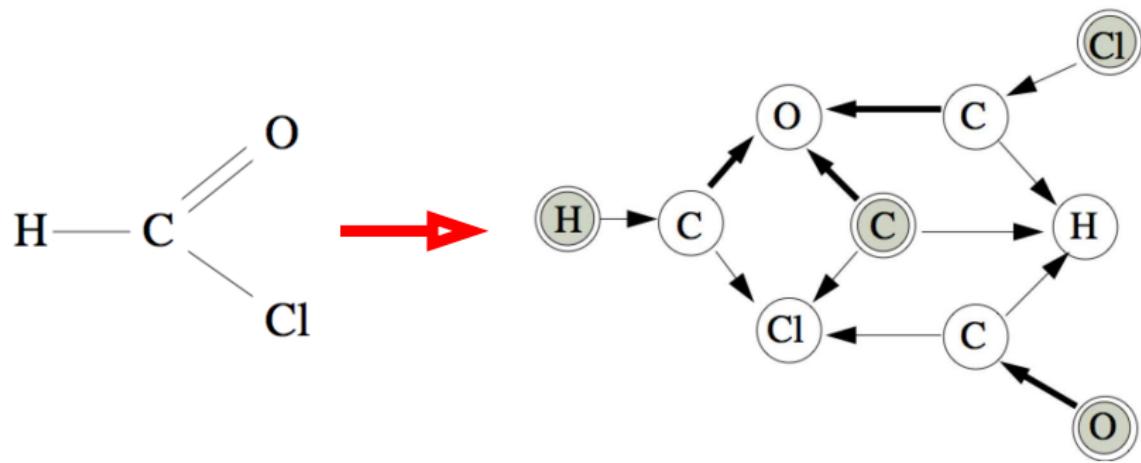
A **tottering walk** is a walk $w = v_1 \dots v_n$ with $v_i = v_{i+2}$ for some i .



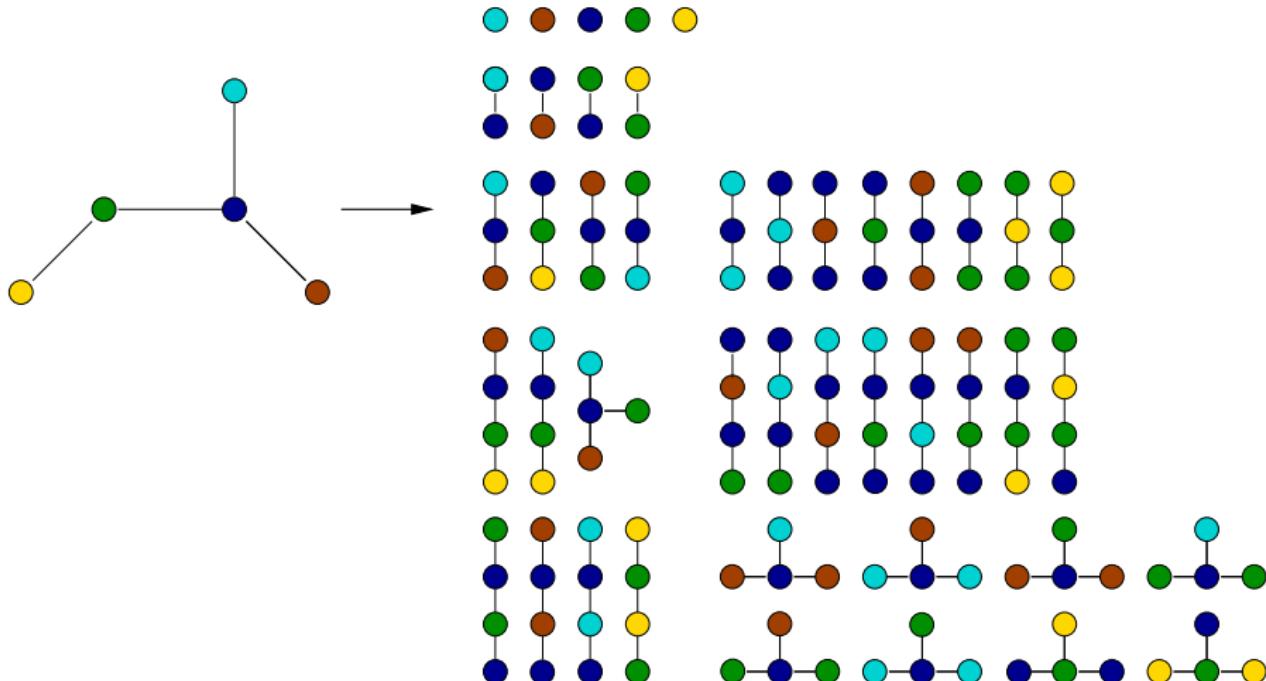
- Tottering walks seem **irrelevant** for many applications.
- Focusing on non-tottering walks is a way to get closer to the **path kernel** (e.g., equivalent on trees).

Computation of the non-tottering walk kernel (Mahé et al., 2005)

- Second-order Markov random walk to prevent tottering walks
- Written as a first-order Markov random walk on an augmented graph
- Normal walk kernel on the augmented graph (which is always a directed graph).

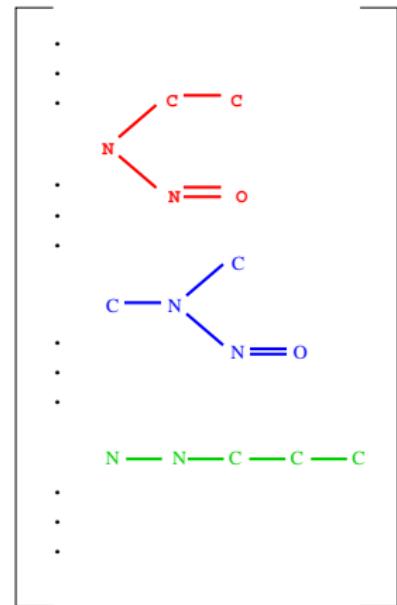
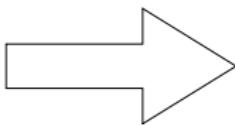
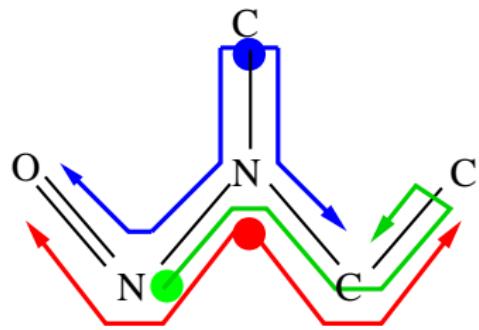


Extension 3: Subtree kernels



Remark: Here and in subsequent slides by *subtree* we mean a tree-like pattern with potentially repeated nodes and edges.

Example: Tree-like fragments of molecules



Computation of the subtree kernel (Ramon and Gärtner, 2003; Mahé and Vert, 2009)

- Like the walk kernel, amounts to computing the (weighted) number of subtrees in the **product graph**.
- Recursion: if $\mathcal{T}(v, n)$ denotes the weighted number of subtrees of depth n rooted at the vertex v , then:

$$\mathcal{T}(v, n+1) = \sum_{R \subset \mathcal{N}(v)} \prod_{v' \in R} \lambda_t(v, v') \mathcal{T}(v', n),$$

where $\mathcal{N}(v)$ is the set of neighbors of v .

- Can be combined with the non-tottering graph transformation as preprocessing to obtain the **non-tottering subtree kernel**.

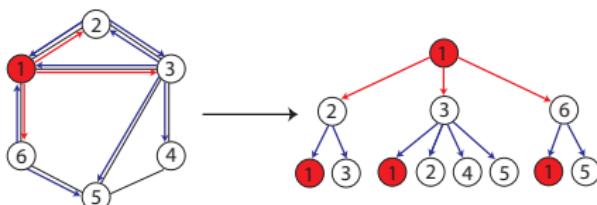
Back to label enrichment

Link between the Morgan index and subtrees

Recall the Morgan index:



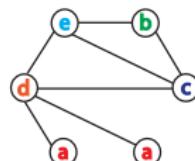
The Morgan index of order k at a node v in fact corresponds to the number of leaves in the k -th order full subtree pattern rooted at v .



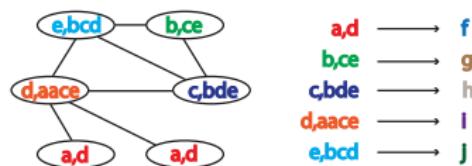
Label enrichment via the Weisfeiler-Lehman algorithm

A slightly more involved label enrichment strategy (Weisfeiler and Lehman, 1968) is exploited in the definition and computation of the Weisfeiler-Lehman subtree kernel (Shervashidze and Borgwardt, 2009).

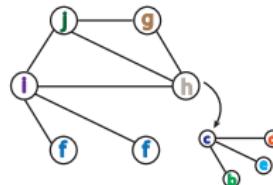
- ① Multiset-label determination and sorting



- ② Label compression



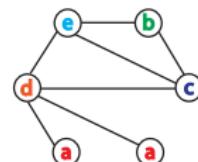
- ③ Relabeling



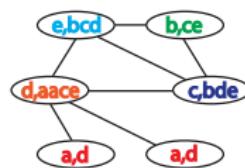
Label enrichment via the Weisfeiler-Lehman algorithm

A slightly more involved label enrichment strategy (Weisfeiler and Lehman, 1968) is exploited in the definition and computation of the Weisfeiler-Lehman subtree kernel (Shervashidze and Borgwardt, 2009).

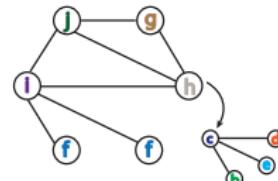
- ## ① Multiset-label determination and sorting



- ## ② Label compression

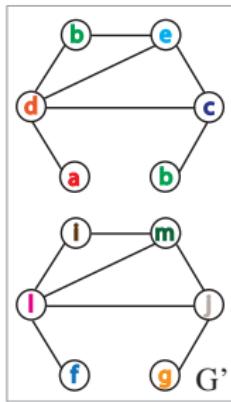
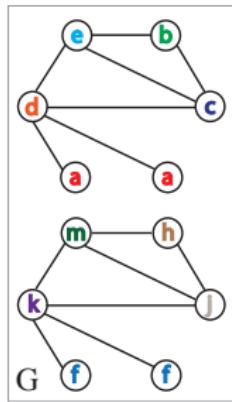


- ### ③ Relabeling



Compressed labels represent full subtree patterns.

Weisfeiler-Lehman (WL) subtree kernel



$$\phi_{WLsubtree}^{(1)}(G) = (2, 1, 1, 1, 1, 2, 0, 1, 0, 1, 1, 0, 1)$$

a b c d e f g h i j k l m

$$\phi_{WLsubtree}^{(1)}(G') = (1, 2, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1)$$

a b c d e f g h i j k l m

Counts of
original
node labels

Counts of
compressed
node labels

Properties

- The WL features up to the k -th order are computed in $O(|E|k)$.
- Similarly to the Morgan index, the WL relabeling can be exploited in combination with any graph kernel (that takes into account categorical node labels) to make it more expressive (Shervashidze et al., 2011).

Summary: graph kernels

What we saw

- Kernels do **not allow** to overcome the NP-hardness of subgraph patterns.
- They allow to work with approximate subgraphs (walks, subtrees) in infinite dimension, thanks to the **kernel trick**.
- However: using kernels makes it difficult to **come back to patterns** after the learning stage.

Outline

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
- 3 Kernels and Graphs
 - Kernels for graphs
 - Kernels on graphs
- 4 Characterizing probabilities with kernels
- 5 Open Problems and Research Topics

Outline

3 Kernels and Graphs

- Kernels for graphs
- **Kernels on graphs**
 - Motivation
 - Graph distance and p.d. kernels
 - Construction by regularization
 - The diffusion kernel
 - Harmonic analysis on graphs
 - Applications

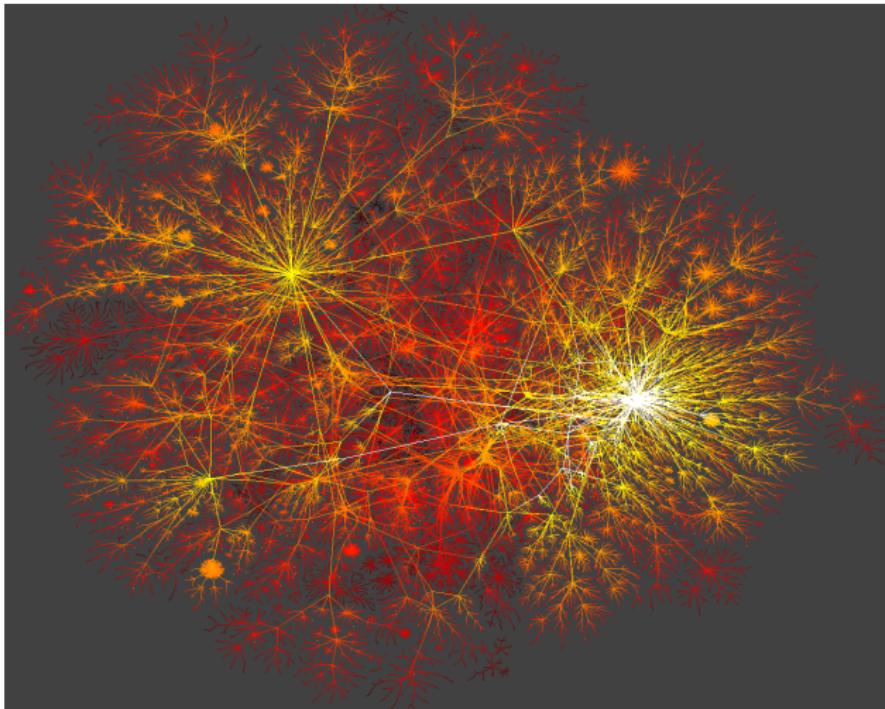
Graphs

Motivation

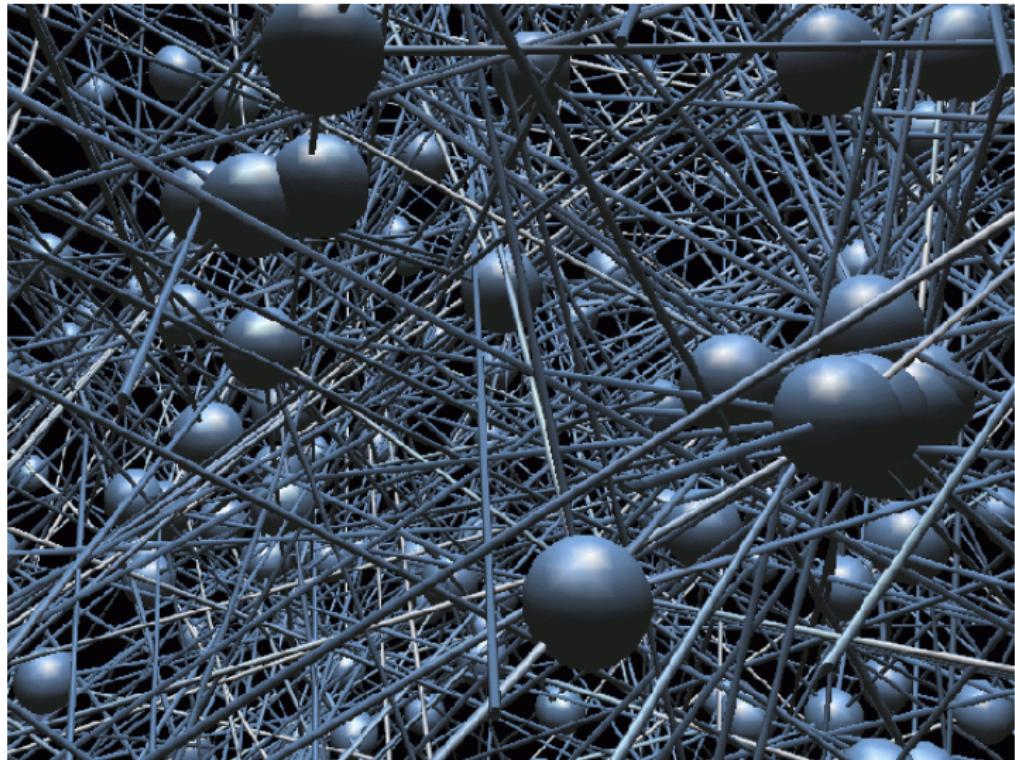
Data often come in the form of **nodes in a graph** for different reasons:

- by **definition** (interaction network, internet...)
- by **discretization**/sampling of a continuous domain
- by **convenience** (e.g., if only a similarity function is available)

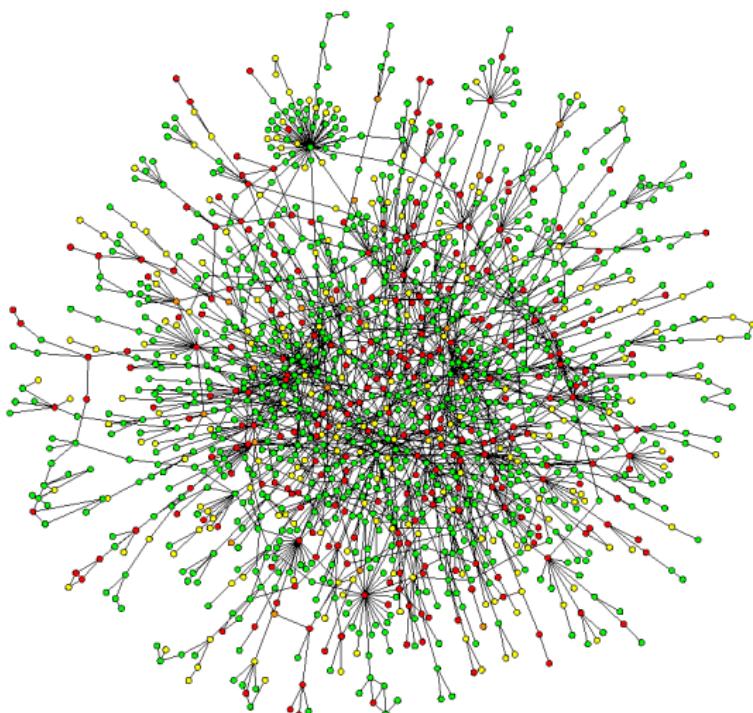
Example: web



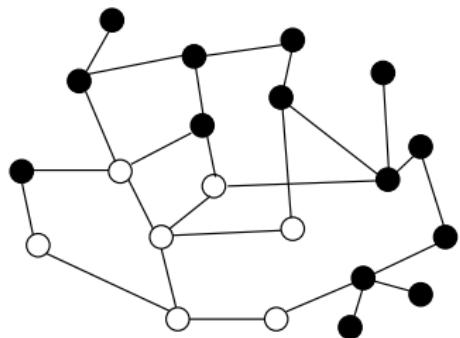
Example: social network



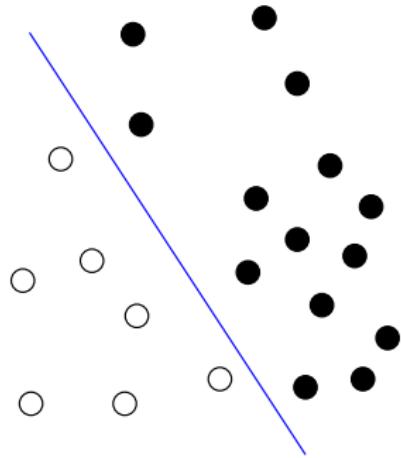
Example: protein-protein interaction



Kernel on a graph



ϕ

A red curved arrow points from the graph on the left to the feature space on the right, indicating a mapping function ϕ .

- We need a kernel $K(x, x')$ between nodes of the graph.
- Example: predict protein functions from high-throughput protein-protein interaction data.

General remarks

Strategies to design a kernel on a graph

- \mathcal{X} being finite, any symmetric semi-definite matrix K defines a valid p.d. kernel on \mathcal{X} .

General remarks

Strategies to design a kernel on a graph

- \mathcal{X} being finite, any symmetric semi-definite matrix K defines a valid p.d. kernel on \mathcal{X} .
- How to “translate” the graph topology into the kernel?
 - Direct geometric approach: $K_{i,j}$ should be “large” when \mathbf{x}_i and \mathbf{x}_j are “close” to each other on the graph?
 - Functional approach: $\|f\|_K$ should be “small” when f is “smooth” on the graph?
 - Link discrete/continuous: is there an equivalent to the continuous Gaussian kernel on the graph (e.g., limit by fine discretization)?

Outline

3 Kernels and Graphs

- Kernels for graphs
- **Kernels on graphs**
 - Motivation
 - **Graph distance and p.d. kernels**
 - Construction by regularization
 - The diffusion kernel
 - Harmonic analysis on graphs
 - Applications

Conditionally p.d. kernels

Hilbert distance

- Any p.d. kernel is an inner product in a Hilbert space

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

- It defines a Hilbert distance:

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

- $-d_K^2$ is **conditionally positive definite (c.p.d.)**, i.e.:

$$\forall t > 0, \quad \exp(-td_K(\mathbf{x}, \mathbf{x}')^2) \text{ is p.d.}$$

Example

A direct approach

- For $\mathcal{X} = \mathbb{R}^n$, the inner product is p.d.:

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'.$$

- The corresponding Hilbert distance is the Euclidean distance:

$$d_K(\mathbf{x}, \mathbf{x}')^2 = \mathbf{x}^\top \mathbf{x} + \mathbf{x}'^\top \mathbf{x}' - 2\mathbf{x}^\top \mathbf{x}' = \|\mathbf{x} - \mathbf{x}'\|^2.$$

- $-d_K^2$ is **conditionally positive definite (c.p.d.)**, i.e.:

$$\forall t > 0, \quad \exp(-t\|\mathbf{x} - \mathbf{x}'\|^2) \text{ is p.d.}$$

Graph distance

Graph embedding in a Hilbert space

- Given a graph $G = (V, E)$, the **graph distance** $d_G(x, x')$ between any two vertices is the **length of the shortest path** between x and x' .
- We say that the graph $G = (V, E)$ can be **embedded** (exactly) in a Hilbert space if $-d_G$ is **c.p.d.**, which implies in particular that $\exp(-td_G(x, x'))$ is p.d. for all $t > 0$.

Graph distance

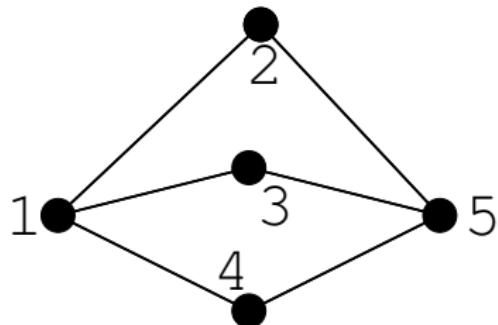
Graph embedding in a Hilbert space

- Given a graph $G = (V, E)$, the **graph distance** $d_G(x, x')$ between any two vertices is the **length of the shortest path** between x and x' .
- We say that the graph $G = (V, E)$ can be **embedded** (exactly) in a Hilbert space if $-d_G$ is **c.p.d.**, which implies in particular that $\exp(-td_G(x, x'))$ is p.d. for all $t > 0$.

Lemma

- In general graphs cannot** be embedded exactly in Hilbert spaces.
- In some cases exact embeddings exist**, e.g.:
 - trees** can be embedded exactly,
 - closed chains** can be embedded exactly.

Example: non-c.p.d. graph distance



$$d_G = \begin{pmatrix} 0 & 1 & 1 & 1 & 2 \\ 1 & 0 & 2 & 2 & 1 \\ 1 & 2 & 0 & 2 & 1 \\ 1 & 2 & 2 & 0 & 1 \\ 2 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\lambda_{\min} \left(\left[e^{(-0.2d_G(i,j))} \right] \right) = -0.028 < 0.$$

Graph distances on trees are c.p.d.

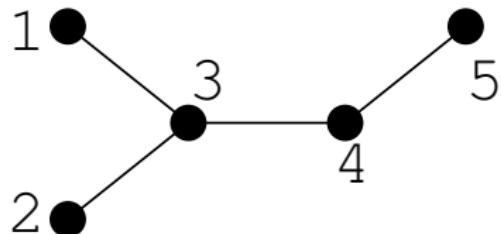
Proof

- Let $G = (V, E)$ be a tree;
- Fix a root $x_0 \in V$;
- Represent any vertex $x \in V$ by a vector $\Phi(x) \in \mathbb{R}^{|E|}$, where $\Phi(x)_i = 1$ if the i -th edge is part of the (unique) path between x and x_0 , 0 otherwise.
- Then

$$d_G(x, x') = \| \Phi(x) - \Phi(x') \|^2,$$

and therefore $-d_G$ is c.p.d., in particular $\exp(-td_G(x, x'))$ is p.d. for all $t > 0$.

Example

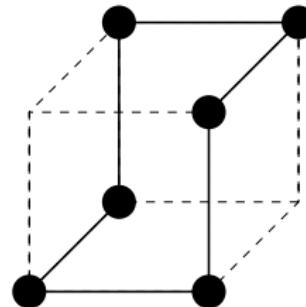
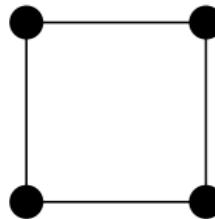


$$\left[e^{-d_G(i,j)} \right] = \begin{pmatrix} 1 & 0.14 & 0.37 & 0.14 & 0.05 \\ 0.14 & 1 & 0.37 & 0.14 & 0.05 \\ 0.37 & 0.37 & 1 & 0.37 & 0.14 \\ 0.14 & 0.14 & 0.37 & 1 & 0.37 \\ 0.05 & 0.05 & 0.14 & 0.37 & 1 \end{pmatrix}$$

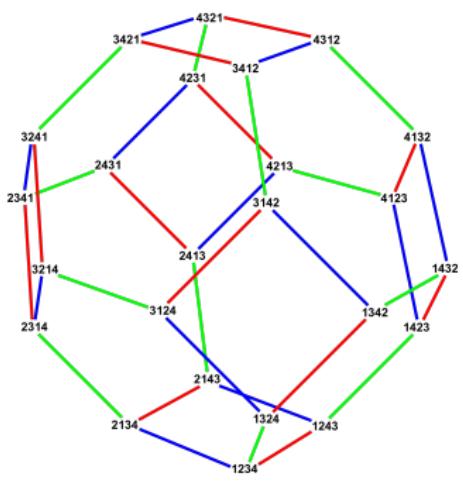
Graph distances on closed chains are c.p.d.

Proof: case $|V| = 2p$

- Let $G = (V, E)$ be a directed cycle with an even number of vertices $|V| = 2p$.
- Fix a root $x_0 \in V$, number the $2p$ edges from x_0 to x_0 ;
- Label the $2p$ edges with $e_1, \dots, e_p, -e_1, \dots, -e_p$ (vectors in \mathbb{R}^p);
- For a vertex v , take $\Phi(v)$ to be the sum of the labels of the edges in the shortest directed path between x_0 and v .



