

Introduction to learning in RKHS

Advanced Statistics

Florence d'Alché

March 27, 2026

LTCI, Télécom Paris

Table of contents

1. Introduction to Kernels and Reproducing Kernel Hilbert Spaces
2. Learning with kernels
3. The example of Kernel Ridge Regression
4. Revisiting SVM

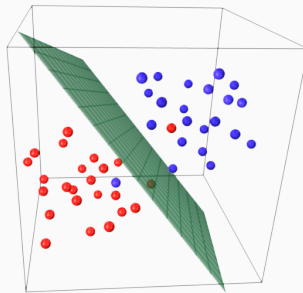
Why being interesting by doing statistics with kernels ?

Define nonlinear models and methods as if they were linear

Leverage non-euclidean data as if they were euclidean

Statistical Learning : key components

- Data representation
- Hypothesis space / architecture
- Loss function
- Constraints and regularization
- Learning algorithm
- Evaluation metrics
- Model selection



Example of supervised learning

General principles to build a model

- Local average (k-neighbours, tree) : $h(x) := \sum_{j=1}^J 1_{x \in \mathcal{R}_j} \tilde{y}_j$
- Agregation/ committee (random forest, boosting) :
 $h(x) := \sum_1^T \alpha_t h_t(x)$
- Layer composition (deep architectures) : $h(x) := h_L \circ h_{L-1} \circ \dots \circ h_1(x)$
- Kernel based models : $h(x) := \sum_{i=1}^n \alpha_i k(x, x_i)$

N.B. the models are here illustrated for regression tasks

Parametric/non-parametric modeling

Parametric modeling : h is entirely determined by a parameter θ (vectors, matrices, tensors) - The form of the function is known.

- Linear models
- Neural networks

Non-parametric modeling : the form of h is not known before training except that training data play the role of main parameters (there are other parameters)

$$h(x) := \sum_{i=1}^n \alpha_i(x) y_i \quad (1)$$

Kernel models enter this category.

Kernel Machines

- Use kernel as a building block for modeling : a **kernel** is a **similarity** with nice properties
- Benefit from the mathematical framework aka Reproducing Kernel Hilbert Space theory
- Principled way to learning with a minimal number of hyperparameters
- Can be applied instead of linear models everywhere : dimension reduction, clustering, regression, novelty detection, classification, multitask...
- Exploit duality principle and convex programming machinery
- Can be shallow or deep
- Suffer from data volume if not **approximated**

Using kernel as a building block

Kernel : a (very) first informal definition

For $(x, x') \in \mathcal{X} \times \mathcal{X}$ (data space),

$$k(x, x') := \text{similarity between } x \text{ and } x',$$

with a very special property : there exist \mathcal{H} and φ such that :

$$k(x, x') := \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$$

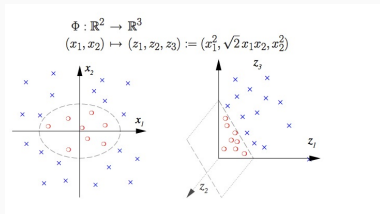
We will say more about \mathcal{H} and φ in a minut.

Examples of kernels : kernels on vectors,

► Kernels on structured objects

A classic kernel example

Polynomial kernel in \mathbb{R}^2 :



$$k_{pol}(x, x') = \langle x, x' \rangle_{\mathbb{R}^2}^2 = (x_1x'_1 + x_2x'_2)^2 = x_1^2x_1'^2 + 2x_1x'_1x_2x'_2 + x_2^2x_2'^2$$

I can exhibit : $\varphi(x) \in \mathbb{R}^3$ defined by :

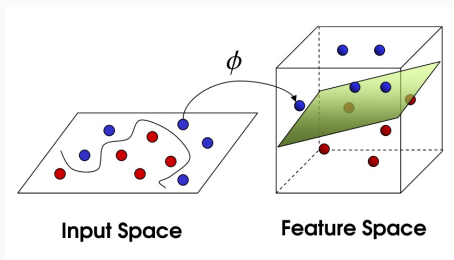
$\varphi_1(x) = x_1^2$, $\varphi_2(x) = \sqrt{2}x_1x_2$, $\varphi_3(x) = x_2^2$ such that :

$$k_{pol}(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathbb{R}^3}$$

KERNEL TRICK : I can compute k_{pol} without working in $\mathcal{H} = \mathbb{R}^3$.

Kernel trick

Instead of working \mathcal{X} , I do as if I was sending my data into a space \mathcal