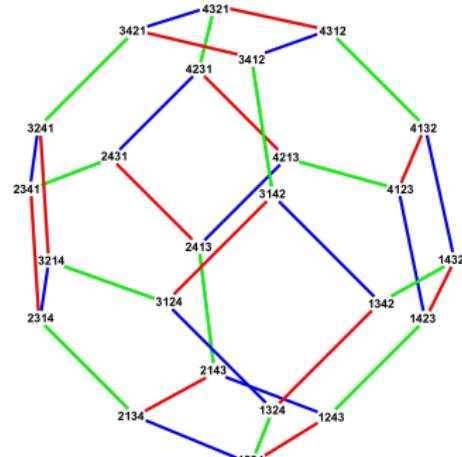
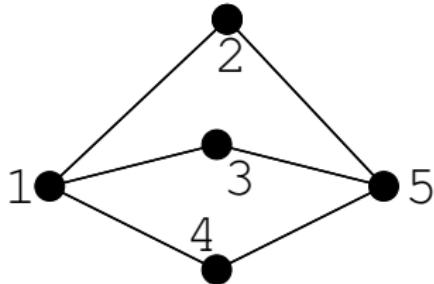
Another interesting graph



Cayley graph of \mathbb{S}_4

- Let \mathbb{S}_n the set of permutations of n items (symmetric group)
- Cayley graph G : connect two permutations when they differ by one adjacent transposition
- d_G can be computed in $O(n \log n)$ how?
- d_G is c.p.d. why?
- See Jiao and Vert (2017)

Summary on graph distance



- Some graph distances are c.p.d, some are not
- There is a large literature in mathematics on how to "approximately" embed a graph; maybe this could be useful for machine learning?
- Graph distance is very sensitive to "noise" in edges
- We need other approaches to define a p.d. kernel that would work for all graphs, and be less sensitive to noise in the edges.

Outline

3 Kernels and Graphs

- Kernels for graphs
- **Kernels on graphs**
 - Motivation
 - Graph distance and p.d. kernels
 - **Construction by regularization**
 - The diffusion kernel
 - Harmonic analysis on graphs
 - Applications

Functional approach

Motivation

- How to design a p.d. kernel on **general graphs**?
- Designing a kernel is equivalent to defining an **RKHS**.
- There are intuitive notions of **smoothness** on a graph.

Idea

- Define a priori a **smoothness functional** on the functions $f : \mathcal{X} \rightarrow \mathbb{R}$;
- Show that **it defines an RKHS** and identify the corresponding kernel.

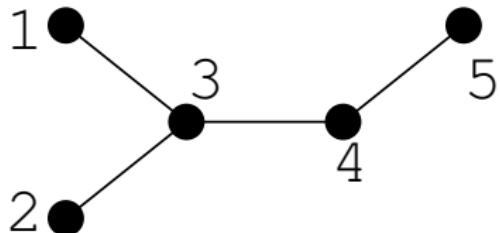
Notations

- $\mathcal{X} = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ is finite.
- For $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, we note $\mathbf{x} \sim \mathbf{x}'$ to indicate the existence of an edge between \mathbf{x} and \mathbf{x}' .
- We assume that there is no self-loop $\mathbf{x} \sim \mathbf{x}$, and that there is a single connected component.
- The adjacency matrix is $A \in \mathbb{R}^{m \times m}$:

$$A_{i,j} = \begin{cases} 1 & \text{if } i \sim j, \\ 0 & \text{otherwise.} \end{cases}$$

- D is the diagonal matrix where $D_{i,i}$ is the number of neighbors of \mathbf{x}_i ($D_{i,i} = \sum_{j=1}^m A_{i,j}$).

Example

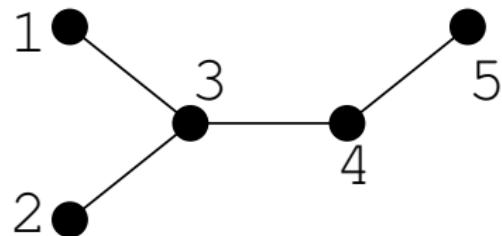


$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Graph Laplacian

Definition

The Laplacian of the graph is the matrix $L = D - A$.



$$L = D - A = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

Properties of the Laplacian

Lemma

Let $L = D - A$ be the Laplacian of a **connected** graph:

- For any $f : \mathcal{X} \rightarrow \mathbb{R}$,

$$\Omega(f) := \sum_{i \sim j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 = f^\top L f$$

- L is a **symmetric positive semi-definite** matrix
- 0 is an **eigenvalue** with multiplicity 1 associated to the constant eigenvector $\mathbf{1} = (1, \dots, 1)$
- The **image** of L is

$$Im(L) = \left\{ f \in \mathbb{R}^m : \sum_{i=1}^m f_i = 0 \right\}$$

Proof: link between $\Omega(f)$ and L

$$\begin{aligned}\Omega(f) &= \sum_{i \sim j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 \\&= \sum_{i \sim j} \left(f(\mathbf{x}_i)^2 + f(\mathbf{x}_j)^2 - 2f(\mathbf{x}_i)f(\mathbf{x}_j) \right) \\&= \sum_{i=1}^m D_{i,i} f(\mathbf{x}_i)^2 - 2 \sum_{i \sim j} f(\mathbf{x}_i)f(\mathbf{x}_j) \\&= \mathbf{f}^\top D \mathbf{f} - \mathbf{f}^\top A \mathbf{f} \\&= \mathbf{f}^\top L \mathbf{f}\end{aligned}$$

Proof: eigenstructure of L

- L is symmetric because A and D are symmetric.
- For any $f \in \mathbb{R}^m$, $f^\top L f = \Omega(f) \geq 0$, therefore the (real-valued) eigenvalues of L are ≥ 0 : L is therefore positive semi-definite.
- f is an eigenvector associated to eigenvalue 0
iff $f^\top L f = 0$
iff $\sum_{i \sim j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 = 0$,
iff $f(\mathbf{x}_i) = f(\mathbf{x}_j)$ when $i \sim j$,
iff f is constant (because the graph is connected).
- L being symmetric, $Im(L)$ is the orthogonal supplement of $Ker(L)$, that is, the set of functions orthogonal to $\mathbf{1}$. \square

Our first graph kernel

Theorem

The set $\mathcal{H} = \{f \in \mathbb{R}^m : \sum_{i=1}^m f_i = 0\}$ endowed with the norm

$$\Omega(f) = \sum_{i \sim j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2$$

is a RKHS whose reproducing kernel is L^* , the pseudo-inverse of the graph Laplacian.

In case of...

Pseudo-inverse of L

Remember the pseudo-inverse L^* of L is the linear application that is equal to:

- 0 on $\text{Ker}(L)$
- L^{-1} on $\text{Im}(L)$, that is, if we write:

$$L = \sum_{i=1}^m \lambda_i u_i u_i^\top$$

the eigendecomposition of L :

$$L^* = \sum_{\lambda_i \neq 0} (\lambda_i)^{-1} u_i u_i^\top.$$

- In particular it holds that $L^* L = L L^* = \Pi_{\mathcal{H}}$, the projection onto $\text{Im}(L) = \mathcal{H}$.

Proof (1/2)

- Restricted to \mathcal{H} , the symmetric bilinear form:

$$\langle f, g \rangle = f^\top L g$$

is positive definite (because L is positive semi-definite, and $\mathcal{H} = \text{Im}(L)$). It is therefore a scalar product, making of \mathcal{H} a **Hilbert space** (in fact Euclidean).

- The norm in this Hilbert space \mathcal{H} is:

$$\|f\|^2 = \langle f, f \rangle = f^\top L f = \Omega(f) .$$

Proof (2/2)

To check that \mathcal{H} is a RKHS with reproducing kernel $K = L^*$, it suffices to show that:

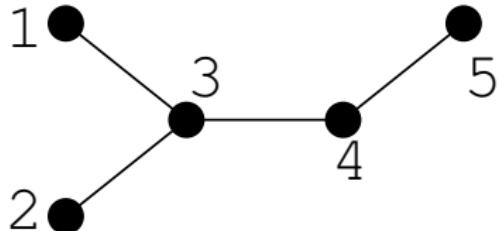
$$\begin{cases} \forall \mathbf{x} \in \mathcal{X}, & K_{\mathbf{x}} \in \mathcal{H}, \\ \forall (\mathbf{x}, f) \in \mathcal{X} \times \mathcal{H}, & \langle f, K_{\mathbf{x}} \rangle = f(\mathbf{x}) . \end{cases}$$

- $\text{Ker}(K) = \text{Ker}(L^*) = \text{Ker}(L)$, implying $K\mathbf{1} = 0$. Therefore, each row/column of K is in \mathcal{H} .
- For any $f \in \mathcal{H}$, if we note $g_i = \langle K(i, \cdot), f \rangle$ we get:

$$g = KLf = L^*Lf = \Pi_{\mathcal{H}}(f) = f .$$

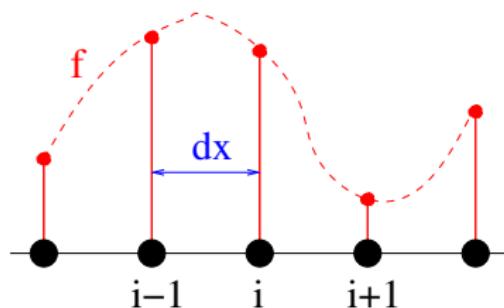
As a conclusion $K = L^*$ is the reproducing kernel of \mathcal{H} . \square

Example



$$L^* = \begin{pmatrix} 0.88 & -0.12 & 0.08 & -0.32 & -0.52 \\ -0.12 & 0.88 & 0.08 & -0.32 & -0.52 \\ 0.08 & 0.08 & 0.28 & -0.12 & -0.32 \\ -0.32 & -0.32 & -0.12 & 0.48 & 0.28 \\ -0.52 & -0.52 & -0.32 & 0.28 & 1.08 \end{pmatrix}$$

Interpretation of the Laplacian



$$\begin{aligned}\Delta f(x) &= f''(x) \\ &\sim \frac{f'(x + dx/2) - f'(x - dx/2)}{dx} \\ &\sim \frac{f(x + dx) - f(x) - f(x) + f(x - dx)}{dx^2} \\ &= \frac{f_{i-1} + f_{i+1} - 2f(i)}{dx^2} \\ &= -\frac{Lf(i)}{dx^2}.\end{aligned}$$

Interpretation of regularization

For $f = [0, 1] \rightarrow \mathbb{R}$ and $x_i = i/m$, we have:

$$\begin{aligned}\Omega(f) &= \sum_{i=1}^m \left(f\left(\frac{i+1}{m}\right) - f\left(\frac{i}{m}\right) \right)^2 \\ &\sim \sum_{i=1}^m \left(\frac{1}{m} \times f'\left(\frac{i}{m}\right) \right)^2 \\ &= \frac{1}{m} \times \frac{1}{m} \sum_{i=1}^m f'\left(\frac{i}{m}\right)^2 \\ &\sim \frac{1}{m} \int_0^1 f'(t)^2 dt.\end{aligned}$$

Outline

3 Kernels and Graphs

- Kernels for graphs
- **Kernels on graphs**
 - Motivation
 - Graph distance and p.d. kernels
 - Construction by regularization
 - **The diffusion kernel**
 - Harmonic analysis on graphs
 - Applications

Motivation

- Consider the normalized Gaussian kernel on \mathbb{R}^d :

$$K_t(\mathbf{x}, \mathbf{x}') = \frac{1}{(4\pi t)^{\frac{d}{2}}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{4t}\right).$$

- In order to transpose it to the graph, replacing the Euclidean distance by the shortest-path distance does not work.
- In this section we provide a characterization of the Gaussian kernel as the **solution of a partial differential equation** involving the Laplacian, which we can transpose to the graph: the **diffusion equation**.
- The solution of the discrete diffusion equation will be called the **diffusion kernel** or **heat kernel**.

The diffusion equation

Lemma

For any $\mathbf{x}_0 \in \mathbb{R}^d$, the function:

$$K_{\mathbf{x}_0}(\mathbf{x}, t) = K_t(\mathbf{x}_0, \mathbf{x}) = \frac{1}{(4\pi t)^{\frac{d}{2}}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_0\|^2}{4t}\right)$$

is solution of the *diffusion equation*:

$$\frac{\partial}{\partial t} K_{\mathbf{x}_0}(\mathbf{x}, t) = \Delta K_{\mathbf{x}_0}(\mathbf{x}, t)$$

with initial condition $K_{\mathbf{x}_0}(\mathbf{x}, 0) = \delta_{\mathbf{x}_0}(\mathbf{x})$

(proof by direct computation).

Discrete diffusion equation

For finite-dimensional $f_t \in \mathbb{R}^m$, the diffusion equation becomes:

$$\frac{\partial}{\partial t} f_t = -L f_t$$

which admits the following solution:

$$f_t = f_0 e^{-tL}$$

with

$$e^{-tL} = I - tL + \frac{t^2}{2!} L^2 - \frac{t^3}{3!} L^3 + \dots$$

Diffusion kernel (Kondor and Lafferty, 2002)

This suggest to consider:

$$K = e^{-tL}$$

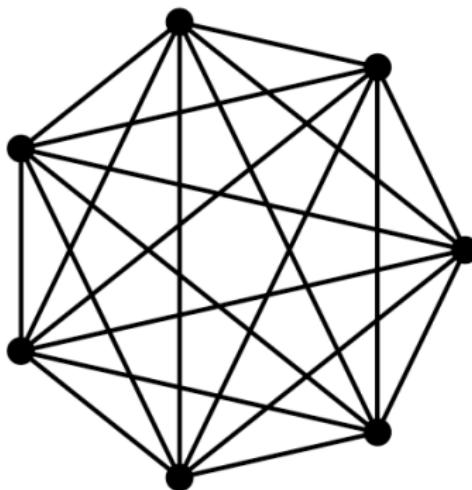
which is indeed symmetric positive semi-definite because if we write:

$$L = \sum_{i=1}^m \lambda_i u_i u_i^\top \quad (\lambda_i \geq 0)$$

we obtain:

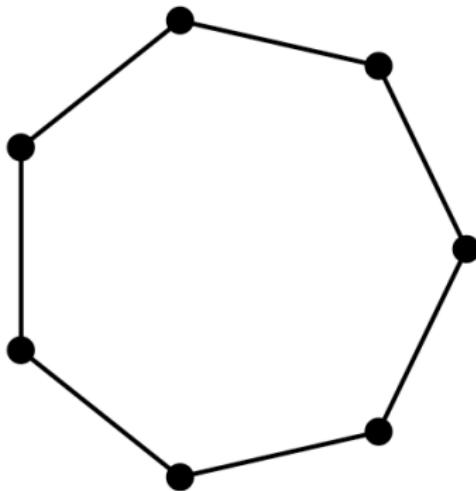
$$K = e^{-tL} = \sum_{i=1}^m e^{-t\lambda_i} u_i u_i^\top$$

Example: complete graph



$$K_{i,j} = \begin{cases} \frac{1+(m-1)e^{-tm}}{m} & \text{for } i = j, \\ \frac{1-e^{-tm}}{m} & \text{for } i \neq j. \end{cases}$$

Example: closed chain



$$K_{i,j} = \frac{1}{m} \sum_{\nu=0}^{m-1} \exp \left[-2t \left(1 - \cos \frac{2\pi\nu}{m} \right) \right] \cos \frac{2\pi\nu(i-j)}{m}.$$

Outline

3 Kernels and Graphs

- Kernels for graphs
- **Kernels on graphs**
 - Motivation
 - Graph distance and p.d. kernels
 - Construction by regularization
 - The diffusion kernel
- **Harmonic analysis on graphs**
- Applications

Motivation

- In this section we show that the diffusion and Laplace kernels can be interpreted in the **frequency domain** of functions
- This shows that our strategy to design kernels on graphs was based on **(discrete) harmonic analysis** on the graph
- This follows the approach we developed for semigroup kernels!

Spectrum of the diffusion kernel

- Let $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_m$ be the eigenvalues of the Laplacian:

$$L = \sum_{i=1}^m \lambda_i u_i u_i^\top \quad (\lambda_i \geq 0)$$

- The diffusion kernel K_t is an **invertible** matrix because its eigenvalues are strictly positive:

$$K_t = \sum_{i=1}^m e^{-t\lambda_i} u_i u_i^\top$$

Norm in the diffusion RKHS

- Any function $f \in \mathbb{R}^m$ can be written as $f = K(K^{-1}f)$, therefore its norm in the diffusion RKHS is:

$$\|f\|_{K_t}^2 = (f^\top K^{-1}) K (K^{-1}f) = f^\top K^{-1} f.$$

- For $i = 1, \dots, m$, let:

$$\hat{f}_i = u_i^\top f$$

be the projection of f onto the eigenbasis of K .

- We then have:

$$\|f\|_{K_t}^2 = f^\top K^{-1} f = \sum_{i=1}^m e^{t\lambda_i} \hat{f}_i^2.$$

- This looks similar to $\int |\hat{f}(\omega)|^2 e^{\sigma^2 \omega^2} d\omega \dots$

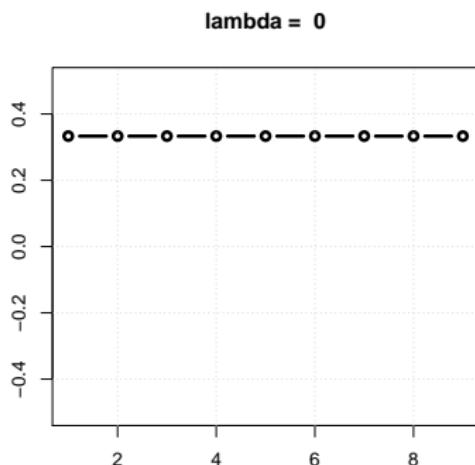
Discrete Fourier transform

Definition

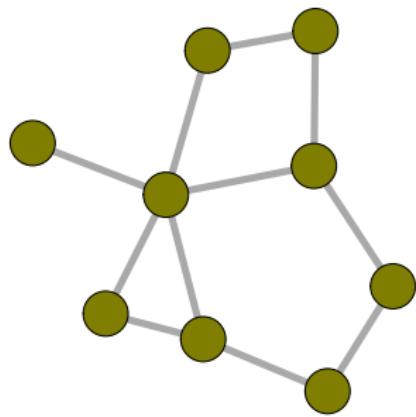
The vector $\hat{f} = (\hat{f}_1, \dots, \hat{f}_m)^\top$ is called the **discrete Fourier transform** of $f \in \mathbb{R}^n$

- The eigenvectors of the Laplacian are the discrete equivalent to the sine/cosine Fourier basis on \mathbb{R}^n .
- The eigenvalues λ_i are the equivalent to the frequencies ω^2
- Successive eigenvectors “oscillate” increasingly as eigenvalues get more and more negative.

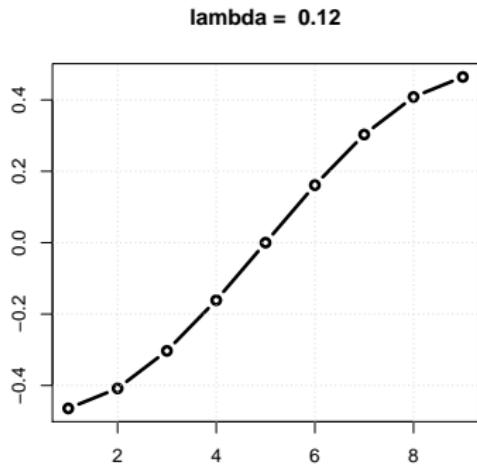
Examples



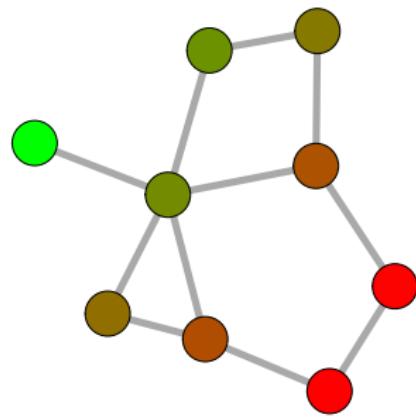
Lambda = 0



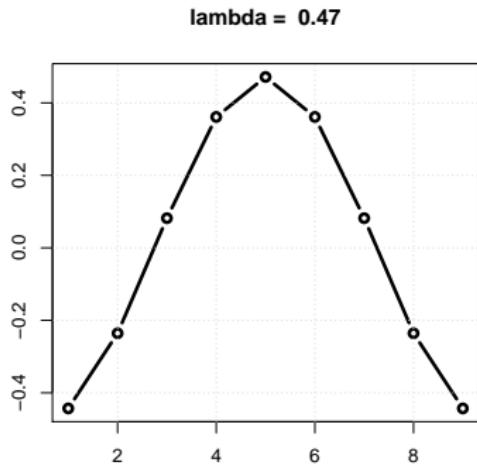
Examples



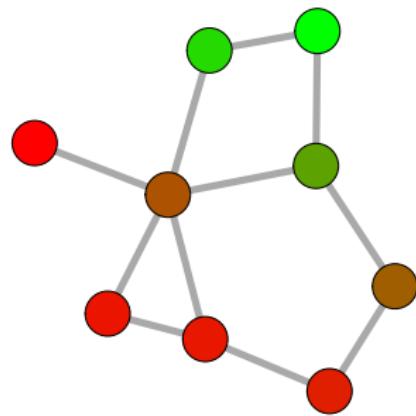
Lambda = 0.76



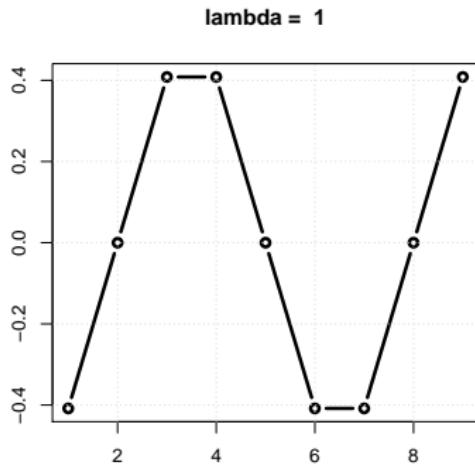
Examples



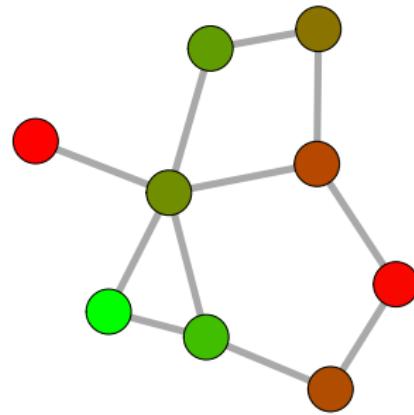
Lambda = 0.83



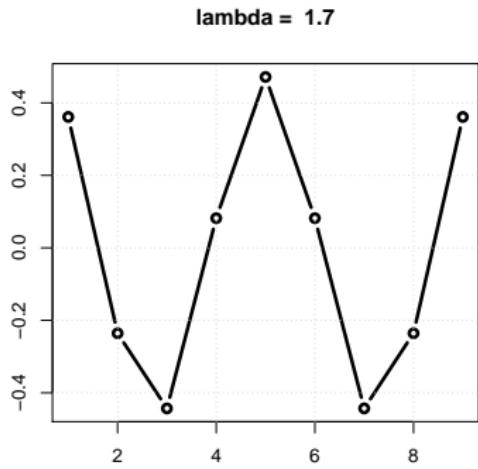
Examples



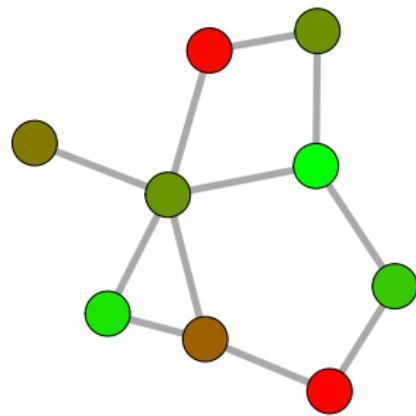
Lambda = 1.3



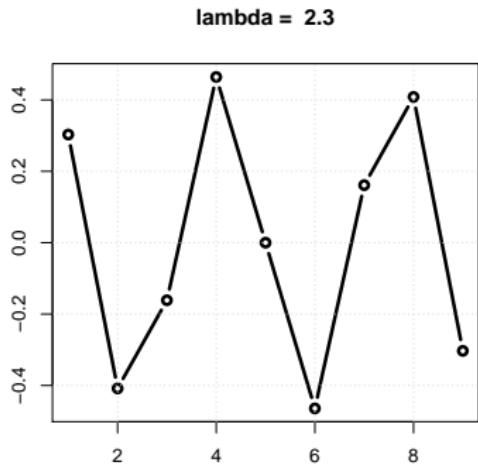
Examples



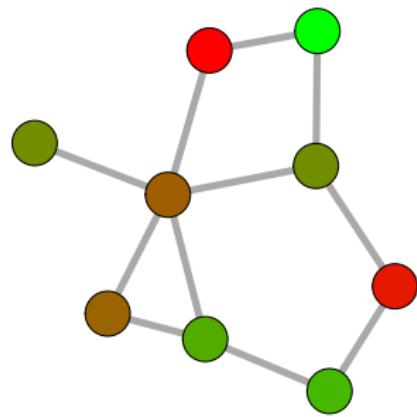
Lambda = 2.2



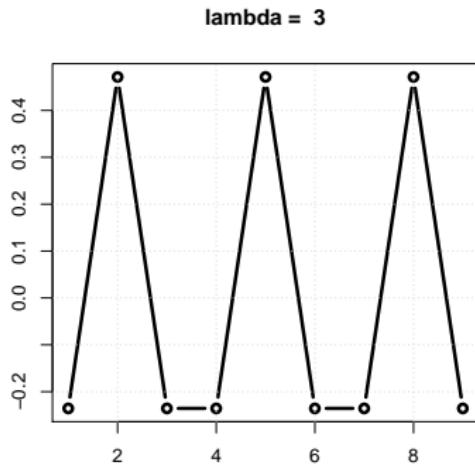
Examples



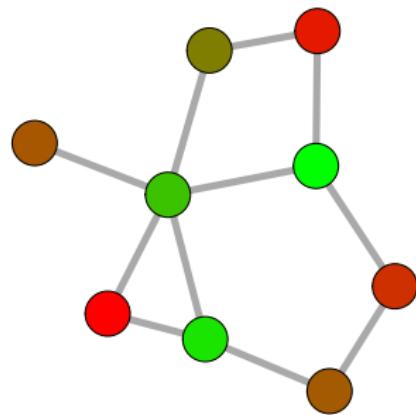
Lambda = 2.8



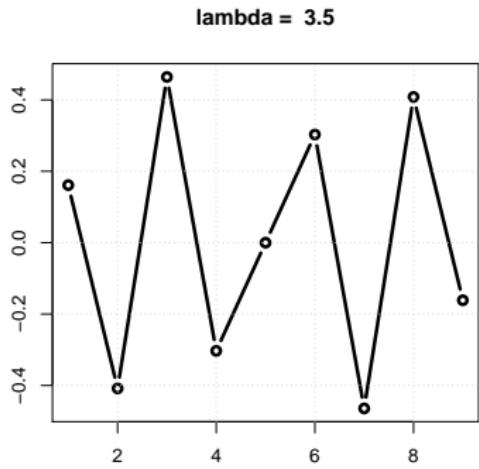
Examples



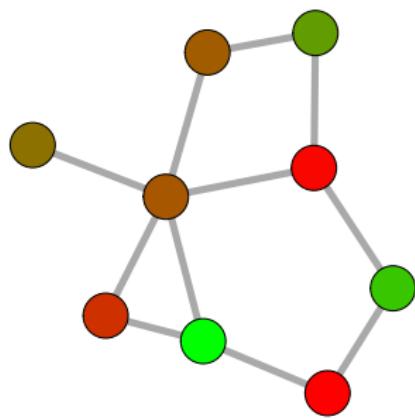
Lambda = 3.6



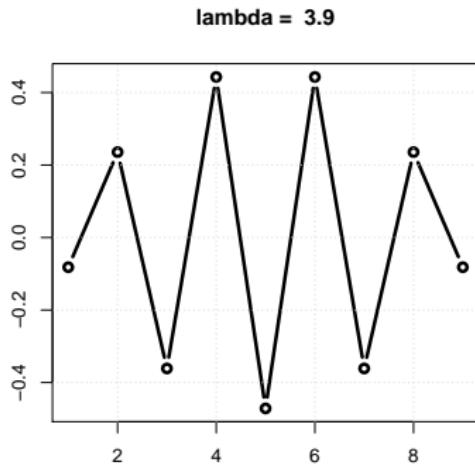
Examples



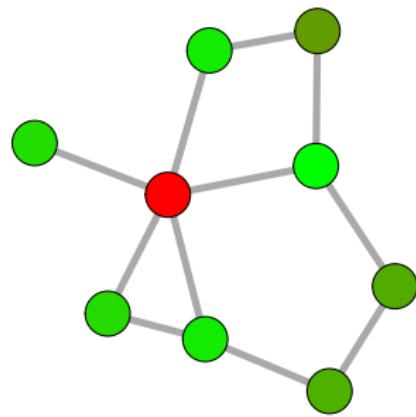
Lambda = 4.2



Examples



Lambda = 6.3



Generalization

This observation suggests to define a whole family of kernels:

$$K_r = \sum_{i=1}^m r(\lambda_i) u_i u_i^\top$$

associated with the following RKHS norms:

$$\|f\|_{K_r}^2 = \sum_{i=1}^m \frac{\hat{f}_i^2}{r(\lambda_i)}$$

where $r : \mathbb{R}^+ \rightarrow \mathbb{R}_*^+$ is a **non-increasing** function.

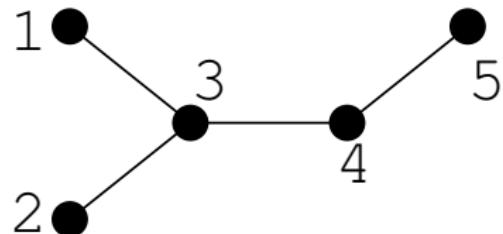
Example : regularized Laplacian

$$r(\lambda) = \frac{1}{\lambda + \epsilon}, \quad \epsilon > 0$$

$$K = \sum_{i=1}^m \frac{1}{\lambda_i + \epsilon} u_i u_i^\top = (L + \epsilon I)^{-1}$$

$$\| f \|_K^2 = f^\top K^{-1} f = \sum_{i \sim j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 + \epsilon \sum_{i=1}^m f(\mathbf{x}_i)^2.$$

Example



$$(L + I)^{-1} = \begin{pmatrix} 0.60 & 0.10 & 0.19 & 0.08 & 0.04 \\ 0.10 & 0.60 & 0.19 & 0.08 & 0.04 \\ 0.19 & 0.19 & 0.38 & 0.15 & 0.08 \\ 0.08 & 0.08 & 0.15 & 0.46 & 0.23 \\ 0.04 & 0.04 & 0.08 & 0.23 & 0.62 \end{pmatrix}$$

Outline

3 Kernels and Graphs

- Kernels for graphs
- **Kernels on graphs**
 - Motivation
 - Graph distance and p.d. kernels
 - Construction by regularization
 - The diffusion kernel
 - Harmonic analysis on graphs
- Applications

Applications 1: graph partitioning

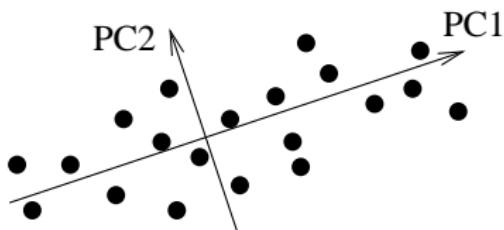
- A classical relaxation of graph partitioning is:

$$\min_{f \in \mathbb{R}^{\mathcal{X}}} \sum_{i \sim j} (f_i - f_j)^2 \quad \text{s.t. } \sum_i f_i^2 = 1$$

- This can be rewritten

$$\max_f \sum_i f_i^2 \text{ s.t. } \|f\|_{\mathcal{H}} \leq 1$$

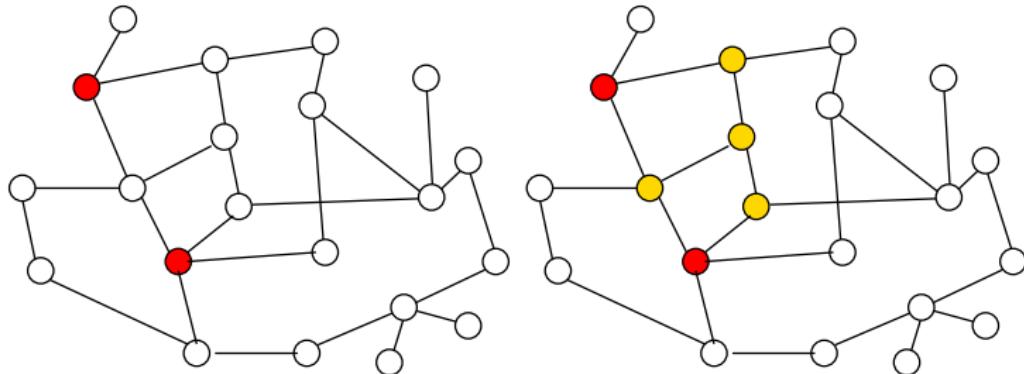
- This is **principal component analysis** in the RKHS (“kernel PCA”)



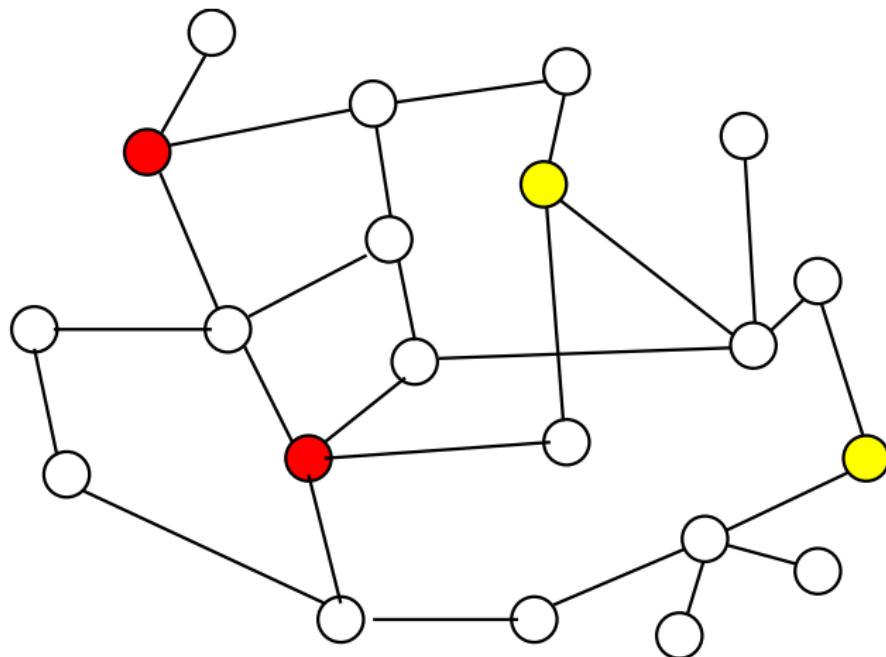
Applications 2: search on a graph

- Let x_1, \dots, x_q be a set of q nodes (the **query**). How to find “similar” nodes (and rank them)?
- One solution:

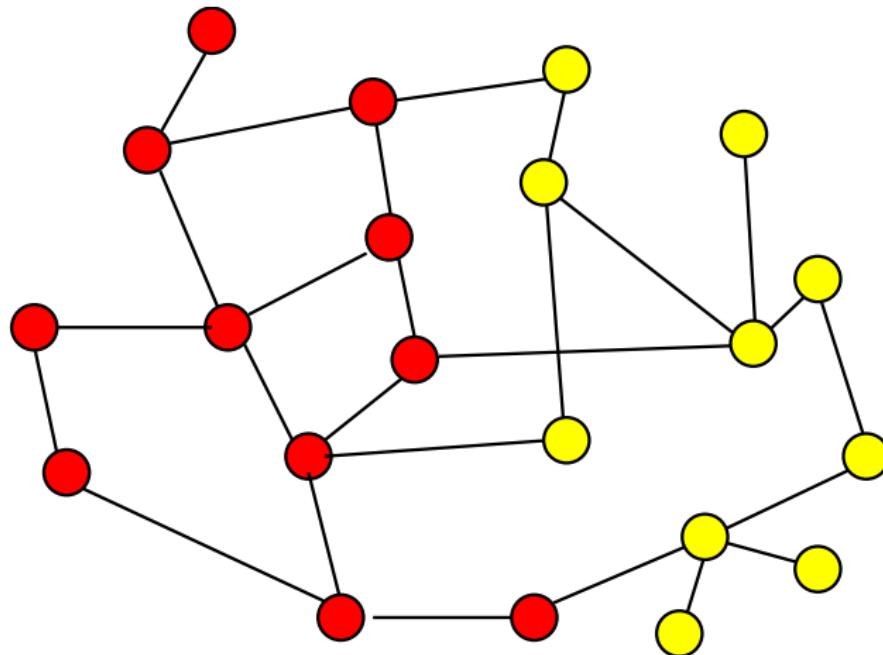
$$\min_f \|f\|_{\mathcal{H}} \quad \text{s.t.} \quad f(x_i) \geq 1 \text{ for } i = 1, \dots, q.$$



Application 3: Semi-supervised learning



Application 3: Semi-supervised learning



Characterizing probabilities with kernels

Introduction

- We have seen how to represent each individual data-point by an embedding in some feature space.
- This allows to compare data points by evaluating the kernel.
- Now we are interested in comparing two or more *sets* of data-points, or more generally *distributions* of data points.

Motivation I: Comparing two distributions

- Data: Samples from unknown distributions P and Q .
- Goal: do P and Q differ?



$\sim P$



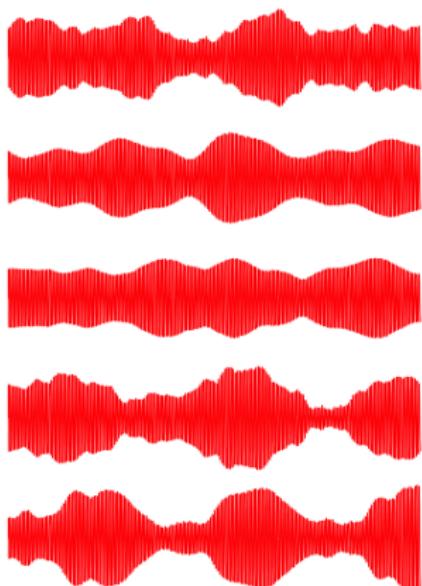
$\sim Q$

Differences between dogs and fish.

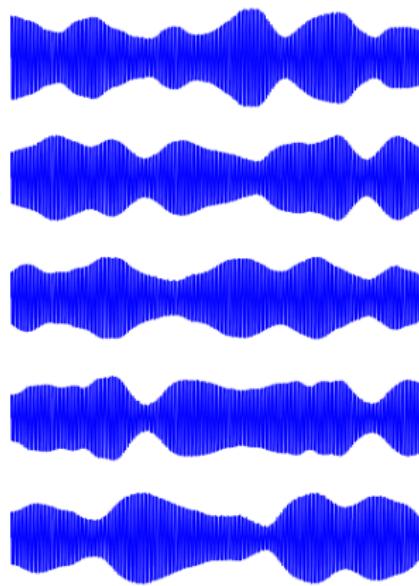
Motivation I: Comparing two distributions

- Data: Samples from unknown distributions P and Q .
- Goal: do P and Q differ?

Samples from P

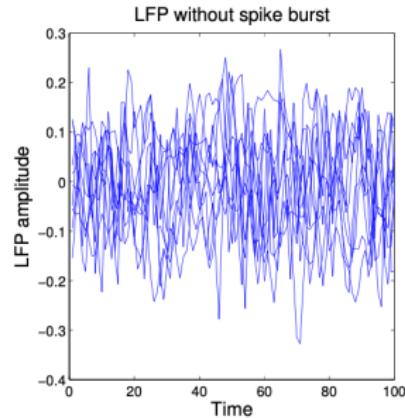
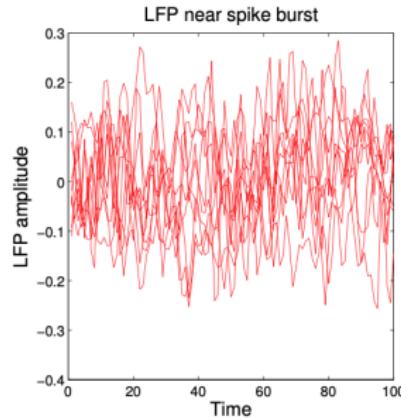


Samples from Q



Motivation I: Comparing two distributions

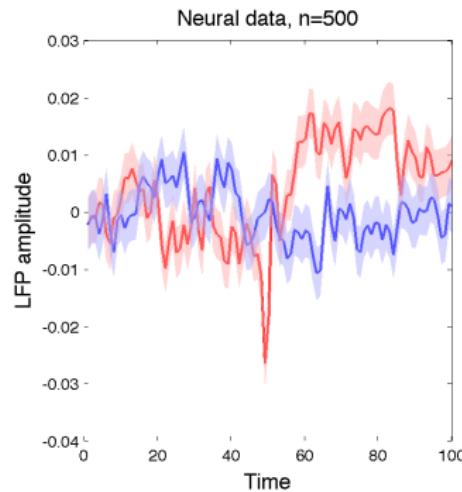
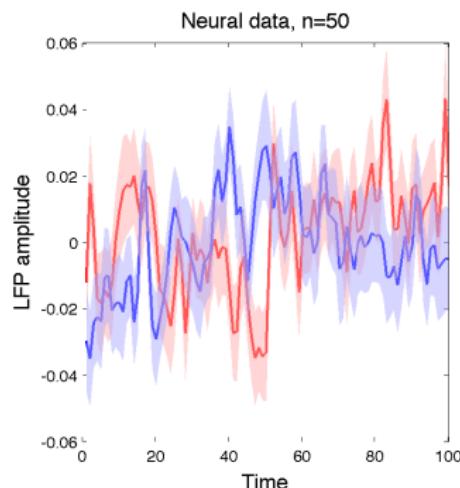
- Data: Samples from unknown distributions \mathbb{P} and \mathbb{Q} .
- Goal: do \mathbb{P} and \mathbb{Q} differ?



Difference in brain signals: Do local field potential (LFP) signals change when measured near a spike burst?

Motivation I: Comparing two distributions

- Data: Samples from unknown distributions \mathbb{P} and \mathbb{Q} .
- Goal: do \mathbb{P} and \mathbb{Q} differ?



Difference in brain signals: Do local field potential (LFP) signals change when measured near a spike burst?

Comparing the means?

Motivation II: Detecting dependence

X₁: Honourable senators, I have a question for the Leader of the Government in the Senate with regard to the support funding to farmers that has been announced. Most farmers have not received any money yet.

X₂: No doubt there is great pressure on provincial and municipal governments in relation to the issue of child care, but the reality is that there have been no cuts to child care funding from the federal government to the provinces. In fact, we have increased federal investments for early childhood development.



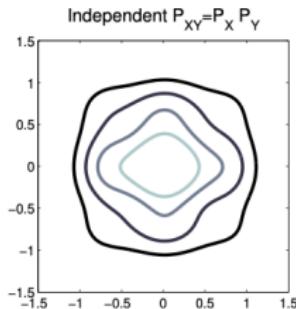
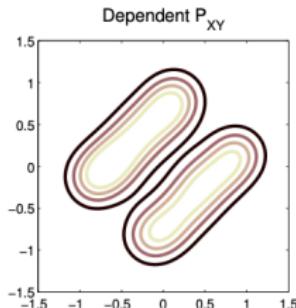
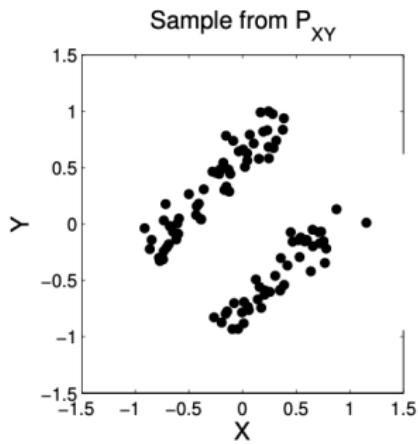
Y₁: Honorables sénateurs, ma question s'adresse au leader du gouvernement au Sénat et concerne l'aide financière qu'on a annoncée pour les agriculteurs. La plupart des agriculteurs n'ont encore rien reu de cet argent.

Y₂: Il est évident que les ordres de gouvernements provinciaux et municipaux subissent de fortes pressions en ce qui concerne les services de garde, mais le gouvernement n'a pas réduit le financement qu'il verse aux provinces pour les services de garde. Au contraire, nous avons augmenté le financement fédéral pour le développement des jeunes enfants.

...

...

Motivation II: Detecting dependence

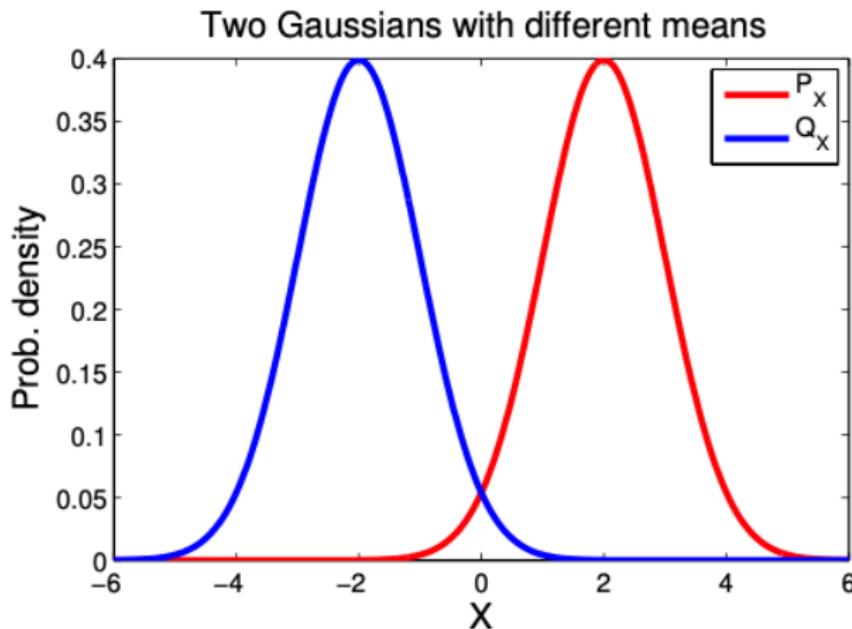


Outline

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
- 3 Kernels and Graphs
- 4 Characterizing probabilities with kernels
 - Kernel mean embedding
 - The Maximum Mean Discrepancy
 - Characteristic kernels
- 5 Open Problems and Research Topics

Feature mean difference

- Simple example: Samples from 2 Gaussians with same variance but different means.
- Idea: Look at difference in *means of features* of the samples.



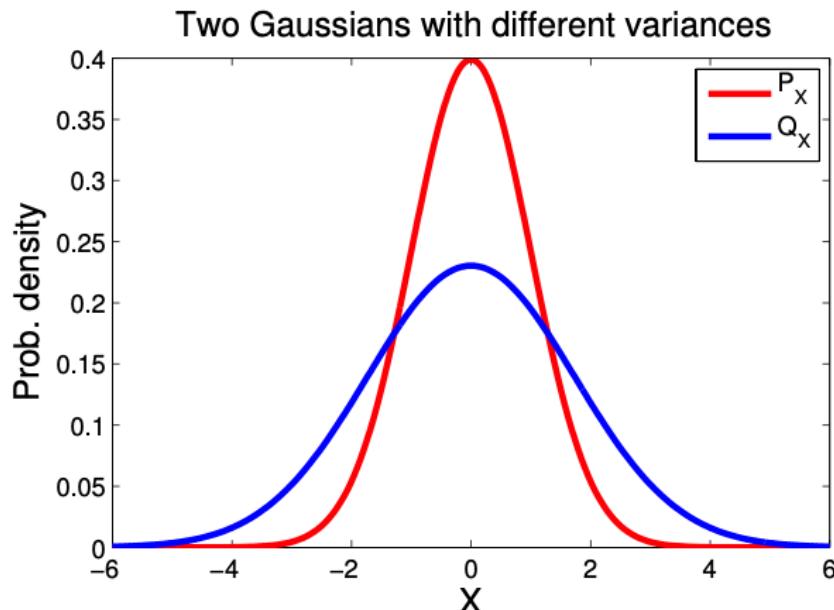
Compare

$$\hat{\mu}_P = \frac{1}{N} \sum_{i=1}^N x_i,$$

$$\hat{\mu}_Q = \frac{1}{M} \sum_{j=1}^M y_j$$

Feature mean difference

- Simple example: Samples from 2 Gaussians with same mean but different variances.
- Idea: Look at difference in *means of features* of the samples. Here $\varphi(x) = (x, x^2)$.



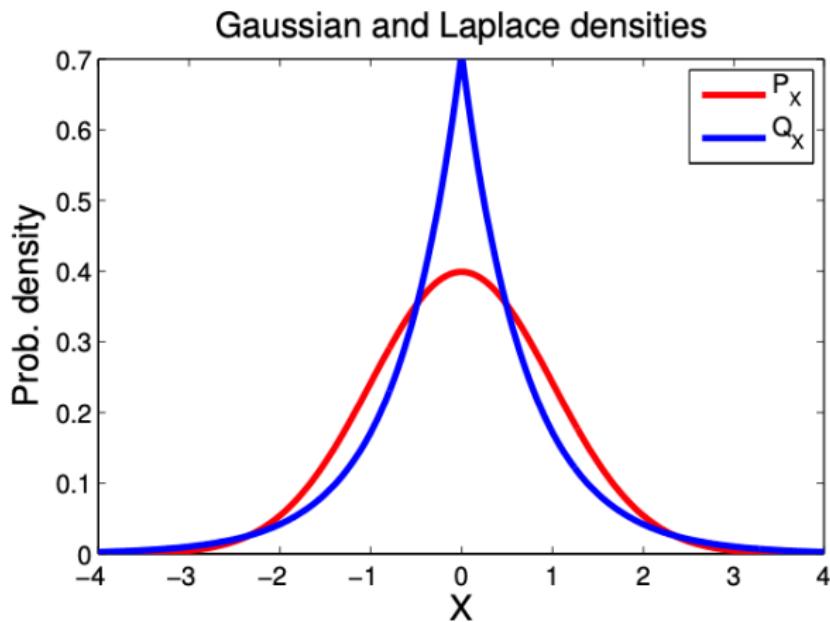
Compare

$$\hat{\mu}_{\mathbb{P}} = \frac{1}{N} \sum_{i=1}^N \varphi(x_i),$$

$$\hat{\mu}_{\mathbb{Q}} = \frac{1}{M} \sum_{j=1}^M \varphi(y_j)$$

Feature mean difference

- Simple example: Centered Gaussian and Laplace distributions: same mean and variance.
- Idea: Look at difference in *means of high order features* of the samples: $\varphi(x) = (x, x^2, \dots)$ (*RKHS*).



Compare

$$\hat{\mu}_{\mathbb{P}} = \frac{1}{N} \sum_{i=1}^N \varphi(x_i),$$

$$\hat{\mu}_{\mathbb{Q}} = \frac{1}{M} \sum_{j=1}^M \varphi(y_j)$$

Kernel Mean Embedding

Definition

Given a kernel K defined on a topological set \mathcal{X} with corresponding RKHS \mathcal{H} , the mean embedding of a *Borel* probability distribution \mathbb{P} on \mathcal{X} is the function $\mu_{\mathbb{P}} : \mathcal{X} \rightarrow \mathbb{R}$ in \mathcal{H} defined as

$$\mu_{\mathbb{P}}(y) := \mathbb{E}_{X \sim \mathbb{P}}[K(X, y)]$$

Kernel Mean Embedding

Definition

Given a kernel K defined on a topological set \mathcal{X} with corresponding RKHS \mathcal{H} , the mean embedding of a *Borel* probability distribution \mathbb{P} on \mathcal{X} is the function $\mu_{\mathbb{P}} : \mathcal{X} \rightarrow \mathbb{R}$ in \mathcal{H} defined as

$$\mu_{\mathbb{P}}(y) := \mathbb{E}_{X \sim \mathbb{P}}[K(X, y)]$$

- For any x, x' in \mathcal{X} ,

$$K(x, x') = \langle K_x, K_{x'} \rangle_{\mathcal{H}},$$

- The **kernel trick**:

For any $f \in \mathcal{H}$ and $x \in \mathcal{X}$,

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

Kernel Mean Embedding

Definition

Given a kernel K defined on a topological set \mathcal{X} with corresponding RKHS \mathcal{H} , the mean embedding of a *Borel* probability distribution \mathbb{P} on \mathcal{X} is the function $\mu_{\mathbb{P}} : \mathcal{X} \rightarrow \mathbb{R}$ in \mathcal{H} defined as

$$\mu_{\mathbb{P}}(y) := \mathbb{E}_{X \sim \mathbb{P}}[K(X, y)]$$

- For any x, x' in \mathcal{X} ,
- For any Borel measure \mathbb{P} and \mathbb{Q} ,

$$K(x, x') = \langle K_x, K_{x'} \rangle_{\mathcal{H}},$$

$$\mathbb{E}_{(X, Y) \sim \mathbb{P}, \mathbb{Q}} K(X, Y) = \langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}},$$

- The **kernel trick**:
For any $f \in \mathcal{H}$ and $x \in \mathcal{X}$,
- The **generalized kernel trick**:
For any $f \in \mathcal{H}$ and Borel measure \mathbb{P} ,

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

$$\mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$$

Kernel Mean Embedding

Kernel Mean Embedding

The kernel mean embedding: $\mu_{\mathbb{P}} = \mathbb{E}_{X \sim \mathbb{P}}[K_X]$

The generalized kernel trick: $\mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$.

Kernel Mean Embedding

Kernel Mean Embedding

The kernel mean embedding: $\mu_{\mathbb{P}} = \mathbb{E}_{X \sim \mathbb{P}}[K_X]$

The generalized kernel trick: $\mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$.

- Mean embedding $\mu_{\mathbb{P}}$ summarizes \mathbb{P} :
Can compute expectations under \mathbb{P} of
all functions in \mathcal{H} using $\mu_{\mathbb{P}}$.
- In practice, you can estimate $\mu_{\mathbb{P}}$ using
 N i.i.d. samples from \mathbb{P} :

$$\hat{\mu}_{\mathbb{P}}(x) = \frac{1}{N} \sum_{i=1}^N K(X_i, x), \quad X_i \stackrel{i.i.d.}{\sim} \mathbb{P}$$

Kernel Mean Embedding

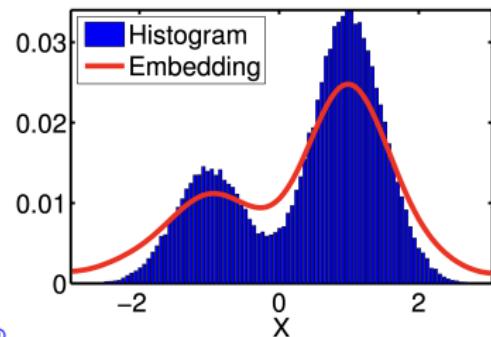
Kernel Mean Embedding

The kernel mean embedding: $\mu_{\mathbb{P}} = \mathbb{E}_{X \sim \mathbb{P}}[K_X]$

The generalized kernel trick: $\mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$.

- **Mean embedding $\mu_{\mathbb{P}}$ summarizes \mathbb{P} :**
Can compute expectations under \mathbb{P} of all functions in \mathcal{H} using $\mu_{\mathbb{P}}$.
- In practice, you can estimate $\mu_{\mathbb{P}}$ using N i.i.d. samples from \mathbb{P} :

$$\hat{\mu}_{\mathbb{P}}(x) = \frac{1}{N} \sum_{i=1}^N K(X_i, x), \quad X_i \stackrel{i.i.d.}{\sim} \mathbb{P}$$



Kernel Mean Embedding

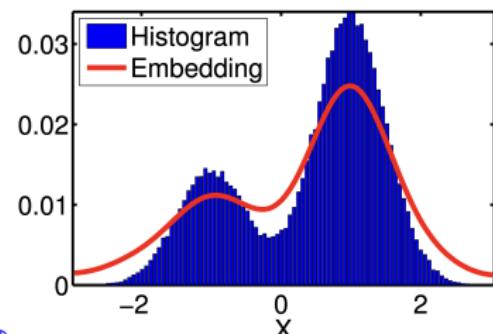
Kernel Mean Embedding

The kernel mean embedding: $\mu_{\mathbb{P}} = \mathbb{E}_{X \sim \mathbb{P}}[K_X]$

The generalized kernel trick: $\mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$.

- **Mean embedding $\mu_{\mathbb{P}}$ summarizes \mathbb{P} :**
Can compute expectations under \mathbb{P} of all functions in \mathcal{H} using $\mu_{\mathbb{P}}$.
- In practice, you can estimate $\mu_{\mathbb{P}}$ using N i.i.d. samples from \mathbb{P} :

$$\hat{\mu}_{\mathbb{P}}(x) = \frac{1}{N} \sum_{i=1}^N K(X_i, x), \quad X_i \stackrel{i.i.d.}{\sim} \mathbb{P}$$



Does the mean embedding $\mu_{\mathbb{P}}$ exist? i.e. an element $\mu_{\mathbb{P}} \in \mathcal{H}$ such that

$$\mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}, \forall f \in \mathcal{H}.$$

Existence of mean embeddings

Proposition

Let \mathbb{P} be a Borel probability distribution on a set \mathcal{X} endowed with its Borel sigma algebra. Let K be a p.d. kernel defined on \mathcal{X} with corresponding RKHS \mathcal{H} . **Assume** that $\mathbb{E}_{X \sim \mathbb{P}}[\sqrt{K(X, X)}] < \infty$. Then there exists a unique element $\mu_{\mathbb{P}} \in \mathcal{H}$ such that

$$\mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}, \forall f \in \mathcal{H}.$$

In particular, for any $y \in \mathcal{X}$, it holds that:

$$\mu_{\mathbb{P}}(y) = \langle K_y, \mu_{\mathbb{P}} \rangle = \mathbb{E}_{X \sim \mathbb{P}}[K(X, y)].$$

Existence of mean embeddings

Proposition

Let \mathbb{P} be a Borel probability distribution on a set \mathcal{X} endowed with its Borel sigma algebra. Let K be a p.d. kernel defined on \mathcal{X} with corresponding RKHS \mathcal{H} . **Assume** that $\mathbb{E}_{X \sim \mathbb{P}}[\sqrt{K(X, X)}] < \infty$. Then there exists a unique element $\mu_{\mathbb{P}} \in \mathcal{H}$ such that

$$\mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}, \forall f \in \mathcal{H}.$$

In particular, for any $y \in \mathcal{X}$, it holds that:

$$\mu_{\mathbb{P}}(y) = \langle K_y, \mu_{\mathbb{P}} \rangle = \mathbb{E}_{X \sim \mathbb{P}}[K(X, y)].$$

Proof:

The linear form on \mathcal{H} : $T_{\mathbb{P}} f = \mathbb{E}_{X \sim \mathbb{P}}[f(X)]$ is bounded by assumption:

$$|T_{\mathbb{P}} f| \leq \mathbb{E}_{X \sim \mathbb{P}}[|f(X)|] = \mathbb{E}_{X \sim \mathbb{P}}[|\langle f, K_X \rangle_{\mathcal{H}}|] \leq \mathbb{E}_{X \sim \mathbb{P}}[\sqrt{K(X, X)} \|f\|_{\mathcal{H}}].$$

Hence, by Riesz's theorem, there exists $\mu_{\mathbb{P}} \in \mathcal{H}$ such that $T_{\mathbb{P}} f = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$.

Outline

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
- 3 Kernels and Graphs
- 4 Characterizing probabilities with kernels
 - Kernel mean embedding
 - **The Maximum Mean Discrepancy**
 - Characteristic kernels
- 5 Open Problems and Research Topics

Motivation: Comparing two distributions

- Data: Samples from unknown distributions P and Q .
- Goal: do P and Q differ?



$\sim P$



$\sim Q$

Differences between dogs and fish.

The Maximum Mean Discrepancy

The maximum mean discrepancy (MMD) is the RKHS distance between mean embeddings:

$$MMD^2(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}^2$$

The Maximum Mean Discrepancy

The maximum mean discrepancy (MMD) is the RKHS distance between mean embeddings:

$$\begin{aligned} MMD^2(\mathbb{P}, \mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}^2 \\ &= \langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} + \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} - 2 \langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \end{aligned}$$

The Maximum Mean Discrepancy

The maximum mean discrepancy (MMD) is the RKHS distance between mean embeddings:

$$\begin{aligned} MMD^2(\mathbb{P}, \mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}^2 \\ &= \langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} + \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} - 2 \langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\ &= \mathbb{E}_{X, X' \sim \mathbb{P} \otimes \mathbb{P}} [k(X, X')] + \mathbb{E}_{Y, Y' \sim \mathbb{Q} \otimes \mathbb{Q}} [k(Y, Y')] \\ &\quad - 2 \mathbb{E}_{X, Y \sim \mathbb{P} \otimes \mathbb{Q}} [k(X, Y)] \end{aligned}$$

The Maximum Mean Discrepancy

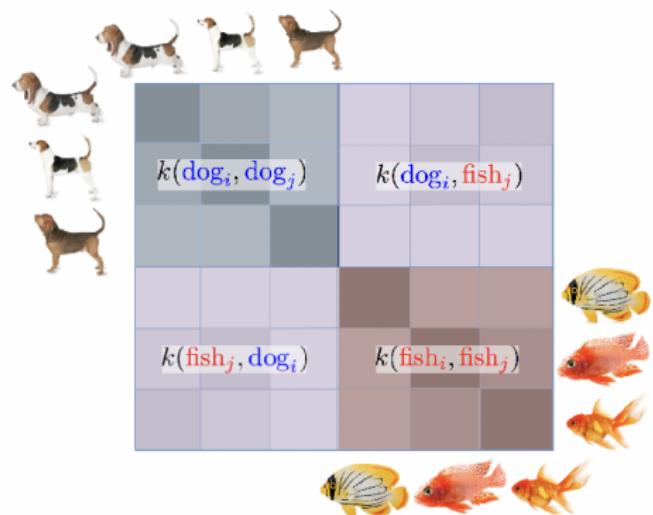
The maximum mean discrepancy (MMD) is the RKHS distance between mean embeddings:

$$\begin{aligned} MMD^2(\mathbb{P}, \mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}^2 \\ &= \langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} + \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} - 2 \langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\ &= \mathbb{E}_{X, X' \sim \mathbb{P} \otimes \mathbb{P}} [k(X, X')] + \mathbb{E}_{Y, Y' \sim \mathbb{Q} \otimes \mathbb{Q}} [k(Y, Y')] \\ &\quad - 2 \mathbb{E}_{X, Y \sim \mathbb{P} \otimes \mathbb{Q}} [k(X, Y)] \end{aligned}$$

- **Intra-similarity** terms : $\mathbb{E}_{X, X' \sim \mathbb{P} \otimes \mathbb{P}} [k(X, X')]$ and $\mathbb{E}_{Y, Y' \sim \mathbb{Q} \otimes \mathbb{Q}} [k(Y, Y')]$.
- **Inter-similarity** term: $\mathbb{E}_{X, Y \sim \mathbb{P} \otimes \mathbb{Q}} [k(X, Y)]$.
- In general, MMD is a semi-metric: ($MMD(\mathbb{P}, \mathbb{Q}) = 0 \Rightarrow \mathbb{P} = \mathbb{Q}$).
- For some kernels (called **characteristic kernels**), MMD is a metric ($MMD(\mathbb{P}, \mathbb{Q}) = 0 \iff \mathbb{P} = \mathbb{Q}$).
- From now on, we assume MMD is a metric. Later, we'll say more about **characteristic kernels**.

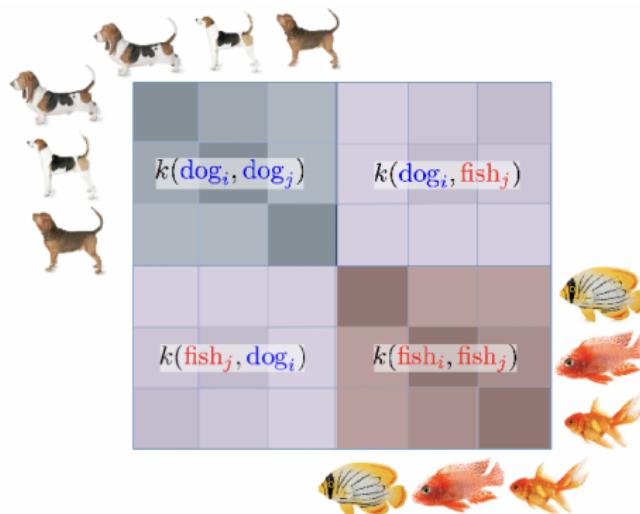
Unbiased estimation of the MMD

- Data: i.i.d. samples from \mathbb{P} and \mathbb{Q}



Unbiased estimation of the MMD

- Data: i.i.d. samples from \mathbb{P} and \mathbb{Q}



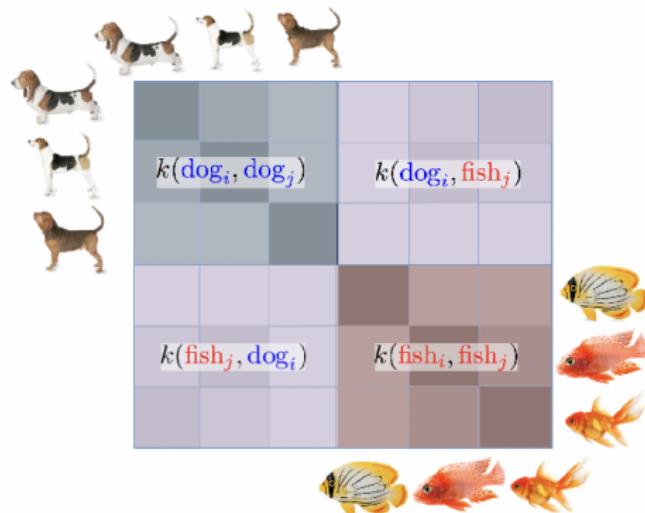
Biased estimate of the MMD^2 :

$$\widehat{MMD^2}(\mathbb{P}, \mathbb{Q}) = \frac{1}{N^2} \sum_{i,j} K(\text{dog}_i, \text{dog}_j) + \frac{1}{M^2} \sum_{i,j} K(\text{fish}_i, \text{fish}_j)$$

$$- \frac{2}{NM} \sum_{i,j} k(\text{dog}_i, \text{fish}_j)$$

Unbiased estimation of the MMD

- Data: i.i.d. samples from \mathbb{P} and \mathbb{Q}



Unbiased estimate of the MMD^2 :

$$\widehat{MMD^2}(\mathbb{P}, \mathbb{Q}) = \frac{1}{N(N-1)} \sum_{i \neq j} K(\text{dog}_i, \text{dog}_j) + \frac{1}{M(M-1)} \sum_{i \neq j} K(\text{fish}_i, \text{fish}_j) - \frac{2}{NM} \sum_{i,j} k(\text{dog}_i, \text{fish}_j)$$

MMD as an Integral Probability Metric

Integral Probability Metric

Let \mathcal{F} be a set of measurable functions. An integral probability metric associated to the class \mathcal{F} is a semi-metric defined as

$$\mathcal{D}_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)].$$

MMD as an Integral Probability Metric

Integral Probability Metric

Let \mathcal{F} be a set of measurable functions. An integral probability metric associated to the class \mathcal{F} is a semi-metric defined as

$$\mathcal{D}_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)].$$

- MMD obtained by choosing $\mathcal{F} = \{f \in \mathcal{H} | \|f\|_{\mathcal{H}} \leq 1\}$:

$$MMD(\mathbb{P}, \mathbb{Q}) := \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}} \leq 1}} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$

MMD as an Integral Probability Metric

Integral Probability Metric

Let \mathcal{F} be a set of measurable functions. An integral probability metric associated to the class \mathcal{F} is a semi-metric defined as

$$\mathcal{D}_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)].$$

- MMD obtained by choosing $\mathcal{F} = \{f \in \mathcal{H} | \|f\|_{\mathcal{H}} \leq 1\}$:

$$MMD(\mathbb{P}, \mathbb{Q}) := \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}} \leq 1}} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$

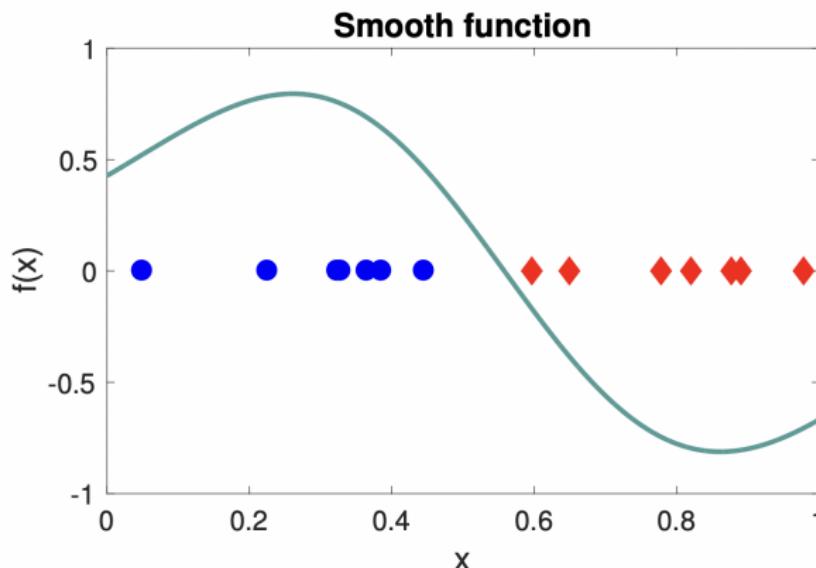
- Other choices for the set \mathcal{F} :

- Bounded continuous \rightarrow Dudley's metric.
- Bounded variations \rightarrow Kolmogorov metric.
- Bounded Lipschitz \rightarrow 1-Wasserstein distance.

MMD as an Integral Probability Metric

- MMD obtained by choosing $\mathcal{F} = \{f \in \mathcal{H} | \|f\|_{\mathcal{H}} \leq 1\}$:

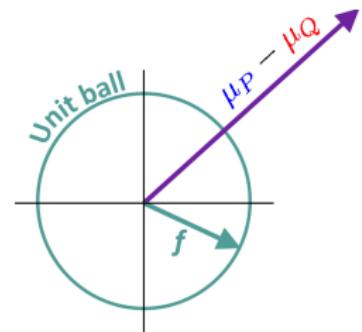
$$MMD(\mathbb{P}, \mathbb{Q}) = \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}} \leq 1}} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]$$



MMD as an Integral Probability Metric

- MMD obtained by choosing $\mathcal{F} = \{f \in \mathcal{H} | \|f\|_{\mathcal{H}} \leq 1\}$:

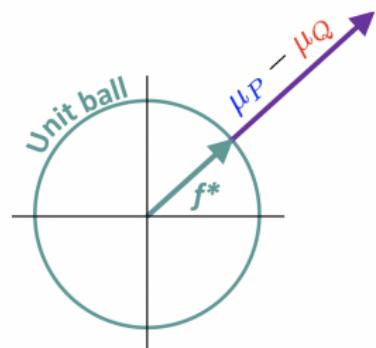
$$\begin{aligned} MMD(\mathbb{P}, \mathbb{Q}) &= \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}} \leq 1}} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)] \\ &= \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}} \leq 1}} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \end{aligned}$$



MMD as an Integral Probability Metric

- MMD obtained by choosing $\mathcal{F} = \{f \in \mathcal{H} | \|f\|_{\mathcal{H}} \leq 1\}$:

$$\begin{aligned} MMD(\mathbb{P}, \mathbb{Q}) &= \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}} \leq 1}} \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)] \\ &= \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}} \leq 1}} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\ &= \langle f^*, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\ &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \end{aligned}$$



$$f^* = \frac{\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}}{\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|}$$

f^* is called the
witness function

Outline

4 Characterizing probabilities with kernels

- Kernel mean embedding
- The Maximum Mean Discrepancy
 - Applications (I): Statistical testing using the MMD
 - Applications (II): Learning generative models
- Characteristic kernels

A statistical test using MMD

- Data: Samples x_1, \dots, x_N and y_1, \dots, y_N from unknown distributions \mathbb{P} and \mathbb{Q} .
- Goal: Is $\mathbb{P} = \mathbb{Q}$?

A statistical test using MMD

- Data: Samples x_1, \dots, x_N and y_1, \dots, y_N from unknown distributions \mathbb{P} and \mathbb{Q} .
- Goal: Is $\mathbb{P} = \mathbb{Q}$?

Empirical estimate of the MMD:

$$\widehat{MMD^2}(\mathbb{P}, \mathbb{Q}) = \frac{1}{N(N-1)} \sum_{i \neq j} K(x_i, x_j) + \frac{1}{N(N-1)} \sum_{i \neq j} K(y_i, y_j) - \frac{2}{N^2} \sum_{i,j} K(x_i, y_j)$$

A statistical test using MMD

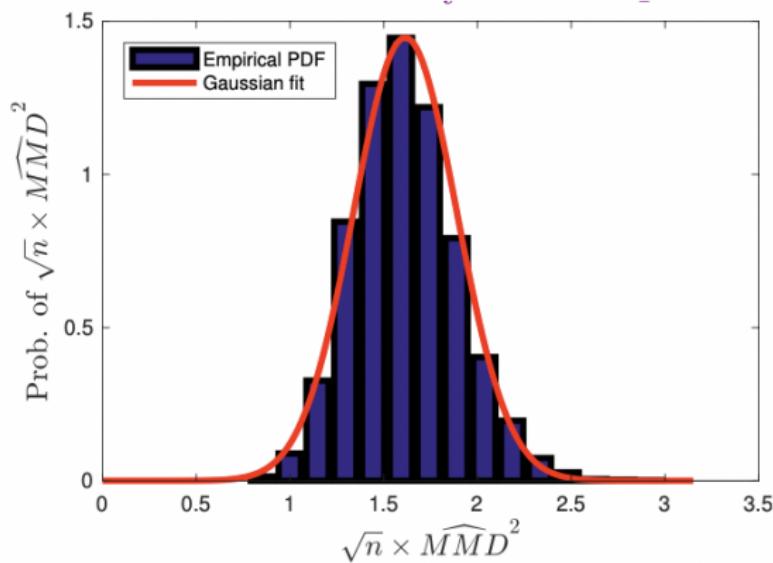
- Data: Samples x_1, \dots, x_N and y_1, \dots, y_N from unknown distributions \mathbb{P} and \mathbb{Q} .
- Goal: Is $\mathbb{P} = \mathbb{Q}$?

Empirical estimate of the MMD:

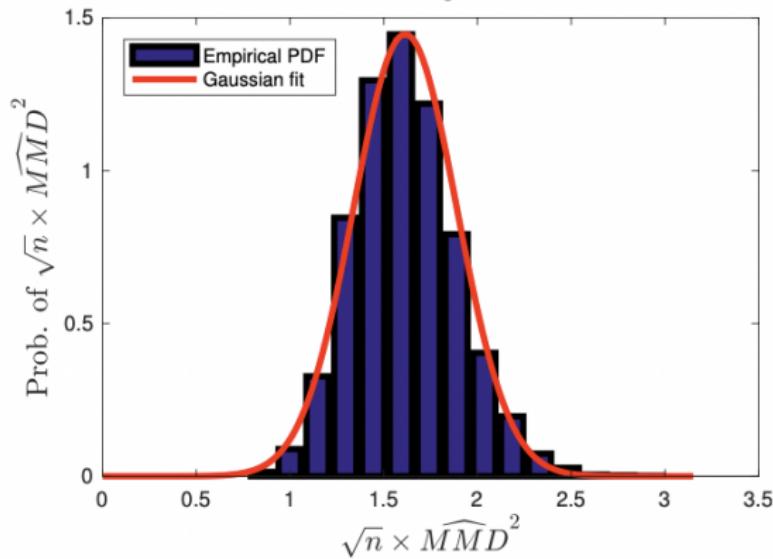
$$\widehat{MMD^2}(\mathbb{P}, \mathbb{Q}) = \frac{1}{N(N-1)} \sum_{i \neq j} K(x_i, x_j) + \frac{1}{N(N-1)} \sum_{i \neq j} K(y_i, y_j) - \frac{2}{N^2} \sum_{i,j} K(x_i, y_j)$$

- **Null hypothesis** h_0 when $\mathbb{P} = \mathbb{Q}$.
 $\widehat{MMD^2}(\mathbb{P}, \mathbb{Q})$ should be close to zero.
- **Alternative hypothesis** h_1 when $\mathbb{P} \neq \mathbb{Q}$.
 $\widehat{MMD^2}(\mathbb{P}, \mathbb{Q})$ should be far away from zero.
- What do close or far away mean here?

Behaviour of MMD when $\mathbb{P} \neq \mathbb{Q}$



Behaviour of MMD when $\mathbb{P} \neq \mathbb{Q}$

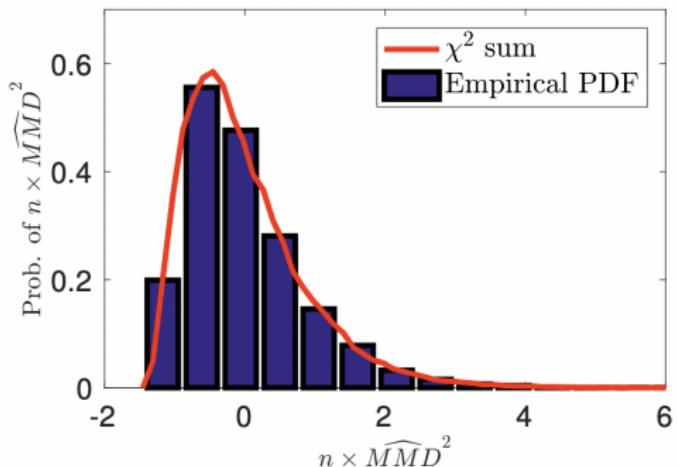


The statistic $\widehat{MMD^2}(\mathbb{P}, \mathbb{Q})$ is asymptotically normal [Gretton, 2006]:

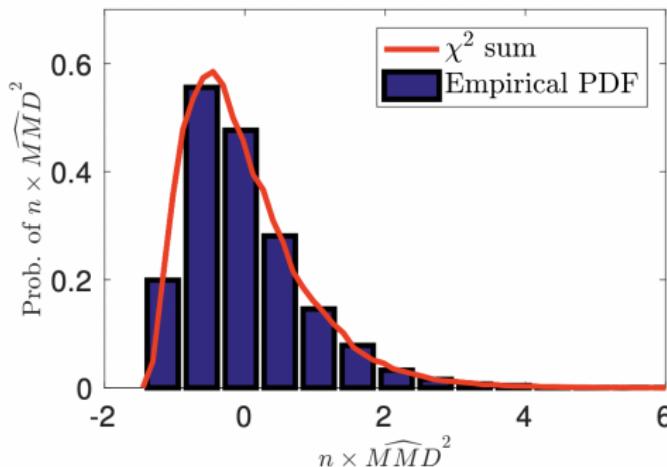
$$\frac{\sqrt{n}(\widehat{MMD^2}(\mathbb{P}, \mathbb{Q}) - MMD^2(\mathbb{P}, \mathbb{Q}))}{\sqrt{V(\mathbb{P}, \mathbb{Q})}} \rightarrow \mathcal{N}(0, 1).$$

where $V(\mathbb{P}, \mathbb{Q})$ is the asymptotic variance of $\sqrt{n} \times (\widehat{MMD^2}(\mathbb{P}, \mathbb{Q}))$.

Behaviour of MMD when $\mathbb{P} = \mathbb{Q}$



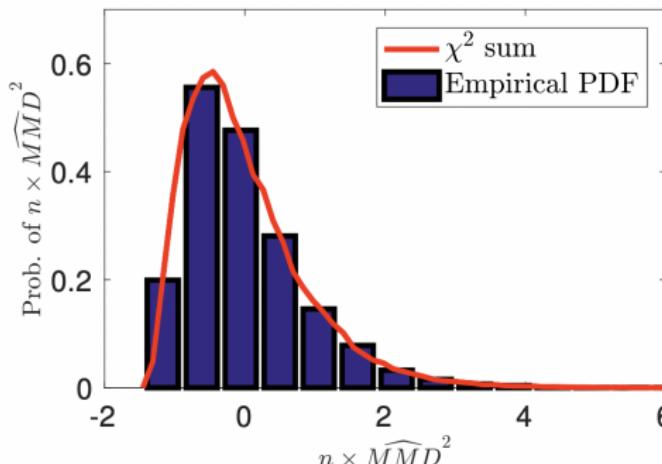
Behaviour of MMD when $\mathbb{P} = \mathbb{Q}$



$n\widehat{MMD}^2(\mathbb{P}, \mathbb{Q})$ has an asymptotic distribution [Gretton, 2006]:

$$n\widehat{MMD}^2(\mathbb{P}, \mathbb{Q}) \sim 2 \sum_{i=1}^{\infty} \lambda_i (z_i^2 - 1)$$

Behaviour of MMD when $\mathbb{P} = \mathbb{Q}$



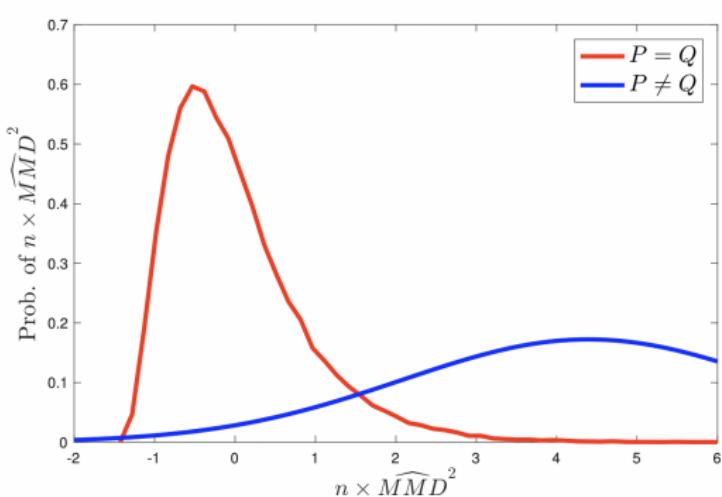
$n \widehat{MMD}^2(\mathbb{P}, \mathbb{Q})$ has an asymptotic distribution [Gretton, 2006]:

$$n \widehat{MMD}^2(\mathbb{P}, \mathbb{Q}) \sim 2 \sum_{i=1}^{\infty} \lambda_i (z_i^2 - 1)$$

- z_i are i.i.d. standard gaussians: $z_i \sim \mathcal{N}(0, 1)$
- λ_i are eigenvalues of the operator $f \mapsto \mathbb{E}_{X \sim \mathbb{P}}[\tilde{K}(X, X')f(X)]$
- \tilde{K} the centered kernel:

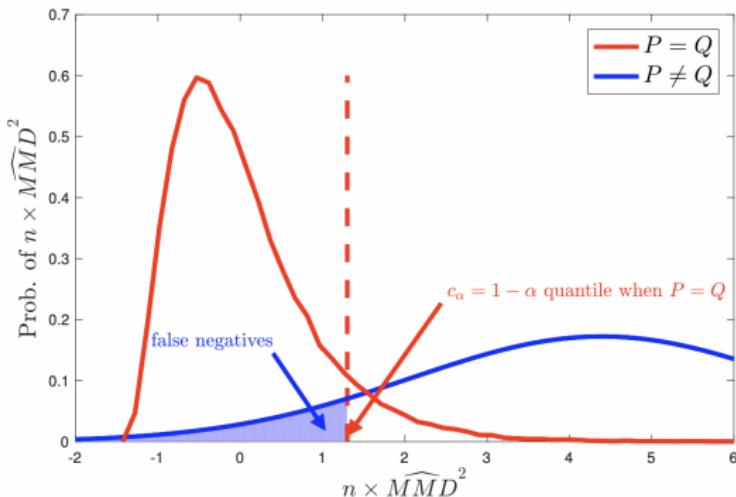
$$\tilde{K}(x, x') = \langle K(x, \cdot) - \mu_{\mathbb{P}}, K(x', \cdot) - \mu_{\mathbb{P}} \rangle_{\mathcal{H}}.$$

A statistical test using MMD



$$T_0 := n \widehat{MMD^2}(\mathbb{P}, \mathbb{Q}) \approx \begin{cases} n MMD^2(\mathbb{P}, \mathbb{Q}) + \sqrt{n} \mathcal{N}(0, V(\mathbb{P}, \mathbb{Q})), & \mathbb{P} \neq \mathbb{Q} \\ 2 \sum_{i=1}^{\infty} \lambda_i (z_i^2 - 1), & \mathbb{P} = \mathbb{Q}. \end{cases}$$

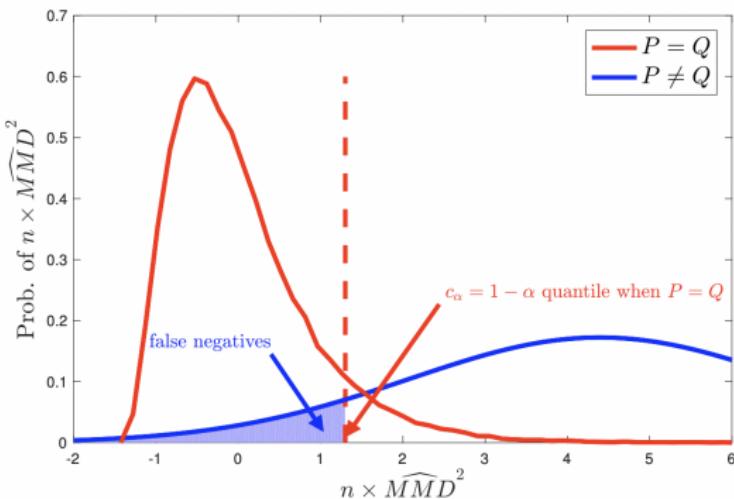
A statistical test using MMD



- Fix a significance level α and quantile c_α s.t. $\mathbb{P}(T_0 > c_\alpha | h_0) = \alpha$.
- If $T_0 \geq c_\alpha$, reject the null, i.e. ($\mathbb{P} = \mathbb{Q}$ unlikely)
- Otherwise, cannot reject ($\mathbb{P} = \mathbb{Q}$ is likely).

$$T_0 := n \widehat{\text{MMD}}^2(\mathbb{P}, \mathbb{Q}) \approx \begin{cases} n \text{MMD}^2(\mathbb{P}, \mathbb{Q}) + \sqrt{n} \mathcal{N}(0, V(\mathbb{P}, \mathbb{Q})), & \mathbb{P} \neq \mathbb{Q} \\ 2 \sum_{i=1}^{\infty} \lambda_i (z_i^2 - 1), & \mathbb{P} = \mathbb{Q}. \end{cases}$$

A statistical test using MMD



- Fix a significance level α and quantile c_α s.t. $\mathbb{P}(T_0 > c_\alpha | h_0) = \alpha$.
- If $T_0 \geq c_\alpha$, reject the null, i.e. ($\mathbb{P} = \mathbb{Q}$ unlikely)
- Otherwise, cannot reject ($\mathbb{P} = \mathbb{Q}$ is likely).

How can we tell if $T_0 := n\widehat{MMD}^2(\mathbb{P}, \mathbb{Q}) \geq c_\alpha$?

- Let T be a r.v. under the null distribution: $T \sim 2 \sum_{i=1}^{\infty} \lambda_i (z_i^2 - 1)$.
- If the p -value $p := \mathbb{P}_T(T > T_0) \leq \alpha$, then $T_0 \geq c_\alpha$.
- For T_1, \dots, T_J samples from the null: $p \approx |\{j | T_j \geq T_0\}| / J$.

Can use a permutation test to construct T_1, \dots, T_J .

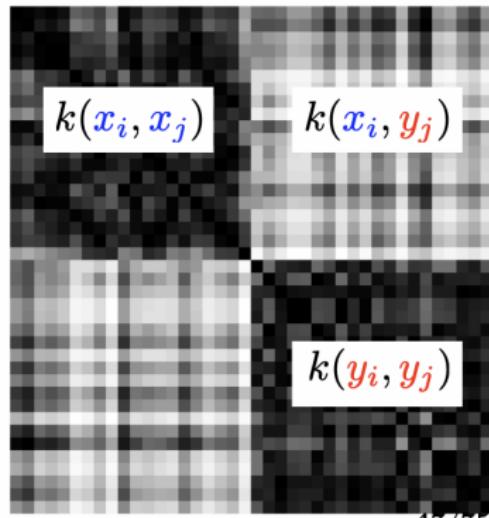
A statistical test using MMD

Original empirical MMD for dogs and fish:

$$X = \begin{bmatrix} \text{dog emoji}_1 & \text{dog emoji}_2 & \text{dog emoji}_3 & \dots \end{bmatrix}$$

$$Y = \begin{bmatrix} \text{fish emoji}_1 & \text{fish emoji}_2 & \text{fish emoji}_3 & \dots \end{bmatrix}$$

$$\widehat{MMD}^2 = \frac{1}{n(n-1)} \sum_{i \neq j} k(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(\mathbf{y}_i, \mathbf{y}_j) - \frac{2}{n^2} \sum_{i,j} k(\mathbf{x}_i, \mathbf{y}_j)$$



For each permutation j set $T_j = nMMD^2(\tilde{\mathbb{P}}, \tilde{\mathbb{Q}})$

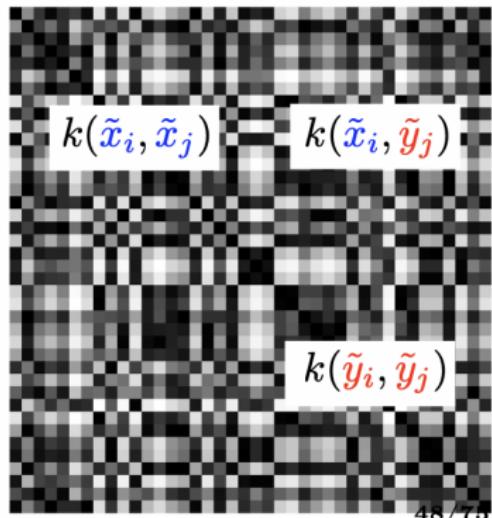
A statistical test using MMD

Permuted dog and fish samples (**merdogs**):

$$\tilde{X} = \begin{bmatrix} \text{fish} & \text{dog} & \text{fish} & \dots \end{bmatrix}$$

$$\tilde{Y} = \begin{bmatrix} \text{dog} & \text{fish} & \text{dog} & \dots \end{bmatrix}$$

$$\widehat{\text{MMD}}^2 = \frac{1}{n(n-1)} \sum_{i \neq j} k(\tilde{x}_i, \tilde{x}_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(\tilde{y}_i, \tilde{y}_j) - \frac{2}{n^2} \sum_{i,j} k(\tilde{x}_i, \tilde{y}_j)$$

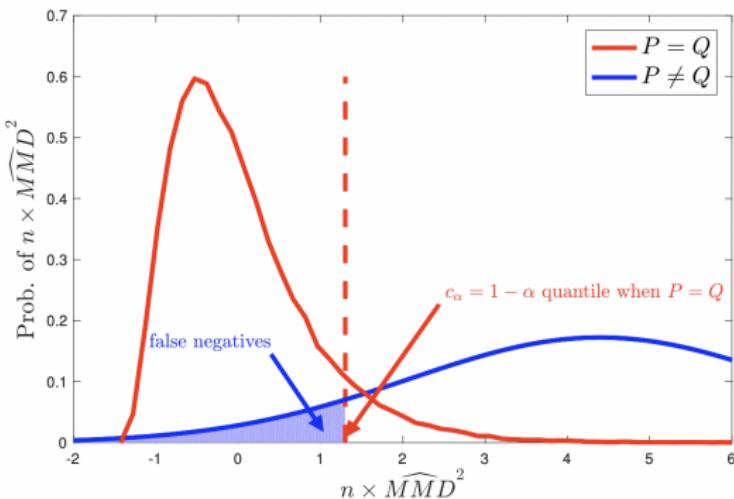


Permutation simulates

$$P = Q$$

For each permutation j set $T_j = n\text{MMD}^2(\tilde{\mathbb{P}}, \tilde{\mathbb{Q}})$

A statistical test using MMD



- Fix a significance level α (usually a small value: 0.05.)
- If $T_0 \geq c_\alpha$, reject the null, i.e. ($\mathbb{P} = \mathbb{Q}$ unlikely)
- Otherwise, cannot reject ($\mathbb{P} = \mathbb{Q}$ is likely).

How can we tell if $T_0 := n\widehat{MMD}^2(\mathbb{P}, \mathbb{Q}) \geq c_\alpha$?

- Let T be a r.v. under the null distribution: $T \sim 2 \sum_{i=1}^{\infty} \lambda_i(z_i^2 - 1)$.
- If the p -value $p := \mathbb{P}_T(T > T_0) \leq \alpha$, then $T_0 \geq c_\alpha$.
- For T_1, \dots, T_J samples from the null: $p \approx |\{j | T_j \geq T_0\}|/J$.

Can use a permutation test to construct T_1, \dots, T_J .

Outline

4 Characterizing probabilities with kernels

- Kernel mean embedding
- The Maximum Mean Discrepancy
 - Applications (I): Statistical testing using the MMD
 - Applications (II): Learning generative models
- Characteristic kernels

Given samples from a distribution \mathbb{P} over \mathcal{X} , want a model that can produce new samples from $\mathbb{Q} \approx \mathbb{P}$



$$X \sim \mathbb{P}$$



$$Y \sim \mathbb{Q}$$

Given samples from a distribution \mathbb{P} over \mathcal{X} , want a model that can produce new samples from $\mathbb{Q} \approx \mathbb{P}$



$$X \sim \mathbb{P}$$



$$Y \sim \mathbb{Q}$$

EGM: \mathbb{Q} has density $q(Y)$.

- **Support:** the whole space.
- **Training** using maximum likelihood or score matching.
- **Sampling** using MCMC.

Given samples from a distribution \mathbb{P} over \mathcal{X} , want a model that can produce new samples from $\mathbb{Q} \approx \mathbb{P}$



$$X \sim \mathbb{P}$$

EGM: \mathbb{Q} has density $q(Y)$.

- **Support:** the whole space.
- **Training** using maximum likelihood or score matching.
- **Sampling** using MCMC.



$$Y \sim \mathbb{Q}$$

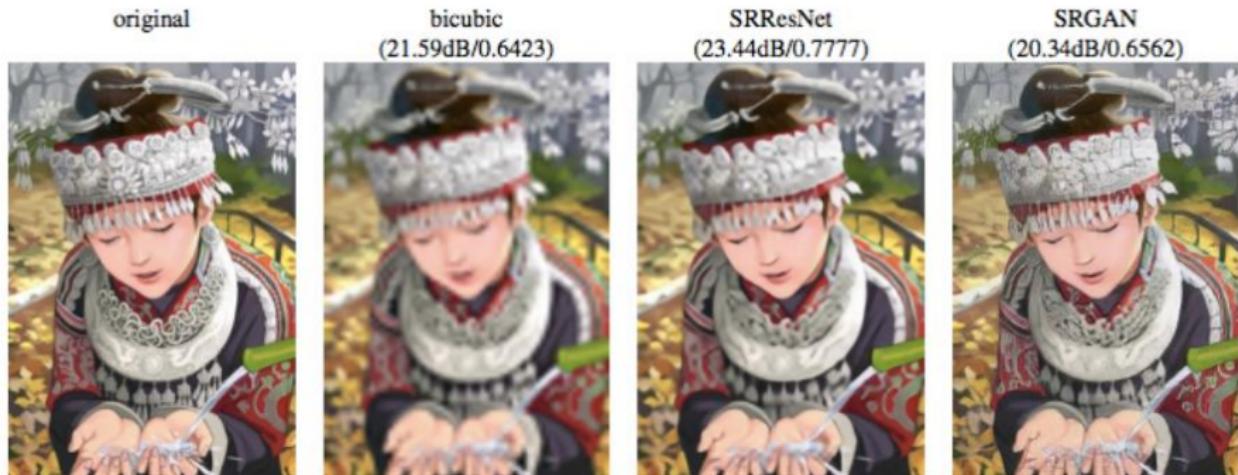
IGM: $Y = G(Z) \sim \mathbb{Q}$ with known $Z \sim \mu$.

- **Support:** low dimensional [Arjovsky 2017].
- **Training** by minimizing some well chosen divergence $D(\mathbb{P}, \mathbb{Q})$.
- **Sampling** by pushing μ forward with G .

Generative Adversarial Networks

Many successful applications:

- Single-image super-resolution

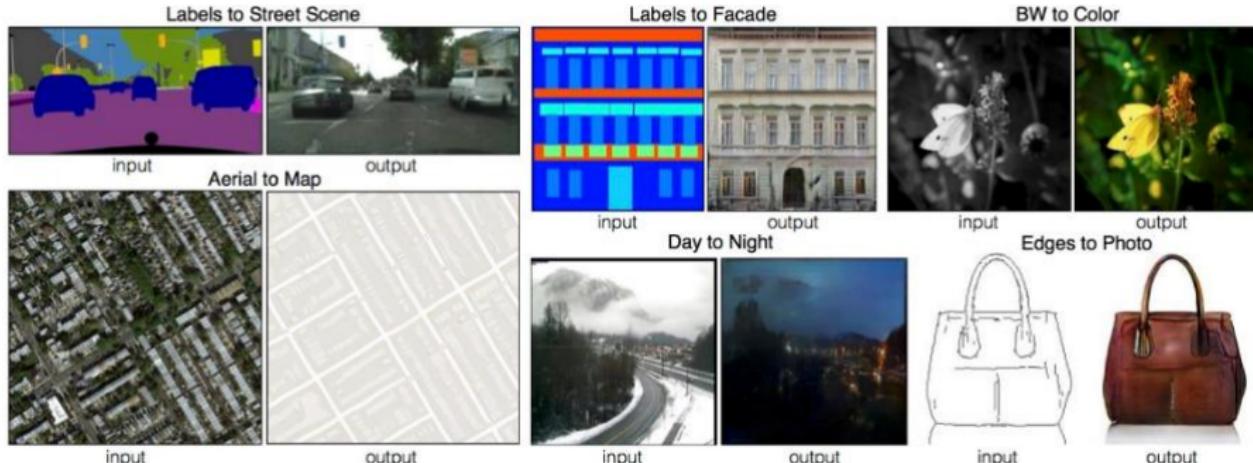


Ledig et al 2015

Generative Adversarial Networks

Many successful applications:

- Image to image translation



Isola et al 2016

Generative Adversarial Networks

Many successful applications:

- Text to image generation

This small blue bird has a short pointy beak and brown on its wings



This bird is completely red with black wings and pointy beak



Zhang et al 2016

Adversarial training [Goodfellow 2014]

Divergence $D(\mathbb{P}, \mathbb{Q})$ defined by maximizing a variational objective \mathcal{G} :

$$D(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}} \mathcal{G}(f, \mathbb{P}, \mathbb{Q})$$

- **Critic:** maximizes $\mathcal{G}(f, \mathbb{P}, \mathbb{Q})$ over $f \in \mathcal{F}$ to find optimal critic f^* .
- **Generator:** minimizes $D(\mathbb{P}, \mathbb{Q}) = \mathcal{G}(f^*, \mathbb{P}, \mathbb{Q})$ over \mathbb{Q} .
- Recover the MMD when \mathcal{F} is the unit ball in an RKHS \mathcal{H} .

Learning generative models using MMD

Goal is to solve the optimization problem:

$$\min_{\theta} MMD^2(\mathbb{P}, \mathbb{Q}_{\theta})$$

Learning generative models using MMD

Goal is to solve the optimization problem:

$$\min_{\theta} MMD^2(\mathbb{P}, \mathbb{Q}_{\theta})$$

- ① Sample a mini-batch of i.i.d samples $X_1, \dots, X_B \sim \mathbb{P}$ from data-set.
- ② Sample a mini-batch of i.i.d. latent noise $Z_1, \dots, Z_B \sim \mu$.
- ③ Generate IGM samples $Y_b = G_{\theta}(Z_b) \sim \mathbb{Q}_{\theta}$ for $1 \leq b \leq B$.
- ④ Compute empirical loss $\hat{\mathcal{L}}(\theta) := \widehat{MMD^2}(\mathbb{P}, \mathbb{Q}_{\theta})$. (Differentiable in θ)
- ⑤ Update parameters of the model using SGD:

$$\theta \leftarrow \theta - \gamma \nabla \hat{\mathcal{L}}(\theta).$$

Learning generative models using MMD

IGM trained using an RBF kernel on MNIST dataset.



Need better image features.

- In practice, choice of the kernel is crucial for good performance.
- Hard to design a kernel for high dimensional data like images.
- Why not learning it?

Learning generative models using MMD

Goal is to solve the optimization problem:

$$\min_{\theta} \sup_{k \in \mathcal{K}} MMD_k^2(\mathbb{P}, \mathbb{Q}_{\theta})$$

- \mathcal{K} is a family of kernels,
 - ex: parameterized by a neural network:

$$k(x, y) = h(\varphi(x), \varphi(y))$$

where φ is a NN and h is a fixed p.d. kernel.

- Adaptively select an MMD that best discriminates between \mathbb{P} and current model \mathbb{Q} .
- In practice, alternate between gradient steps on k and on θ : (Adversarial training).

Learning generative models using MMD

IGM trained on MNIST dataset.



Samples are better!

Learning generative models using MMD

IGM trained on CelebA dataset.



[A., Sutherland , Binkowski and Gretton, 2018]

Learning generative models using MMD

IGM trained on CelebA dataset.



[A., Sutherland , Binkowski and Gretton, 2018]

- More to the story: regularization, stability in optimization, evaluation, etc

Summary

- It is possible to represent probability distributions using kernels through the concept of **mean embeddings**.
- The **maximum mean discrepancy** (MMD), allows to compare probabilities by comparing their mean embeddings.
- MMD can be used for various applications:
 - Two sample tests
 - Learning implicit generative models (like GANs)
- Other applications include
 - Dependence detection
 - Feature selection
 - Bling source separaion (e.g. ICA)
- Often assume **good kernels** which do not discard information about distributions: **characteristic kernels**.

Outline

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
- 3 Kernels and Graphs
- 4 Characterizing probabilities with kernels
 - Kernel mean embedding
 - The Maximum Mean Discrepancy
 - Characteristic kernels
- 5 Open Problems and Research Topics

Can mean embeddings characterize probabilities?

Question: Given two probability distributions \mathbb{P} and \mathbb{Q} with mean embeddings $\mu_{\mathbb{P}}$ and $\mu_{\mathbb{Q}}$, can we **confidently** tell if \mathbb{P} and \mathbb{Q} are different or not based only on the summary given by $\mu_{\mathbb{P}}$ and $\mu_{\mathbb{Q}}$?

Can mean embeddings characterize probabilities?

Question: Given two probability distributions \mathbb{P} and \mathbb{Q} with mean embeddings $\mu_{\mathbb{P}}$ and $\mu_{\mathbb{Q}}$, can we **confidently** tell if \mathbb{P} and \mathbb{Q} are different or not based only on the summary given by $\mu_{\mathbb{P}}$ and $\mu_{\mathbb{Q}}$?

It depends on the kernel!

Characteristic kernels

Definition

Let \mathcal{X} be a topological set and \mathcal{P} the set of Borel probability measures on \mathcal{X} . Consider a **bounded measurable p.d.** kernel K defined on \mathcal{X} and let \mathcal{H} be its RKHS. The kernel K is said to be **characteristic** if the map $\mathcal{P} \ni \mathbb{P} \mapsto \mu_{\mathbb{P}} = \mathbb{E}_{X \sim \mathbb{P}}[K_X] \in \mathcal{H}$ is **injective**, i.e.:

$$\forall \mathbb{P}, \mathbb{Q} \in \mathcal{P} : \mu_{\mathbb{P}} = \mu_{\mathbb{Q}} \implies \mathbb{P} = \mathbb{Q}.$$

- Equality of mean embeddings \iff equality of expectations of functions in \mathcal{H} , i.e.:

$$\mu_{\mathbb{P}} = \mu_{\mathbb{Q}} \iff \mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)], \quad \forall f \in \mathcal{H}$$

Characteristic kernels

Definition

Let \mathcal{X} be a topological set and \mathcal{P} the set of Borel probability measures on \mathcal{X} . Consider a **bounded measurable p.d.** kernel K defined on \mathcal{X} and let \mathcal{H} be its RKHS. The kernel K is said to be **characteristic** if the map $\mathcal{P} \ni \mathbb{P} \mapsto \mu_{\mathbb{P}} = \mathbb{E}_{X \sim \mathbb{P}}[K_X] \in \mathcal{H}$ is **injective**, i.e.:

$$\forall \mathbb{P}, \mathbb{Q} \in \mathcal{P} : \mu_{\mathbb{P}} = \mu_{\mathbb{Q}} \implies \mathbb{P} = \mathbb{Q}.$$

- Equality of mean embeddings \iff equality of expectations of functions in \mathcal{H} , i.e.:

$$\mu_{\mathbb{P}} = \mu_{\mathbb{Q}} \iff \mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)], \quad \forall f \in \mathcal{H}$$

- Equality of probability distributions \iff Equality of expectations of continuous and bounded functions on \mathcal{X} , i.e.:

$$\mathbb{P} = \mathbb{Q} \iff \mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)], \quad \forall f \in \mathcal{C}_b(\mathcal{X}).$$

Characteristic kernels

Definition

Let \mathcal{X} be a topological set and \mathcal{P} the set of Borel probability measures on \mathcal{X} . Consider a **bounded measurable p.d.** kernel K defined on \mathcal{X} and let \mathcal{H} be its RKHS. The kernel K is said to be **characteristic** if the map $\mathcal{P} \ni \mathbb{P} \mapsto \mu_{\mathbb{P}} = \mathbb{E}_{X \sim \mathbb{P}}[K_X] \in \mathcal{H}$ is **injective**, i.e.:

$$\forall \mathbb{P}, \mathbb{Q} \in \mathcal{P} : \mu_{\mathbb{P}} = \mu_{\mathbb{Q}} \implies \mathbb{P} = \mathbb{Q}.$$

- Equality of mean embeddings \iff equality of expectations of functions in \mathcal{H} , i.e.:

$$\mu_{\mathbb{P}} = \mu_{\mathbb{Q}} \iff \mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)], \quad \forall f \in \mathcal{H}$$

- Equality of probability distributions \iff Equality of expectations of continuous and bounded functions on \mathcal{X} , i.e.:

$$\mathbb{P} = \mathbb{Q} \iff \mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)], \quad \forall f \in \mathcal{C}_b(\mathcal{X}).$$

- A kernel K is characteristic if RKHS \mathcal{H} is rich enough!

Characteristic kernels via Universality

Definition

Let K be a p.d. kernel with RKHS \mathcal{H} on a **compact** set \mathcal{X} . K is universal if $y \mapsto K(x, y)$ is continuous for all $x \in \mathcal{X}$ and \mathcal{H} is dense in $\mathcal{C}(\mathcal{X})$ in the maximum norm $\|\cdot\|_\infty$.

Characteristic kernels via Universality

Definition

Let K be a p.d. kernel with RKHS \mathcal{H} on a **compact** set \mathcal{X} . K is universal if $y \mapsto K(x, y)$ is continuous for all $x \in \mathcal{X}$ and \mathcal{H} is dense in $\mathcal{C}(\mathcal{X})$ in the maximum norm $\|.\|_\infty$.

Proposition

Assume \mathcal{X} is compact. If K is universal, then K is characteristic.

Characteristic kernels via Universality

Definition

Let K be a p.d. kernel with RKHS \mathcal{H} on a **compact** set \mathcal{X} . K is universal if $y \mapsto K(x, y)$ is continuous for all $x \in \mathcal{X}$ and \mathcal{H} is dense in $\mathcal{C}(\mathcal{X})$ in the maximum norm $\| \cdot \|_\infty$.

Proposition

Assume \mathcal{X} is compact. If K is universal, then K is characteristic.

proof: Let \mathbb{P} and \mathbb{Q} such that $\mu_{\mathbb{P}} = \mu_{\mathbb{Q}}$. We need to show that

$$\mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)], \forall f \in \mathcal{C}(\mathcal{X}).$$

Fix $f \in \mathcal{C}(\mathcal{X})$. By universality of K , \mathcal{H} is dense in $\mathcal{C}(\mathcal{X})$ in the sup norm. Hence, for any $\epsilon > 0$, there exists $\mathbf{g} \in \mathcal{H}$ such that $\|f - \mathbf{g}\|_\infty \leq \epsilon$.

Characteristic kernels via Universality

Proof Next we make the expansion

$$\begin{aligned} |\mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]| &\leq |\mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{X \sim \mathbb{P}}[g(X)]| \\ &\quad + |\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)] - \mathbb{E}_{Y \sim \mathbb{Q}}[g(Y)]| \\ &\quad + |\mathbb{E}_{X \sim \mathbb{P}}[g(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[g(Y)]|. \end{aligned}$$

The first two terms are upper-bounded by ϵ by definition of g . The last term is equal to 0 since $\mathbb{E}_{X \sim \mathbb{P}}[g(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[g(Y)] = \langle g, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}$ and $\mu_{\mathbb{P}} = \mu_{\mathbb{Q}}$ by assumption.

Hence, we have shown that for any $\epsilon > 0$:

$$|\mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]| \leq 2\epsilon$$

directly implying that $|\mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)]| = 0$.
The above holds for any $f \in \mathcal{C}(\mathcal{X})$, meaning that $\mathbb{P} = \mathbb{Q}$.

General criterion for Universality

Theorem: General criterion for universality (Steinwart, 2001)

Let \mathcal{X} be a compact metric space and k be a continuous kernel on \mathcal{X} with $k(x, x) > 0$. Suppose there is an injective map $\Phi(x) = \{\varphi_i(x)\}_{i \geq 0}$ such that $k(x, y) = \sum_{i=0}^{\infty} \varphi_i(x)\varphi_i(y)$. If the set $A := \text{span}\{\varphi_i | i \geq 0\}$ is an algebra, then k is universal.

General criterion for Universality

Theorem: General criterion for universality (Steinwart, 2001)

Let \mathcal{X} be a compact metric space and k be a continuous kernel on \mathcal{X} with $k(x, x) > 0$. Suppose there is an injective map $\Phi(x) = \{\varphi_i(x)\}_{i \geq 0}$ such that $k(x, y) = \sum_{i=0}^{\infty} \varphi_i(x)\varphi_i(y)$. If the set $A := \text{span}\{\varphi_i | i \geq 0\}$ is an algebra, then k is universal.

Definition (Algebra)

Let A be a vector space and $\times : A \times A \rightarrow A$ be a binary operation on A . Then A is an algebra if \times is bilinear, i.e. for all $x, y, z \in A$ and $a, b \in \mathbb{R}$:

$$z \times (x + y) = z \times x + z \times y$$

$$(x + y) \times z = x \times z + y \times z$$

$$(ax) \times (by) = (ab)(x \times y).$$

General criterion for Universality

Theorem: General criterion for universality (Steinwart, 2001)

Let \mathcal{X} be a compact metric space and k be a continuous kernel on \mathcal{X} with $k(x, x) > 0$. Suppose there is an injective map $\Phi(x) = \{\varphi_i(x)\}_{i \geq 0}$ such that $k(x, y) = \sum_{i=0}^{\infty} \varphi_i(x)\varphi_i(y)$. If the set $A := \text{span}\{\varphi_i | i \geq 0\}$ is an **algebra**, then k is universal.

Main ingredient: Stone-Weierstrass's theorem

Let (\mathcal{X}, d) be a compact metric space and A a linear subspace of $\mathcal{C}(\mathcal{X})$. Then A is dense in $\mathcal{C}(\mathcal{X})$ if

- **A is an algebra** for the product of functions.
- **A does not vanish:** For all $x \in \mathcal{X}$, there exists $f \in A$ s.t. $f(x) \neq 0$.
- **A separates points:** For all $x, y \in \mathcal{X}$ with $x \neq y$, there exists $f \in A$, s.t. $f(x) \neq f(y)$.

General criterion for Universality

Theorem: General criterion for universality (Steinwart, 2001)

Let \mathcal{X} be a compact metric space and k be a continuous kernel on \mathcal{X} with $k(x, x) > 0$. Suppose there is an injective map $\Phi(x) = \{\varphi_i(x)\}_{i \geq 0}$ such that $k(x, y) = \sum_{i=0}^{\infty} \varphi_i(x)\varphi_i(y)$. If the set $A := \text{span}\{\varphi_i | i \geq 0\}$ is an **algebra**, then k is universal.

Proof:

- **A is a subset of $\mathcal{C}(\mathcal{X})$.** Follows by continuity of the map $x \mapsto \Phi(x)$.
Indeed, $\|\Phi(x) - \Phi(y)\|^2 = K(x, x) + K(y, y) - 2K(x, y) \leq \epsilon$ for any $\epsilon > 0$ provided that y is close enough to x since K is continuous.
- **A does not vanish.** Otherwise, we can find x such that $\varphi_i(x) = 0$ for all $i \geq 0$, meaning that $K(x, x) = 0$: contradicts $K(x, x) > 0$.
- **A separates points.** Otherwise, there exists x, y with $x \neq y$ and $\varphi_i(x) = \varphi_i(y)$ for all $i \geq 0$, hence $\Phi(x) = \Phi(y)$: contradicts Φ injective.

Hence A is dense in $\mathcal{C}(\mathcal{X})$ by Stone-Weierstrass theorem.

General criterion for Universality

Theorem: General criterion for universality (Steinwart, 2001)

Let \mathcal{X} be a compact metric space and k be a continuous kernel on \mathcal{X} with $k(x, x) > 0$. Suppose there is an injective map $\Phi(x) = \{\varphi_i(x)\}_{i \geq 0}$ such that $k(x, y) = \sum_{i=0}^{\infty} \varphi_i(x)\varphi_i(y)$. If the set $A := \text{span}\{\varphi_i | i \geq 0\}$ is an algebra, then k is universal.

Proof Continued: Let $f \in \mathcal{C}(\mathcal{X})$ and $\epsilon > 0$.

- Since A is dense in $\mathcal{C}(\mathcal{X})$, there exists $g \in A$ s.t. $\|f - g\|_{\infty} < \epsilon$.
- By definition of A , the function g is of the form $g(x) = \langle w, \Phi(x) \rangle_{\mathcal{H}}$ with $w = (w_i)_{i \geq 0}$ s.t. $w_i = 0$ for any $i > N$ for some $N < \infty$.
- Hence, g belongs to the unique RKHS \mathcal{H} of K . This shows that \mathcal{H} is dense in $\mathcal{C}(\mathcal{X})$, hence K is universal.

Criteria for Universality

Proposition (Steinwart 2001)

Let $0 < r \leq \infty$ and $f : (-r, r) \rightarrow \mathbb{R}$ be a C^∞ function that admits an expansion as a Taylor series in 0: $f(x) = \sum_{i=0}^{\infty} a_i x^i$. Let \mathcal{X} be a compact set in the open centered ball in \mathbb{R}^d of radius \sqrt{r} . If $a_i > 0$ for all $i \geq 0$, then $k(x, y) = f(\langle x, y \rangle)$ defines a universal kernel on \mathcal{X} .

Criteria for Universality

Proposition (Steinwart 2001)

Let $0 < r \leq \infty$ and $f : (-r, r) \rightarrow \mathbb{R}$ be a C^∞ function that admits an expansion as a Taylor series in 0: $f(x) = \sum_{i=0}^{\infty} a_i x^i$. Let \mathcal{X} be a compact set in the open centered ball in \mathbb{R}^d of radius \sqrt{r} . If $a_i > 0$ for all $i \geq 0$, then $k(x, y) = f(\langle x, y \rangle)$ defines a universal kernel on \mathcal{X} .

Example 1: Exp kernel: $K(x, y) = \exp \langle x, y \rangle$ on any compact \mathcal{X} .

$$f(x) = \exp(x) = \sum_{i=0}^{\infty} \frac{1}{i!} x^i, \quad K(x, y) = f(\langle x, y \rangle).$$

Criteria for Universality

Proposition (Steinwart 2001)

Let $0 < r \leq \infty$ and $f : (-r, r) \rightarrow \mathbb{R}$ be a C^∞ function that admits an expansion as a Taylor series in 0: $f(x) = \sum_{i=0}^{\infty} a_i x^i$. Let \mathcal{X} be a compact set in the open centered ball in \mathbb{R}^d of radius \sqrt{r} . If $a_i > 0$ for all $i \geq 0$, then $k(x, y) = f(\langle x, y \rangle)$ defines a universal kernel on \mathcal{X} .

Example 1: Exp kernel: $K(x, y) = \exp \langle x, y \rangle$ on any compact \mathcal{X} .

$$f(x) = \exp(x) = \sum_{i=0}^{\infty} \frac{1}{i!} x^i, \quad K(x, y) = f(\langle x, y \rangle).$$

Example 2: Gaussian kernel on the Unit Sphere

$$K(x, y) = \exp(-\frac{1}{2} \|x - y\|^2).$$

$$f(x) = e^{-1} \exp(x) = e^{-1} \sum_{i=0}^{\infty} \frac{1}{i!} x^i, \quad K(x, y) = f(\langle x, y \rangle).$$

Criteria for Universality

Proposition

Let $0 < r \leq \infty$ and $f : (-r, r) \rightarrow \mathbb{R}$ be a C^∞ function that admits an expansion as a Taylor series in 0: $f(x) = \sum_{i=0}^{\infty} a_i x^i$. Let \mathcal{X} be a compact set in the open centered ball in \mathbb{R}^d of radius \sqrt{r} . If $a_i > 0$ for all $i \geq 0$, then $k(x, y) = f(\langle x, y \rangle)$ defines a universal kernel on \mathcal{X} .

Proof: For simplicity, take $d = 1$.

- K is continuous and of the form:

$$K(x, y) := \sum_{i=0}^{\infty} a_i x^i y^i = \langle \Phi(x), \Phi(y) \rangle_{L_2}$$

with $\Phi(x) = (\sqrt{a_i} x^i)_{i \geq 0}$ which is injective.

- $K(x, x) = \sum_{i=0}^{\infty} a_i x^{2i} > 0$ since $a_i > 0$ for all $i \geq 0$.
- $A := \text{span}(\{\varphi_n | n \geq 0\})$ is the algebra of polynomials.
- Hence K universal by the general criterion for universality.

Criteria for Universality

Proposition (Steinwart 2001)

Let $f : [0, 2\pi] \rightarrow \mathbb{R}$ be a continuous function that can be expanded in a pointwise absolutely convergent Fourier series: $f(t) = \sum_{n=0}^{\infty} a_n \cos(nt)$. If $a_n > 0$ for all $n \geq 0$, then the Kernel $K(x, y) := \prod_{i=1}^d f(|x_i - y_i|)$ defines a universal kernel on every compact subset of $[0, 2\pi]^d$.

Criteria for Universality

Proposition (Steinwart 2001)

Let $f : [0, 2\pi] \rightarrow \mathbb{R}$ be a continuous function that can be expanded in a pointwise absolutely convergent Fourier series: $f(t) = \sum_{n=0}^{\infty} a_n \cos(nt)$. If $a_n > 0$ for all $n \geq 0$, then the Kernel $K(x, y) := \prod_{i=1}^d f(|x_i - y_i|)$ defines a universal kernel on every compact subset of $[0, 2\pi]^d$.

Example 1: The stronger regularized Fourier kernel (Vapnik 1998, p.470)

$$k(x, y) = (1 - q^2) / (2 - 4q \cos(x - y) + 2q^2)$$

for any $0 < q < 1$.

Criteria for Universality

Proposition (Steinwart 2001)

Let $f : [0, 2\pi] \rightarrow \mathbb{R}$ be a continuous function that can be expanded in a pointwise absolutely convergent Fourier series: $f(t) = \sum_{n=0}^{\infty} a_n \cos(nt)$. If $a_n > 0$ for all $n \geq 0$, then the Kernel $K(x, y) := \prod_{i=1}^d f(|x_i - y_i|)$ defines a universal kernel on every compact subset of $[0, 2\pi]^d$.

Proof: For simplicity, take $d=1$.

- K is continuous and of the form:

$$K(x, y) = a_0 + \sum_{n=0}^{\infty} a_n (\sin(nx)\sin(ny) + \cos(nx)\cos(ny)) = \langle \Phi(x), \Phi(y) \rangle_{l_2}$$

where $\Phi(x) = (\varphi_n(x))_{n \geq 0}$ defined by $\varphi_0(x) = a_0$, $\varphi_{2n-1}(x) = \sqrt{a_n} \sin(nx)$ and $\varphi_{2n}(x) = \sqrt{a_n} \cos(nx)$ for $n \geq 1$ is injective.

- $K(x, x) = \sum_{n=0}^{\infty} a_n > 0$ since $a_n > 0$ for all $n \geq 0$.
- $A := \text{span}(\{\varphi_n | n \geq 0\})$ is an algebra (by trigonometric identities).
- Hence K universal by the general criterion for universality.

Summary: Characteristic kernels via Universality

- On a compact metric set \mathcal{X} , a universal kernel is a continuous kernel whose RKHS (H) is dense in $\mathcal{C}(\mathcal{X})$ in the maximum norm.
- Any universal kernel on \mathcal{X} is characteristic, i.e. the mean embedding map $\mathbb{P} \mapsto \mu_{\mathbb{P}} = \mathbb{E}_{x \sim \mathbb{P}}[K_x] \in \mathcal{H}$ defined on the set \mathcal{P} of probability distributions on \mathcal{X} is injective:

$$\forall \mathbb{P}, \mathbb{Q} \in \mathcal{P} : \mu_{\mathbb{P}} = \mu_{\mathbb{Q}} \implies \mathbb{P} = \mathbb{Q}.$$

- Can construct a large class of universal kernels using Taylor series or Fourier series with positive coefficients.
- Both constructions follow from the General criterion for universality, itself a consequence of Stone-Weierstrass theorem for compact metric sets.
- Question: What if \mathcal{X} is not compact?

Translation invariant kernels on \mathbb{R}^d

Definition

A kernel $K : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ is called **translation invariant** (t.i.), or **shift-invariant**, if it only depends on the difference between its argument, i.e.:

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \quad K(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x} - \mathbf{y})$$

for some function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$. Such a function φ is called positive definite if the corresponding kernel K is p.d.

Translation invariant kernels on \mathbb{R}^d

Definition

A kernel $K : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ is called **translation invariant** (t.i.), or **shift-invariant**, if it only depends on the difference between its argument, i.e.:

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \quad K(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x} - \mathbf{y})$$

for some function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$. Such a function φ is called positive definite if the corresponding kernel K is p.d.

Theorem (Bochner)

A **continuous** function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$ is p.d. if and only if it is the Fourier-Stieltjes transform of a symmetric and positive finite Borel measure $\mu \in M(\mathbb{R}^d)$

Characteristic kernels via Fourier transform

Translation invariant characteristic kernels: (Sriperumbudur 2008)

Let K be a translation invariant kernel on \mathbb{R}^d of the form

$K(x, y) = \kappa(x - y)$ with $\kappa(z) = \int e^{-iz^\top w} d\Lambda(w)$ for some finite non-negative Borel measure Λ on \mathbb{R}^d . The kernel K is characteristic if and only if $\text{supp}(\Lambda) = \mathbb{R}^d$.

Characteristic kernels via Fourier transform

Translation invariant characteristic kernels: (Sriperumbudur 2008)

Let K be a translation invariant kernel on \mathbb{R}^d of the form

$K(x, y) = \kappa(x - y)$ with $\kappa(z) = \int e^{-iz^\top w} d\Lambda(w)$ for some finite non-negative Borel measure Λ on \mathbb{R}^d . The kernel K is characteristic if and only if $\text{supp}(\Lambda) = \mathbb{R}^d$.

Example 1: Gaussian kernel $K(x, y) = e^{-\frac{\sigma^2}{2}\|x-y\|^2}$. The measure Λ is a gaussian on \mathbb{R}^d with density $w \mapsto (1/\sqrt{2\pi\sigma^2})^d e^{-\frac{1}{2\sigma^2}\|w\|^2}$. Since $\text{supp}(\Lambda) = \mathbb{R}^d$, K is characteristic.

Characteristic kernels via Fourier transform

Translation invariant characteristic kernels: (Sriperumbudur 2008)

Let K be a translation invariant kernel on \mathbb{R}^d of the form

$K(x, y) = \kappa(x - y)$ with $\kappa(z) = \int e^{-iz^\top w} d\Lambda(w)$ for some finite non-negative Borel measure Λ on \mathbb{R}^d . The kernel K is characteristic if and only if $\text{supp}(\Lambda) = \mathbb{R}^d$.

Example 1: Gaussian kernel $K(x, y) = e^{-\frac{\sigma^2}{2}\|x-y\|^2}$. The measure Λ is a gaussian on \mathbb{R}^d with density $w \mapsto (1/\sqrt{2\pi\sigma^2})^d e^{-\frac{1}{2\sigma^2}\|w\|^2}$. Since $\text{supp}(\Lambda) = \mathbb{R}^d$, K is characteristic.

Example 2: Let $\kappa(z) = z^{-1} \sin(z)$. Then $K(x, y) = \kappa(x - y)$ is not characteristic: Λ is the uniform distribution on the $[-1, 1]$.

Proof sketch

- By bochner's theorem:

$$K(x, y) = \int \underbrace{e^{-ix^\top \omega}}_{\Phi(x)(w)} e^{iy^\top \omega} d\Lambda(\omega) = \langle \Phi(x), \Phi(y) \rangle_{L_2(\Lambda)}.$$

Proof sketch

- By bochner's theorem:

$$K(x, y) = \int \underbrace{e^{-ix^\top \omega}}_{\Phi(x)(w)} e^{iy^\top \omega} d\Lambda(\omega) = \langle \Phi(x), \Phi(y) \rangle_{L_2(\Lambda)}.$$

- Can express the mean embedding $\mu_{\mathbb{P}}$ in terms of $\mathcal{F}(\mathbb{P}) = \mathbb{E}_{X \sim \mathbb{P}}[\Phi(X)]$ the of Fourier transform of \mathbb{P} :

$$\mu_{\mathbb{P}}(y) = \mathbb{E}_{X \sim \mathbb{P}}[\langle \Phi(X), \Phi(y) \rangle_{L_2(\Lambda)}] = \langle \mathcal{F}(\mathbb{P}), \Phi(y) \rangle_{L_2(\Lambda)}$$

Proof sketch

- By bochner's theorem:

$$K(x, y) = \int \underbrace{e^{-ix^\top \omega}}_{\Phi(x)(w)} e^{iy^\top \omega} d\Lambda(\omega) = \langle \Phi(x), \Phi(y) \rangle_{L_2(\Lambda)}.$$

- Can express the mean embedding $\mu_{\mathbb{P}}$ in terms of $\mathcal{F}(\mathbb{P}) = \mathbb{E}_{X \sim \mathbb{P}}[\Phi(X)]$ the of Fourier transform of \mathbb{P} :

$$\mu_{\mathbb{P}}(y) = \mathbb{E}_{X \sim \mathbb{P}}[\langle \Phi(X), \Phi(y) \rangle_{L_2(\Lambda)}] = \langle \mathcal{F}(\mathbb{P}), \Phi(y) \rangle_{L_2(\Lambda)}$$

- Equality in mean embeddings ($\mu_{\mathbb{P}} = \mu_{\mathbb{Q}}$) iff $\mathcal{F}(\mathbb{P})(\omega) = \mathcal{F}(\mathbb{Q})(\omega)$ Λ -almost surely.

Proof sketch

- By Bochner's theorem:

$$K(x, y) = \int \underbrace{e^{-ix^\top \omega}}_{\Phi(x)(w)} e^{iy^\top \omega} d\Lambda(\omega) = \langle \Phi(x), \Phi(y) \rangle_{L_2(\Lambda)}.$$

- Can express the mean embedding $\mu_{\mathbb{P}}$ in terms of $\mathcal{F}(\mathbb{P}) = \mathbb{E}_{X \sim \mathbb{P}}[\Phi(X)]$ the Fourier transform of \mathbb{P} :

$$\mu_{\mathbb{P}}(y) = \mathbb{E}_{X \sim \mathbb{P}}[\langle \Phi(X), \Phi(y) \rangle_{L_2(\Lambda)}] = \langle \mathcal{F}(\mathbb{P}), \Phi(y) \rangle_{L_2(\Lambda)}$$

- Equality in mean embeddings ($\mu_{\mathbb{P}} = \mu_{\mathbb{Q}}$) iff $\mathcal{F}(\mathbb{P})(\omega) = \mathcal{F}(\mathbb{Q})(\omega)$ Λ -almost surely.

Fourier inversion theorem (Dudley 2002, Theorem 9.5.4)

If \mathbb{P} and \mathbb{Q} are two probability distributions on \mathbb{R}^d with the same Fourier transform: $\mathcal{F}(\mathbb{P}) = \mathcal{F}(\mathbb{Q})$, then $\mathbb{P} = \mathbb{Q}$.

Proof sketch

- By bochner's theorem:

$$K(x, y) = \int e^{\overbrace{-ix^\top \omega}^{\Phi(x)(\omega)}} e^{iy^\top \omega} d\Lambda(\omega) = \langle \Phi(x), \Phi(y) \rangle_{L_2(\Lambda)}.$$

- Can express the mean embedding $\mu_{\mathbb{P}}$ in terms of $\mathcal{F}(\mathbb{P}) = \mathbb{E}_{X \sim \mathbb{P}}[\Phi(X)]$ the of Fourier transform of \mathbb{P} :

$$\mu_{\mathbb{P}}(y) = \mathbb{E}_{X \sim \mathbb{P}}[\langle \Phi(X), \Phi(y) \rangle_{L_2(\Lambda)}] = \langle \mathcal{F}(\mathbb{P}), \Phi(y) \rangle_{L_2(\Lambda)}$$

- Equality in mean embeddings ($\mu_{\mathbb{P}} = \mu_{\mathbb{Q}}$) iff $\mathcal{F}(\mathbb{P})(\omega) = \mathcal{F}(\mathbb{Q})(\omega)$ Λ -almost surely.

Fourier inversion theorem (Dudley 2002, Theorem 9.5.4)

If \mathbb{P} and \mathbb{Q} are two probability distributions on \mathbb{R}^d with the same Fourier transform: $\mathcal{F}(\mathbb{P}) = \mathcal{F}(\mathbb{Q})$, then $\mathbb{P} = \mathbb{Q}$.

The measure Λ must be supported on the whole space.

Characteristic kernels: Summary

Definition

Let \mathcal{X} be a topological set and \mathcal{P} the set of Borel probability measures on \mathcal{X} . Consider a **bounded measurable p.d.** kernel K defined on \mathcal{X} and let \mathcal{H} be its RKHS. The kernel K is said to be **characteristic** if the map $\mathcal{P} \ni \mathbb{P} \mapsto \mu_{\mathbb{P}} = \mathbb{E}_{x \sim \mathbb{P}}[K_x] \in \mathcal{H}$ is **injective**, i.e.:

$$\forall \mathbb{P}, \mathbb{Q} \in \mathcal{P} : \mu_{\mathbb{P}} = \mu_{\mathbb{Q}} \implies \mathbb{P} = \mathbb{Q}.$$

Criteria for characteristic kernels

- On a compact set \mathcal{X} , can use criteria for universality: A kernel is universal if it continuous and its RKHS is dense in $C(\mathcal{X})$.
 - If K admits a Taylor expansion with positive coefficients.
 - If K admits a Fourier expansion with positive coefficients.
- If $\mathcal{X} = \mathbb{R}^d$ and K is translation invariant with associated non-negative measure Λ : **K characteristic $\iff \text{supp}(\Lambda) = \mathbb{R}^d$**

Open Problems and Research Topics

Outline

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
- 3 Kernels and Graphs
- 4 Characterizing probabilities with kernels
- 5 Open Problems and Research Topics
 - Large-scale learning with kernels
 - Foundations of deep learning from a kernel point of view

Outline

5 Open Problems and Research Topics

- Large-scale learning with kernels
 - Motivation
 - Nyström approximations
 - Random Fourier features
- Foundations of deep learning from a kernel point of view

Motivation

Main problem

All methods we have seen require computing the $n \times n$ Gram matrix, which is infeasible when n is significantly greater than 100 000 both in terms of memory and computation.

Solutions

- low-rank approximation of the kernel;
- random Fourier features.

The goal is to find an approximate embedding $\psi : \mathcal{X} \rightarrow \mathbb{R}^d$ such that

$$K(\mathbf{x}, \mathbf{x}') \approx \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle_{\mathbb{R}^d}.$$

and use large-scale optimization techniques dedicated to linear models!

Motivation

Then, functions f in \mathcal{H} may be approximated by linear ones in \mathbb{R}^d , e.g.,.

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i K(\mathbf{x}_i, \mathbf{x}) \approx \left\langle \sum_{i=1}^n \alpha_i \psi(\mathbf{x}_i), \psi(\mathbf{x}) \right\rangle_{\mathbb{R}^d} = \langle \mathbf{w}, \psi(\mathbf{x}) \rangle_{\mathbb{R}^d}.$$

Then, the ERM problem

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

becomes, approximately,

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n L(y_i, \mathbf{w}^\top \psi(\mathbf{x}_i)) + \lambda \|\mathbf{w}\|_2^2,$$

which we know how to solve when n is large.

Outline

5 Open Problems and Research Topics

- Large-scale learning with kernels
 - Motivation
 - Nyström approximations
 - Random Fourier features
- Foundations of deep learning from a kernel point of view

Nyström approximations: principle

Consider a p.d. kernel $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and RKHS \mathcal{H} , with the mapping $\varphi : \mathcal{X} \rightarrow \mathcal{H}$ such that

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

The Nyström method consists of replacing any point $\varphi(\mathbf{x})$ in \mathcal{H} , for \mathbf{x} in \mathcal{X} by its orthogonal projection onto a **finite-dimensional subspace**

$$\mathcal{F} := \text{Span}(f_1, \dots, f_p) \quad \text{with } p \ll n,$$

where the f_i 's are **anchor points** in \mathcal{H} (to be defined later).

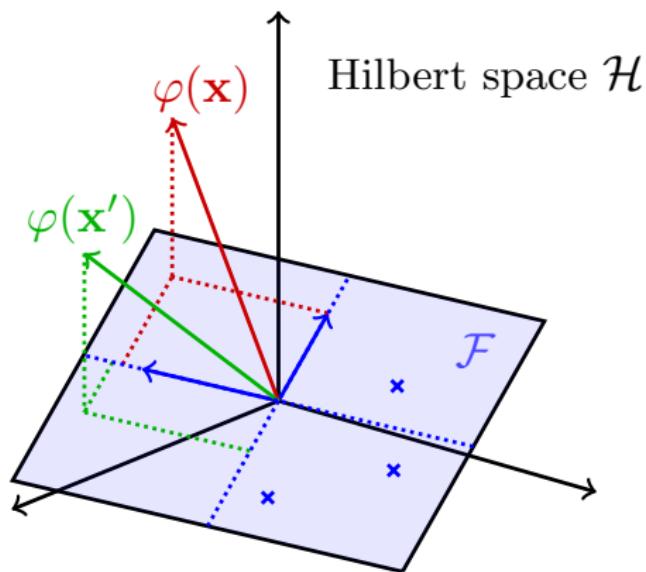
Motivation

- This principle allows us to work explicitly in a **finite-dimensional space**; it was introduced several times in the kernel literature [Williams and Seeger, 2002], [Smola and Schölkopf, 2000], [Fine and Scheinberg, 2001].

Nyström approximations: principle

The orthogonal projection is defined as

$$\Pi_{\mathcal{F}}[\mathbf{x}] := \operatorname{argmin}_{f \in \mathcal{F}} \|\varphi(\mathbf{x}) - f\|_{\mathcal{H}}^2,$$



Nyström approximations: principle

The projection is equivalent to

$$\Pi_{\mathcal{F}}[\mathbf{x}] := \sum_{j=1}^p \beta_j^* f_j \quad \text{with} \quad \beta^* \in \operatorname{argmin}_{\beta \in \mathbb{R}^p} \left\| \varphi(\mathbf{x}) - \sum_{j=1}^p \beta_j f_j \right\|_{\mathcal{H}}^2,$$

and β^* is the solution of the problem

$$\min_{\beta \in \mathbb{R}^p} -2 \sum_{j=1}^p \beta_j \langle f_j, \varphi(\mathbf{x}) \rangle_{\mathcal{H}} + \sum_{j,l=1}^p \beta_j \beta_l \langle f_j, f_l \rangle_{\mathcal{H}},$$

or also

$$\min_{\beta \in \mathbb{R}^p} -2 \sum_{j=1}^p \beta_j f_j(\mathbf{x}) + \sum_{j,l=1}^p \beta_j \beta_l \langle f_j, f_l \rangle_{\mathcal{H}}.$$

Nyström approximations: principle

Then, call $[\mathbf{K}_f]_{jl} = \langle f_j, f_l \rangle_{\mathcal{H}}$ and $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_p(\mathbf{x})]$ in \mathbb{R}^p . The problem may be rewritten as

$$\min_{\beta \in \mathbb{R}^p} -2\beta^\top \mathbf{f}(\mathbf{x}) + \beta^\top \mathbf{K}_f \beta,$$

and, assuming \mathbf{K}_f to be non-singular for simplicity, the solution is $\beta^*(\mathbf{x}) = \mathbf{K}_f^{-1} \mathbf{f}(\mathbf{x})$. Then,

$$\varphi(\mathbf{x}) \approx \sum_{j=1}^p \beta_j^*(\mathbf{x}) f_j,$$

and

$$\begin{aligned} \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{H}} &\approx \left\langle \sum_{j=1}^p \beta_j^*(\mathbf{x}) f_j, \sum_{j=1}^p \beta_j^*(\mathbf{x}') f_j \right\rangle_{\mathcal{H}} \\ &= \sum_{j,l=1}^p \beta_j^*(\mathbf{x}) \beta_l^*(\mathbf{x}') \langle f_j, f_l \rangle_{\mathcal{H}} = \beta^*(\mathbf{x})^\top \mathbf{K}_f \beta^*(\mathbf{x}'). \end{aligned}$$

Nyström approximations: principle

This allows us to define the mapping

$$\psi(\mathbf{x}) = \mathbf{K}_f^{1/2} \beta^*(\mathbf{x}) = \mathbf{K}_f^{-1/2} \mathbf{f}(\mathbf{x}),$$

and we have the approximation $K(\mathbf{x}, \mathbf{x}') \approx \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle_{\mathbb{R}^p}$.

Remarks

- the mapping provides low-rank approximations of the kernel matrix.
Given an $n \times n$ Gram matrix \mathbf{K} computed on a training set
 $\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, we have

$$\mathbf{K} \approx \psi(\mathcal{S})^\top \psi(\mathcal{S}),$$

where $\psi(\mathcal{S}) := [\psi(\mathbf{x}_1), \dots, \psi(\mathbf{x}_n)]$.

- the approximation has a **geometric interpretation**.
- We need to **define a good strategy** for choosing the f_j 's.

Nyström approximation via kernel PCA

Let us now try to **learn** the f_j 's given training data $\mathbf{x}_1, \dots, \mathbf{x}_n$ in \mathcal{X} :

$$\min_{\substack{f_1, \dots, f_p \in \mathcal{H} \\ \beta_{ij} \in \mathbb{R}}} \sum_{i=1}^n \left\| \varphi(\mathbf{x}_i) - \sum_{j=1}^p \beta_{ij} f_j \right\|_{\mathcal{H}}^2.$$

Using similar calculation as before, the objective is equivalent to

$$\min_{\substack{f_1, \dots, f_p \in \mathcal{H} \\ \beta_i \in \mathbb{R}^p}} \sum_{i=1}^n -2\beta_i^\top \mathbf{f}(\mathbf{x}_i) + \beta_i^\top \mathbf{K}_f \beta_i,$$

and, by minimizing with respect to all β_i with \mathbf{f} fixed, we have that $\beta_i = \mathbf{K}_f^{-1} \mathbf{f}(\mathbf{x}_i)$ (assuming \mathbf{K}_f to be invertible), which leads to

$$\max_{f_1, \dots, f_p \in \mathcal{H}} \sum_{i=1}^n \mathbf{f}(\mathbf{x}_i)^\top \mathbf{K}_f^{-1} \mathbf{f}(\mathbf{x}_i).$$

Nyström approximation via kernel PCA

Remember the objective:

$$\max_{f_1, \dots, f_p \in \mathcal{H}} \sum_{i=1}^n \mathbf{f}(\mathbf{x}_i)^\top \mathbf{K}_{\mathbf{f}}^{-1} \mathbf{f}(\mathbf{x}_i).$$

Consider an optimal solution \mathbf{f}^* and compute the eigenvalue decomposition of $\mathbf{K}_{\mathbf{f}^*} = \mathbf{U}\Delta\mathbf{U}^\top$. Then, define the functions

$$\mathbf{g}^*(\mathbf{x}) := [g_1^*(\mathbf{x}), \dots, g_p^*(\mathbf{x})] = \Delta^{-1/2} \mathbf{U}^\top \mathbf{f}^*(\mathbf{x}).$$

The functions g_j^* are points in the RKHS \mathcal{H} since they are linear combinations of the functions f_j^* in \mathcal{H} .

Nyström approximation via kernel PCA

Remember the objective:

$$\max_{f_1, \dots, f_p \in \mathcal{H}} \sum_{i=1}^n \mathbf{f}(\mathbf{x}_i)^\top \mathbf{K}_{\mathbf{f}}^{-1} \mathbf{f}(\mathbf{x}_i).$$

Consider an optimal solution \mathbf{f}^* and compute the eigenvalue decomposition of $\mathbf{K}_{\mathbf{f}^*} = \mathbf{U}\Delta\mathbf{U}^\top$. Then, define the functions

$$\mathbf{g}^*(\mathbf{x}) := [g_1^*(\mathbf{x}), \dots, g_p^*(\mathbf{x})] = \Delta^{-1/2} \mathbf{U}^\top \mathbf{f}^*(\mathbf{x}).$$

The functions g_j^* are points in the RKHS \mathcal{H} since they are linear combinations of the functions f_j^* in \mathcal{H} .

Exercise: check that all we do here and in the next slides can be extended to deal with singular Gram matrices $\mathbf{K}_{\mathbf{f}^}$ and $\mathbf{K}_{\mathbf{f}}$.*

Nyström approximation via kernel PCA

Besides, by construction

$$\begin{aligned} [\mathbf{K}_{\mathbf{g}^*}]_{jl} &:= \langle g_j^*, g_l^* \rangle_{\mathcal{H}} \\ &= \left\langle \frac{1}{\sqrt{\Delta_{jj}}} \sum_{k=1}^p [\mathbf{U}]_{kj} f_k^*, \frac{1}{\sqrt{\Delta_{ll}}} \sum_{k=1}^p [\mathbf{U}]_{kl} f_k^* \right\rangle_{\mathcal{H}} \\ &= \frac{1}{\sqrt{\Delta_{jj}}} \frac{1}{\sqrt{\Delta_{ll}}} \sum_{k,k'=1}^p [\mathbf{U}]_{kj} [\mathbf{U}]_{k'l} \langle f_k^*, f_{k'}^* \rangle_{\mathcal{H}} \\ &= \frac{1}{\sqrt{\Delta_{jj}}} \frac{1}{\sqrt{\Delta_{ll}}} \sum_{k,k'=1}^p [\mathbf{U}]_{kj} [\mathbf{U}]_{k'l} [\mathbf{K}_{f^*}]_{kk'} \\ &= \frac{1}{\sqrt{\Delta_{jj}}} \frac{1}{\sqrt{\Delta_{ll}}} \mathbf{u}_j^\top \mathbf{K}_{f^*} \mathbf{u}_l \\ &= \delta_{j=l}. \end{aligned}$$

Nyström approximation via kernel PCA

Then, $\mathbf{K}_{\mathbf{g}^*} = \mathbf{I}$ and \mathbf{g}^* is also a solution of the problem

$$\max_{f_1, \dots, f_p \in \mathcal{H}} \sum_{i=1}^n \mathbf{f}(\mathbf{x}_i)^\top \mathbf{K}_{\mathbf{f}}^{-1} \mathbf{f}(\mathbf{x}_i),$$

since

$$\begin{aligned}\mathbf{f}^*(\mathbf{x}_i)^\top \mathbf{K}_{\mathbf{f}^*}^{-1} \mathbf{f}^*(\mathbf{x}_i) &= \mathbf{f}^*(\mathbf{x}_i)^\top \mathbf{U} \mathbf{\Delta}^{-1} \mathbf{U}^\top \mathbf{f}^*(\mathbf{x}_i) \\ &= \mathbf{g}^*(\mathbf{x}_i)^\top \mathbf{g}^*(\mathbf{x}_i) = \mathbf{g}^*(\mathbf{x}_i)^\top \mathbf{K}_{\mathbf{g}^*}^{-1} \mathbf{g}^*(\mathbf{x}_i),\end{aligned}$$

and also a solution of the problem

$$\max_{g_1, \dots, g_p \in \mathcal{H}} \sum_{j=1}^p \sum_{i=1}^n g_j(\mathbf{x}_i)^2 \quad \text{s.t. } g_j \perp g_k \text{ for } k \neq j \text{ and } \|g_j\|_{\mathcal{H}} = 1.$$

Nyström approximation via kernel PCA

Then, $\mathbf{K}_{\mathbf{g}^*} = \mathbf{I}$ and \mathbf{g}^* is also a solution of the problem

$$\max_{f_1, \dots, f_p \in \mathcal{H}} \sum_{i=1}^n \mathbf{f}(\mathbf{x}_i)^\top \mathbf{K}_{\mathbf{f}}^{-1} \mathbf{f}(\mathbf{x}_i),$$

since

$$\begin{aligned}\mathbf{f}^*(\mathbf{x}_i)^\top \mathbf{K}_{\mathbf{f}^*}^{-1} \mathbf{f}^*(\mathbf{x}_i) &= \mathbf{f}^*(\mathbf{x}_i)^\top \mathbf{U} \mathbf{\Delta}^{-1} \mathbf{U}^\top \mathbf{f}^*(\mathbf{x}_i) \\ &= \mathbf{g}^*(\mathbf{x}_i)^\top \mathbf{g}^*(\mathbf{x}_i) = \mathbf{g}^*(\mathbf{x}_i)^\top \mathbf{K}_{\mathbf{g}^*}^{-1} \mathbf{g}^*(\mathbf{x}_i),\end{aligned}$$

and also a solution of the problem

$$\max_{g_1, \dots, g_p \in \mathcal{H}} \sum_{j=1}^p \sum_{i=1}^n g_j(\mathbf{x}_i)^2 \quad \text{s.t. } g_j \perp g_k \text{ for } k \neq j \text{ and } \|g_j\|_{\mathcal{H}} = 1.$$

This is the kernel PCA formulation!

Nyström approximation via kernel PCA

Our first recipe with kernel PCA

Given a dataset of n training points $\mathbf{x}_1, \dots, \mathbf{x}_n$ in \mathcal{X} ,

- randomly choose a subset $\mathcal{Z} = [\mathbf{x}_{z_1}, \dots, \mathbf{x}_{z_m}]$ of $m \leq n$ training points;
- compute the $m \times m$ kernel matrix $\mathbf{K}_{\mathcal{Z}}$.
- **perform kernel PCA** to find the $p \leq m$ largest principal directions (parametrized by p vectors α_j in \mathbb{R}^m);

Then, every point \mathbf{x} in \mathcal{X} may be approximated by

$$\begin{aligned}\psi(\mathbf{x}) &= \mathbf{K}_{\mathbf{g}^*}^{-1/2} \mathbf{g}^*(\mathbf{x}) = \mathbf{g}^*(\mathbf{x}) = [g_1^*(\mathbf{x}), \dots, g_p^*(\mathbf{x})]^\top \\ &= \left[\sum_{i=1}^m \alpha_{1i} K(\mathbf{x}_{z_i}, \mathbf{x}), \dots, \sum_{i=1}^m \alpha_{pi} K(\mathbf{x}_{z_i}, \mathbf{x}) \right]^\top.\end{aligned}$$

Nyström approximation via kernel PCA

Remarks

- The vector $\psi(\mathbf{x})$ can be interpreted as coordinates of the projection of $\varphi(\mathbf{x})$ onto the (orthogonal) PCA basis.
- The complexity of training is $O(m^3)$ (eig decomposition of $\mathbf{K}_{\mathcal{Z}}$) + $O(m^2)$ kernel evaluations.
- The complexity of encoding a new point \mathbf{x} is $O(mp)$ (matrix vector multiplication) + $O(m)$ kernel evaluations.

Nyström approximation via kernel PCA

Remarks

- The vector $\psi(\mathbf{x})$ can be interpreted as coordinates of the projection of $\varphi(\mathbf{x})$ onto the (orthogonal) PCA basis.
- The complexity of training is $O(m^3)$ (eig decomposition of $\mathbf{K}_{\mathcal{Z}}$) + $O(m^2)$ kernel evaluations.
- The complexity of encoding a new point \mathbf{x} is $O(mp)$ (matrix vector multiplication) + $O(m)$ kernel evaluations.

The main issue is the encoding time, which depends linearly on $m > p$.

Nyström approximation via random sampling

A popular alternative is instead to select the anchor points among the training data points $\mathbf{x}_1, \dots, \mathbf{x}_n$ —that is,

$$\mathcal{F} := \text{span}(\varphi(\mathbf{x}_{z_1}), \dots, \varphi(\mathbf{x}_{z_p})).$$

In other words, choose $f_1 = \varphi(\mathbf{x}_{z_1}), \dots, f_p = \varphi(\mathbf{x}_{z_p})$.

Second recipe with random point sampling

Given a dataset of n training points $\mathbf{x}_1, \dots, \mathbf{x}_n$ in \mathcal{X} ,

- randomly choose a subset $\mathcal{Z} = [\mathbf{x}_{z_1}, \dots, \mathbf{x}_{z_p}]$ of p training points;
- compute the $p \times p$ kernel matrix $\mathbf{K}_{\mathcal{Z}}$.

Then, a new point \mathbf{x} is encoded as

$$\begin{aligned}\psi(\mathbf{x}) &= \mathbf{K}_{\mathcal{Z}}^{-1/2} \mathbf{f}_{\mathcal{Z}}(\mathbf{x}) \\ &= \mathbf{K}_{\mathcal{Z}}^{-1/2} [K(\mathbf{x}_{z_1}, \mathbf{x}), \dots, K(\mathbf{x}_{z_p}, \mathbf{x})]^\top\end{aligned}$$

Nyström approximation via random sampling

- The complexity of training is $O(p^3)$ (eig decomposition) + $O(p^2)$ kernel evaluations.
- The complexity of encoding a point \mathbf{x} is $O(p^2)$ (matrix vector multiplication) + $O(p)$ kernel evaluations.

Nyström approximation via random sampling

- The complexity of training is $O(p^3)$ (eig decomposition) + $O(p^2)$ kernel evaluations.
- The complexity of encoding a point \mathbf{x} is $O(p^2)$ (matrix vector multiplication) + $O(p)$ kernel evaluations.

The main issue complexity is better, but we lose the “optimality” of the PCA basis and the random choice of anchor points is not clever.

Nyström approximation via greedy approach

Better approximation can be obtained with a **greedy algorithm** that iteratively selects one column at a time with largest residual (Bach and Jordan, 2002; Smola and Shölkopf, 2000, Fine and Scheinberg, 2000).

At iteration k , assume that $\mathcal{Z} = \{\mathbf{x}_{z_1}, \dots, \mathbf{x}_{z_k}\}$; then, the residual for a data point \mathbf{x} encoded with k anchor points f_1, \dots, f_k is

$$\min_{\beta \in \mathbb{R}^k} \left\| \varphi(\mathbf{x}) - \sum_{j=1}^k \beta_j \varphi(\mathbf{x}_{z_j}) \right\|_{\mathcal{H}}^2,$$

which is equal to

$$\|\varphi(\mathbf{x})\|_{\mathcal{H}}^2 - \mathbf{f}_{\mathcal{Z}}(\mathbf{x})^\top \mathbf{K}_{\mathcal{Z}}^{-1} \mathbf{f}_{\mathcal{Z}}(\mathbf{x}),$$

and since $f_j = \varphi(\mathbf{x}_{z_j})$ for all j , the data point \mathbf{x}_i with largest residual is the one that maximizes

$$K(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{f}_{\mathcal{Z}}(\mathbf{x}_i) \mathbf{K}_{\mathcal{Z}}^{-1} \mathbf{f}_{\mathcal{Z}}(\mathbf{x}_i) \quad \text{with} \quad \mathbf{f}_{\mathcal{Z}}(\mathbf{x}_i) = [K(\mathbf{x}_{z_1}, \mathbf{x}_i), \dots, K(\mathbf{x}_{z_k}, \mathbf{x}_i)]^\top.$$

Nyström approximation via greedy approach

This brings us to the following algorithm

Third recipe with greedy anchor point selection

Initialize $Z = \emptyset$. For $k = 1, \dots, p$ do

- **data point selection**

$$z_k \leftarrow \operatorname{argmax}_{i \in \{1, \dots, n\}} K(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{f}_Z(\mathbf{x}_i) \mathbf{K}_Z^{-1} \mathbf{f}_Z(\mathbf{x}_i);$$

- **update the set \mathcal{Z}**

$$\mathcal{Z} \leftarrow \mathcal{Z} \cup \{\mathbf{x}_{z_k}\}.$$

Remarks

- A naive implementation costs $(O(k^2n + k^3))$ at every iteration.
- To get a reasonable complexity, one has to use simple linear algebra tricks (see next slide).

Nyström approximation via greedy approach

If $\mathcal{Z}' = \mathcal{Z} \cup \{\mathbf{z}\}$,

$$\mathbf{K}_{\mathcal{Z}'}^{-1} = \begin{bmatrix} \mathbf{K}_{\mathcal{Z}} & \mathbf{f}_{\mathcal{Z}}(\mathbf{z}) \\ \mathbf{f}_{\mathcal{Z}}(\mathbf{z})^\top & K(\mathbf{z}, \mathbf{z}) \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{K}_{\mathcal{Z}}^{-1} + \frac{1}{s} \mathbf{b} \mathbf{b}^\top & -\frac{1}{s} \mathbf{b} \\ -\frac{1}{s} \mathbf{b}^\top & \frac{1}{s} \end{bmatrix},$$

where s is the Schur complement $s = K(\mathbf{z}, \mathbf{z}) - \mathbf{f}_{\mathcal{Z}}(\mathbf{z}) \mathbf{K}_{\mathcal{Z}}^{-1} \mathbf{f}_{\mathcal{Z}}(\mathbf{z})$, and $\mathbf{b} = \mathbf{K}_{\mathcal{Z}}^{-1} \mathbf{f}_{\mathcal{Z}}(\mathbf{z})$.

Complexity analysis

- $\mathbf{K}_{\mathcal{Z}'}^{-1}$ can be obtained from $\mathbf{K}_{\mathcal{Z}}^{-1}$ and $\mathbf{f}_{\mathcal{Z}}(\mathbf{z})$ in $O(k^2)$ float operations; for that we need to always keep into memory the n vectors $\mathbf{f}_{\mathcal{Z}}(\mathbf{x}_i)$.
- updating the $\mathbf{f}_{\mathcal{Z}'}(\mathbf{x}_i)$'s from $\mathbf{f}_{\mathcal{Z}}(\mathbf{x}_i)$ requires n kernel evaluations;

The total training complexity is $O(p^2 n)$ float operations and $O(pn)$ kernel evaluations

Nyström approximation via K-means

When $\mathcal{X} = \mathbb{R}^d$, it is also possible to synthesize points $\mathbf{z}_1, \dots, \mathbf{z}_p$ such that they represented well some training data $\mathbf{x}_1, \dots, \mathbf{x}_n$, leading to the **Clustred Nyström approximation** (Zhang and Kwok, 2008).

Fourth recipe with K-means

- ① Perform the regular K-means algorithm on the training data, to obtain p centroids $\mathbf{z}_1, \dots, \mathbf{z}_p$ in \mathbb{R}^p .
- ② Define the anchor points $f_j = \varphi(\mathbf{z}_j)$ for $j = 1, \dots, p$, and perform the classical Nyström approximation.

Remarks

- The complexity is the same as Nyström with random selection (except for the K-means step);
- The method is data-dependent and can significantly outperform the other variants in practice.

Nyström approximation: conclusion

Concluding remarks

- The greedy selection rule is equivalent to computing an **incomplete Cholesky factorization** of the kernel matrix (Bach and Jordan, 2002; Schölkopf and Smola, 2000, Fine and Scheinberg, 2001);
- The techniques we have seen produce low-rank approximations of the kernel matrix $\mathbf{K} \approx \mathbf{L}\mathbf{L}^\top$;
- The method admits a **geometric interpretation** in terms of orthogonal projection onto a finite-dimensional subspace.
- The approximation **provides points in the RKHS**. As such, many operations on the mapping are valid (translations, linear combinations, projections), unlike the method that will come next.

Outline

5 Open Problems and Research Topics

- Large-scale learning with kernels
 - Motivation
 - Nyström approximations
 - Random Fourier features
- Foundations of deep learning from a kernel point of view

Random Fourier features [Rahimi and Recht, 2007] (1/5)

A large class of approximations for shift-invariant kernels are based on sampling techniques. Consider a real-valued positive-definite continuous translation-invariant kernel $K(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{x} - \mathbf{y})$ with $\kappa : \mathbb{R}^d \rightarrow \mathbb{R}$. Then, if $\kappa(0) = 1$, Bochner theorem tells us that κ is a **valid characteristic function for some probability measure**

$$\kappa(\mathbf{z}) = \mathbb{E}_{\mathbf{w}}[e^{i\mathbf{w}^\top \mathbf{z}}].$$

Remember indeed that, with the right assumptions on κ ,

$$\kappa(\mathbf{x} - \mathbf{y}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{\kappa}(\mathbf{w}) e^{i\mathbf{w}^\top \mathbf{x}} e^{-i\mathbf{w}^\top \mathbf{y}} d\mathbf{w},$$

and the probability measure admits a density $q(\mathbf{w}) = \frac{1}{(2\pi)^d} \hat{\kappa}(\mathbf{w})$ (non-negative, real-valued, sum to 1 since $\kappa(0) = 1$).

Random Fourier features (2/5)

Then,

$$\begin{aligned}\kappa(\mathbf{x} - \mathbf{y}) &= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{\kappa}(\mathbf{w}) e^{i\mathbf{w}^\top \mathbf{x}} e^{-i\mathbf{w}^\top \mathbf{y}} d\mathbf{w} \\ &= \int_{\mathbb{R}^d} q(\mathbf{w}) \cos(\mathbf{w}^\top \mathbf{x} - \mathbf{w}^\top \mathbf{y}) d\mathbf{w} \\ &= \int_{\mathbb{R}^d} q(\mathbf{w}) \left(\cos(\mathbf{w}^\top \mathbf{x}) \cos(\mathbf{w}^\top \mathbf{y}) + \sin(\mathbf{w}^\top \mathbf{x}) \sin(\mathbf{w}^\top \mathbf{y}) \right) d\mathbf{w} \\ &= \int_{\mathbb{R}^d} \int_{b=0}^{2\pi} \frac{q(\mathbf{w})}{2\pi} 2 \cos(\mathbf{w}^\top \mathbf{x} + b) \cos(\mathbf{w}^\top \mathbf{y} + b) d\mathbf{w} db \quad (\text{exercise}) \\ &= \mathbb{E}_{\mathbf{w} \sim q(\mathbf{w}), b \sim \mathcal{U}[0, 2\pi]} \left[\sqrt{2} \cos(\mathbf{w}^\top \mathbf{x} + b) \sqrt{2} \cos(\mathbf{w}^\top \mathbf{y} + b) \right]\end{aligned}$$

Random Fourier features (3/5)

Random Fourier features recipe

- Compute the Fourier transform of the kernel $\hat{\kappa}$ and define the probability density $q(\mathbf{w}) = \hat{\kappa}(\mathbf{w})/(2\pi)^d$;
- Draw p i.i.d. samples $\mathbf{w}_1, \dots, \mathbf{w}_p$ from q and p i.i.d. samples b_1, \dots, b_p from the uniform distribution on $[0, 2\pi]$;
- define the mapping

$$\mathbf{x} \mapsto \psi(\mathbf{x}) = \sqrt{\frac{2}{d}} \begin{bmatrix} \cos(\mathbf{w}_1^\top \mathbf{x} + b_1), \dots, \cos(\mathbf{w}_p^\top \mathbf{x} + b_p) \end{bmatrix}^\top.$$

Then, we have that

$$\kappa(\mathbf{x} - \mathbf{y}) \approx \langle \psi(\mathbf{x}), \psi(\mathbf{y}) \rangle_{\mathbb{R}^p}.$$

The two quantities are equal in expectation.

Random Fourier features (4/5)

Theorem, [Rahimi and Recht, 2007]

On any compact subset \mathcal{X} of \mathbb{R}^m , for all $\varepsilon > 0$,

$$\mathbb{P} \left[\sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} |\kappa(\mathbf{x} - \mathbf{y}) - \langle \psi(\mathbf{x}), \psi(\mathbf{y}) \rangle_{\mathbb{R}^p}| \geq \varepsilon \right] \leq 2^8 \left(\frac{\sigma_q \text{diam}(\mathcal{X})}{\varepsilon} \right)^2 e^{-\frac{p\varepsilon^2}{4(m+2)}},$$

where $\sigma_q^2 = \mathbb{E}_{\mathbf{w} \sim q(\mathbf{w})} [\mathbf{w}^\top \mathbf{w}]$ is the second moment of the Fourier transform of κ .

Remarks

- The convergence is uniform, **not data dependent**;
- Take the sequence $\varepsilon_p = \sqrt{\frac{\log(p)}{p}} \sigma_q \text{diam}(\mathcal{X})$; Then the term on the right converges to zero when p grows to infinity;
- **Prediction functions with Random Fourier features are not in \mathcal{H} .**

Random Fourier features (5/5)

Ingredients of the proof

- For a *fixed* pair of points \mathbf{x}, \mathbf{y} , Hoeffding's inequality says that

$$\mathbb{P}\left[\underbrace{|\kappa(\mathbf{x} - \mathbf{y}) - \langle \psi(\mathbf{x}), \psi(\mathbf{y}) \rangle_{\mathbb{R}^d}|}_{f(\mathbf{x}, \mathbf{y})} \geq \varepsilon\right] \leq 2e^{-\frac{p\varepsilon^2}{4}}.$$

- Consider a net (set of balls of radius r) that covers $\mathcal{X}_\Delta = \{\mathbf{x} - \mathbf{y} : (\mathbf{x}, \mathbf{y}) \in \mathcal{X}\}$ with at most $T = (4\text{diam}(\mathcal{X})/r)^m$ balls.
- Apply the Hoeffding's inequality to the centers $\mathbf{x}_i - \mathbf{y}_i$ of the balls;
- Use a basic union bound

$$\mathbb{P}\left[\sup_i f(\mathbf{x}_i, \mathbf{y}_i) \geq \frac{\varepsilon}{2}\right] \leq \sum_i \mathbb{P}\left[f(\mathbf{x}_i, \mathbf{y}_i) \geq \frac{\varepsilon}{2}\right] \leq 2Te^{-\frac{p\varepsilon^2}{8}}.$$

- Glue things together: control the probability for points (\mathbf{x}, \mathbf{y}) inside each ball, and adjust the radius r (a bit technical).

Outline

- 1 Kernels and RKHS
- 2 Kernel tricks and applications
- 3 Kernels and Graphs
- 4 Characterizing probabilities with kernels
- 5 Open Problems and Research Topics
 - Large-scale learning with kernels
 - Foundations of deep learning from a kernel point of view

Outline

5 Open Problems and Research Topics

- Large-scale learning with kernels
- Foundations of deep learning from a kernel point of view
 - Motivation
 - Deep kernel machines

Understanding deep learning

The challenge of deep learning theory

- **Over-parameterized** (millions of parameters)
- **Expressive** (can approximate any function)
- Complex **architectures** for exploiting problem structure
- Yet, **easy to optimize** with (stochastic) gradient descent!

Understanding deep learning

The challenge of deep learning theory

- **Over-parameterized** (millions of parameters)
- **Expressive** (can approximate any function)
- Complex **architectures** for exploiting problem structure
- Yet, **easy to optimize** with (stochastic) gradient descent!

A functional space viewpoint

- View deep networks as functions in some functional space;
- Non-parametric models, natural measures of complexity (e.g., norms).

Understanding deep learning

The challenge of deep learning theory

- **Over-parameterized** (millions of parameters)
- **Expressive** (can approximate any function)
- Complex **architectures** for exploiting problem structure
- Yet, **easy to optimize** with (stochastic) gradient descent!

A functional space viewpoint

- View deep networks as functions in some functional space;
- Non-parametric models, natural measures of complexity (e.g., norms).

What is an appropriate functional space?

Success of deep learning



ENGLISH - DETECTED ENGLISH CHI FRENCH CHINESE (TRADITIONAL)

where is the train station? × où est la gare? ☆

Microphone icon Speaker icon 27/5000 Edit icon Speaker icon □ ⋮

In the context of supervised learning

The goal is to learn a **prediction function** $f : \mathcal{X} \rightarrow \mathcal{Y}$ given labeled training data $(x_i, y_i)_{i=1,\dots,n}$ with x_i in \mathcal{X} , and y_i in \mathcal{Y} :

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

What is specific to multilayer neural networks?

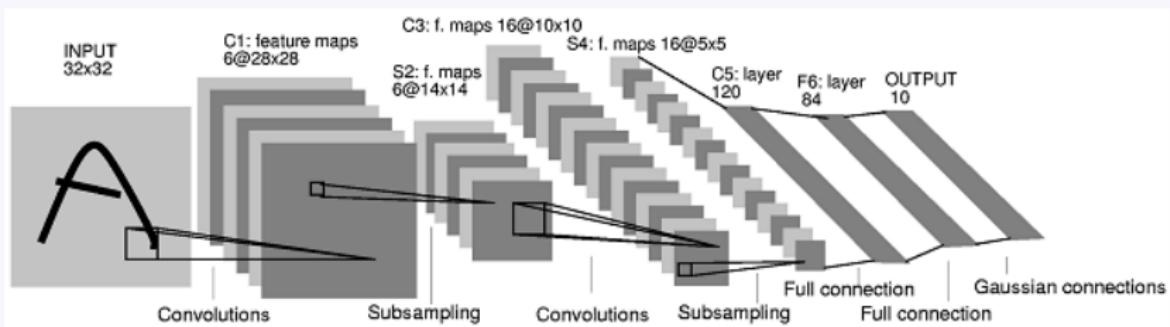
- The “neural network” space \mathcal{F} is explicitly parametrized by:

$$f(\mathbf{x}) = \sigma_k(\mathbf{A}_k \sigma_{k-1}(\mathbf{A}_{k-1} \dots \sigma_2(\mathbf{A}_2 \sigma_1(\mathbf{A}_1 \mathbf{x})) \dots)).$$

- Linear operations are either unconstrained (fully connected) or involve parameter sharing (e.g., convolutions).
- Finding the optimal $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_k$ yields a **non-convex** optimization problem.

Convolutional Neural Networks

Picture from LeCun et al. (1998)

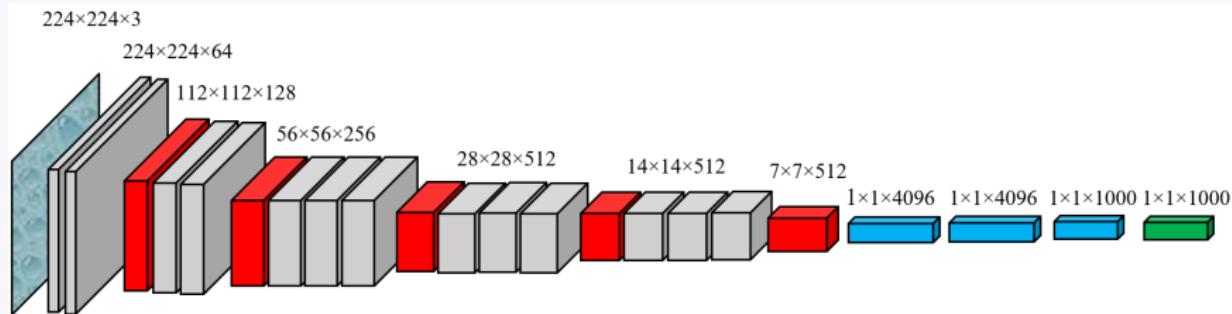


What are the main features of CNNs?

- they capture **compositional** and **multiscale** structures in images;
- they provide some **invariance**;
- they model the **local stationarity** of images at several scales;

Convolutional Neural Networks

(Simonyan and Zisserman, 2014)



What are the main features of CNNs?

- they capture **compositional** and **multiscale** structures in images;
- they provide some **invariance**;
- they model the **local stationarity** of images at several scales;

CNNs (Picture from unknown source)

ImageNet: 1000 image categories, 10M hand-labeled images; top-5 error rate.

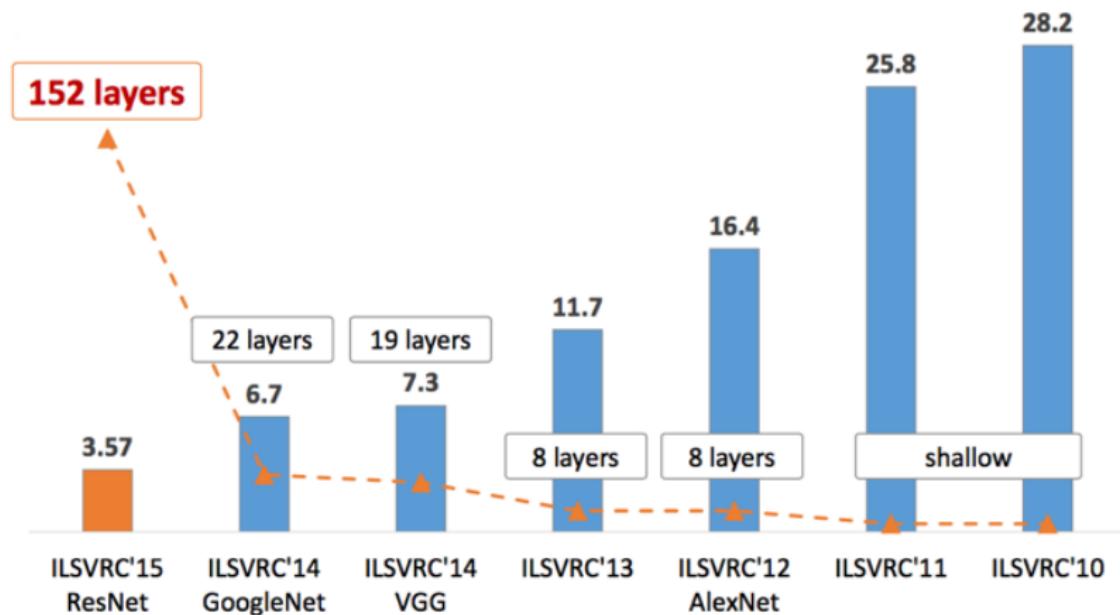


Figure: Top-5 error rate

Convolutional neural networks for biological sequences

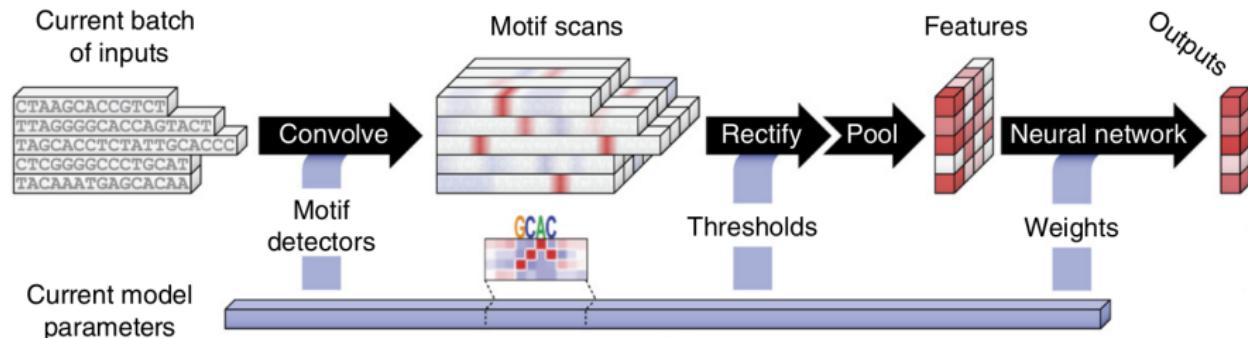


Figure: two-layer CNN architecture from Alipanahi et al. (2015)

- Sequences are represented by one-hot encoding ($A=(1,0,0,0), C=(0,1,0,0), \dots$).
- Single convolution layer followed by linear classifier.

Convolutional Neural Networks

What are current important problems to solve?

- ① lack of **stability and robustness** (see next slide).
- ② learning without **large amounts of data**.
- ③ making **interpretable** decisions.
- ④ ...

Adversarial examples, Picture from Kurakin et al. (2016)



Figure: Adversarial examples are generated by computer; then printed on paper; a new picture taken on a smartphone fools the classifier.

Adversarial examples



(b)

clean + noise → “**ostrich**” (Szegedy et al., 2013).

Adversarial examples



(a real ostrich)

Adversarial examples



adversarial
perturbation →



88% **tabby cat**

99% **guacamole**

<https://github.com/anishathalye/obfuscated-gradients>

Convolutional Neural Networks

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

The issue of regularization

- today, heuristics are used (DropOut, weight decay, early stopping)...
- ...but they are not sufficient.
- how to **control variations of prediction functions?**

$|f(\mathbf{x}) - f(\mathbf{x}')|$ should be close if \mathbf{x} and \mathbf{x}' are “similar”.

- what does it mean for x and x' to be “similar”?
- what should be a good **regularization function** Ω ?

Outline

5 Open Problems and Research Topics

- Large-scale learning with kernels
- Foundations of deep learning from a kernel point of view
 - Motivation
 - Deep kernel machines

Relevant concepts

- Dot-product kernels:

$$K(x, x') = \kappa(x^\top x') \quad \text{or} \quad K(x, x') = \|x\| \|x'\| \kappa\left(\frac{x^\top x'}{\|x\| \|x'\|}\right)$$

- Hierarchical composition of feature spaces:

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle \text{ with } \Phi(x) = \varphi_2(\varphi_1(x))$$

- **NTK**: Asymptotic behavior of over-parametrized deep neural networks learned by gradient descent.
- **CKN**: Convolutional and hierarchical kernel constructions + **end-to-end learning** with kernels.

Relevant concepts

- Dot-product kernels:

$$K(x, x') = \kappa(x^\top x') \quad \text{or} \quad K(x, x') = \|x\| \|x'\| \kappa\left(\frac{x^\top x'}{\|x\| \|x'\|}\right)$$

- Hierarchical composition of feature spaces:

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle \text{ with } \Phi(x) = \varphi_2(\varphi_1(x))$$

- NTK: Asymptotic behavior of over-parametrized deep neural networks learned by gradient descent.
- CKN: Convolutional and hierarchical kernel constructions + end-to-end learning with kernels.

What does it mean to do end-to-end learning with kernels?

Kernels for deep models: deep kernel machines

Hierarchical kernels (Cho and Saul, 2009b)

- Kernels can be constructed **hierarchically**

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle \text{ with } \Phi(x) = \varphi_2(\varphi_1(x))$$

- e.g., dot-product kernels on the sphere

$$K(x, x') = \kappa_2(\langle \varphi_1(x), \varphi_1(x') \rangle) = \kappa_2(x^\top x')$$

Kernels for deep models: deep kernel machines

Hierarchical kernels (Cho and Saul, 2009b)

- Kernels can be constructed **hierarchically**

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle \text{ with } \Phi(x) = \varphi_2(\varphi_1(x))$$

- e.g., dot-product kernels on the sphere

$$K(x, x') = \kappa_2(\langle \varphi_1(x), \varphi_1(x') \rangle) = \kappa_2(x^\top x')$$

A classical old result (Schoenberg, 1942)

Let $\mathcal{X} = \mathbb{S}$ be the unit sphere of some Hilbert space \mathcal{H}_0 . The kernel $K : \mathcal{X}^2 \rightarrow \mathbb{R}$

$$K(\mathbf{x}, \mathbf{y}) = \kappa(\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{H}_0}),$$

is positive definite for all \mathcal{H}_0 if and only if κ is smooth and admits an expansion $\kappa(u) = \sum_i a_i u^i$ with non-negative coefficients a_i .

Kernels for deep models: dot-product kernels

linear kernel	$\langle z, z' \rangle$
exponential kernel	$e^{\alpha(\langle z, z' \rangle - 1)}$
inverse polynomial kernel	$\frac{1}{2 - \langle z, z' \rangle}$
polynomial kernel of degree p	$(c + \langle z, z' \rangle)^p$
arc-cosine kernel of degree 1	$\frac{1}{\pi} (\sin(\theta) + (\pi - \theta) \cos(\theta))$ with $\theta = \arccos(\langle z, z' \rangle)$
Vovk's kernel of degree 3	$\frac{1}{3} \left(\frac{1 - \langle z, z' \rangle^3}{1 - \langle z, z' \rangle} \right) = \frac{1}{3} (1 + \langle z, z' \rangle + \langle z, z' \rangle^2)$

Kernels for deep models: dot-product kernels

linear kernel	$\langle z, z' \rangle$
exponential kernel	$e^{\alpha(\langle z, z' \rangle - 1)}$
inverse polynomial kernel	$\frac{1}{2 - \langle z, z' \rangle}$
polynomial kernel of degree p	$(c + \langle z, z' \rangle)^p$
arc-cosine kernel of degree 1	$\frac{1}{\pi} (\sin(\theta) + (\pi - \theta) \cos(\theta))$ with $\theta = \arccos(\langle z, z' \rangle)$
Vovk's kernel of degree 3	$\frac{1}{3} \left(\frac{1 - \langle z, z' \rangle^3}{1 - \langle z, z' \rangle} \right) = \frac{1}{3} (1 + \langle z, z' \rangle + \langle z, z' \rangle^2)$

Remark

if $\|z\| = \|z'\| = 1$, the exponential kernel recovers the Gaussian kernel

$$\kappa_{\exp}(\langle z, z' \rangle) = e^{\alpha(\langle z, z' \rangle - 1)} = e^{-\frac{\alpha}{2} \|z - z'\|^2},$$

Kernels for deep models: random feature kernels

$$f_\theta(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^m v_i \sigma(w_i^\top x), \quad m \rightarrow \infty$$

Random feature kernels (RF, Neal, 1996; Rahimi and Recht, 2007)

- $\theta = (v_i)_i$, fixed random weights $w_i \sim N(0, I)$

$$K_{RF}(x, y) = \mathbb{E}_{w \sim N(0, I)} [\sigma(w^\top x) \sigma(w^\top y)]$$

Kernels for deep models: random feature kernels

$$f_\theta(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^m v_i \sigma(w_i^\top x), \quad m \rightarrow \infty$$

Random feature kernels (RF, Neal, 1996; Rahimi and Recht, 2007)

- $\theta = (v_i)_i$, fixed random weights $w_i \sim N(0, I)$

$$K_{RF}(x, y) = \mathbb{E}_{w \sim N(0, I)} [\sigma(w^\top x) \sigma(w^\top y)]$$

- integral representations are not only available for t.i. kernels. They also work for several dot-product kernels (Cho and Saul, 2009b):

$$k_n(x, y) = \frac{1}{\pi} \|x\|^n \|y\|^n J_n(\theta) \quad \text{with} \quad \theta = \cos^{-1} \left(\frac{x^\top y}{\|x\| \|y\|} \right)$$

with

$$J_n(\theta) = (-1)^n (\sin \theta)^{2n+1} \left(\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \right)^n \left(\frac{\pi - \theta}{\sin \theta} \right)$$

Kernels for deep models: random feature kernels

$$f_\theta(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^m v_i \sigma(w_i^\top x), \quad m \rightarrow \infty$$

Random feature kernels (RF, Neal, 1996; Rahimi and Recht, 2007)

- $\theta = (v_i)_i$, fixed random weights $w_i \sim N(0, I)$

$$K_{RF}(x, y) = \mathbb{E}_{w \sim N(0, I)} [\sigma(w^\top x) \sigma(w^\top y)]$$

- integral representations are not only available for t.i. kernels. They also work for several dot-product kernels (Cho and Saul, 2009b):

$$k_n(x, y) = \frac{1}{\pi} \|x\|^n \|y\|^n J_n(\theta) \quad \text{with} \quad \theta = \cos^{-1} \left(\frac{x^\top y}{\|x\| \|y\|} \right)$$

with

$$\begin{cases} J_0(\theta) &= \pi - \theta \\ J_1(\theta) &= \sin(\theta) + (\pi - \theta) \cos(\theta) \\ J_2(\theta) &= 3 \sin(\theta) \cos(\theta) + (\pi - \theta)(1 + 2 \cos^2(\theta)) \end{cases}$$

Kernels for deep models: random feature kernels

Theorem, (Cho and Saul, 2009a)

Consider

$$k_n(x, y) = \frac{1}{\pi} \|x\|^n \|y\|^n J_n(\theta) \quad \text{with} \quad \theta = \cos^{-1} \left(\frac{x^\top y}{\|x\| \|y\|} \right).$$

Then

$$k_n(x, y) = \mathbb{E}_{w \sim N(0, I)} [\sigma(w^\top x) \sigma(w^\top y)],$$

with $\sigma(u) = \frac{u^n}{\sqrt{2}} (1 + \text{sign}(u))$.

- Note that $k_1(x, y) = \mathbb{E}_{w \sim N(0, I)} [\text{RELU}(w^\top x) \text{RELU}(w^\top y)]$.
- One of the fundamental tool to analyze RELU networks.

Kernels for deep models: neural tangent kernels

$$f_\theta(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^m v_i \sigma(w_i^\top x), \quad m \rightarrow \infty$$

Neural tangent kernels (NTK, Jacot et al., 2018)

- $\theta = (v_i, w_i)_i$, initialization $\theta_0 \sim N(0, I)$
- **Lazy training** (Chizat et al., 2019): θ stays close to θ_0 when training with large m

$$f_\theta(x) \approx f_{\theta_0}(x) + \langle \theta - \theta_0, \nabla_\theta f_\theta(x)|_{\theta=\theta_0} \rangle.$$

- Gradient descent for $m \rightarrow \infty \approx$ kernel ridge regression with **neural tangent kernel**

Kernels for deep models: neural tangent kernels

$$f_\theta(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^m v_i \sigma(w_i^\top x), \quad m \rightarrow \infty$$

Neural tangent kernels (NTK, Jacot et al., 2018)

- $\theta = (v_i, w_i)_i$, initialization $\theta_0 \sim N(0, I)$
- **Lazy training** (Chizat et al., 2019): θ stays close to θ_0 when training with large m

$$f_\theta(x) \approx f_{\theta_0}(x) + \langle \theta - \theta_0, \nabla_\theta f_{\theta_0}(x) |_{\theta=\theta_0} \rangle.$$

- Gradient descent for $m \rightarrow \infty \approx$ kernel ridge regression with **neural tangent kernel**

$$K_{NTK}(x, y) = \lim_{m \rightarrow \infty} \langle \nabla_\theta f_{\theta_0}(x), \nabla_\theta f_{\theta_0}(y) \rangle$$

Kernels for deep models: neural tangent kernels

$$f_\theta(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^m v_i \sigma(w_i^\top x), \quad m \rightarrow \infty$$

Neural tangent kernels (NTK, Jacot et al., 2018)

- $\theta = (v_i, w_i)_i$, initialization $\theta_0 \sim N(0, I)$
- **Lazy training** (Chizat et al., 2019): θ stays close to θ_0 when training with large m

$$f_\theta(x) \approx f_{\theta_0}(x) + \langle \theta - \theta_0, \nabla_\theta f_\theta(x) |_{\theta=\theta_0} \rangle.$$

- Gradient descent for $m \rightarrow \infty \approx$ kernel ridge regression with **neural tangent kernel**

$$K_{NTK}(x, y) = \mathbb{E}_{\mathbf{w}}[\sigma(\mathbf{w}^\top x)\sigma(\mathbf{w}^\top y) + (x^\top y)\sigma'(\mathbf{w}^\top x)\sigma'(\mathbf{w}^\top y)]$$

- with RELU networks, we obtain a dot-product kernel.

Conclusion of the course

What we saw

- Basic definitions of p.d. kernels and RKHS
- How to use RKHS in machine learning
- The importance of the choice of kernels, and how to include “prior knowledge” there.
- Several approaches for kernel design (there are many!)
- Review of kernels for strings and on graphs
- Recent research topics about kernel methods

What we did not see

- How to **automatize** the process of kernel design (kernel selection? kernel optimization?)
- How to deal with **non p.d.** kernels
- Bayesian view of kernel methods, called **Gaussian processes**.

References I

- B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.
- N. Aronszajn. Theory of reproducing kernels. *Trans. Am. Math. Soc.*, 68:337 – 404, 1950.
URL <http://www.jstor.org/stable/1990404>.
- K. M. Borgwardt and H.-P. Kriegel. Shortest-path kernels on graphs. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 74–81, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2278-5. doi: <http://dx.doi.org/10.1109/ICDM.2005.132>.
- L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. 2019.
- Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Adv. NIPS*, 2009a.
- Y. Cho and L. K. Saul. Kernel methods for deep learning. 2009b.
- T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: hardness results and efficient alternatives. In B. Schölkopf and M. Warmuth, editors, *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory and the Seventh Annual Workshop on Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pages 129–143, Heidelberg, 2003. Springer. doi: 10.1007/b12006. URL <http://dx.doi.org/10.1007/b12006>.

References II

- Z. Harchaoui and F. Bach. Image classification with segmentation graph kernels. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8. IEEE Computer Society, 2007. doi: 10.1109/CVPR.2007.383049. URL <http://dx.doi.org/10.1109/CVPR.2007.383049>.
- C. Helma, T. Cramer, S. Kramer, and L. De Raedt. Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds. *J. Chem. Inf. Comput. Sci.*, 44(4):1402–11, 2004. doi: 10.1021/ci034254q. URL <http://dx.doi.org/10.1021/ci034254q>.
- A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. 2018.
- Y. Jiao and J.-P. Vert. The Kendall and Mallows kernels for permutations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. doi: 10.1109/TPAMI.2017.2719680. URL <http://dx.doi.org/10.1109/TPAMI.2017.2719680>.
- R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 315–322, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. URL <http://dx.doi.org/10.1109/5.726791>.

References III

- P. Mahé and J. P. Vert. Graph kernels based on tree patterns for molecules. *Mach. Learn.*, 75(1):3–35, 2009. doi: 10.1007/s10994-008-5086-2. URL
<http://dx.doi.org/10.1007/s10994-008-5086-2>.
- P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert. Extensions of marginalized graph kernels. In R. Greiner and D. Schuurmans, editors, *Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004)*, pages 552–559. ACM Press, 2004.
- R. M. Neal. *Bayesian learning for neural networks*. Springer, 1996.
- A. Nicholls. Oechem, version 1.3.4, openeye scientific software. website, 2005.
- A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Adv. NIPS*, 2007.
- J. Ramon and T. Gärtner. Expressivity versus efficiency of graph kernels. In T. Washio and L. De Raedt, editors, *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*, pages 65–74, 2003.
- I. Schoenberg. Positive definite functions on spheres. *Duke Math. J.*, 1942.
- B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002. URL
<http://www.learning-with-kernels.org>.
- B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. MIT Press, The MIT Press, Cambridge, Massachusetts, 2004.

References IV

- J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.
- N. Shervashidze and K. M. Borgwardt. Fast subtree kernels on graphs. In *Advances in Neural Information Processing Systems*, pages 1660–1668, 2009.
- N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt. Efficient graphlet kernels for large graph comparison. In *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 488–495, Clearwater Beach, Florida USA, 2009. Society for Artificial Intelligence and Statistics.
- N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *The Journal of Machine Learning Research*, 12: 2539–2561, 2011.
- K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.
- C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
- 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>.
- V. N. Vapnik. *Statistical Learning Theory*. Wiley, New-York, 1998.

References V

- G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1990.
- B. Weisfeiler and A. A. Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia, Ser. 2, 9*, 1968.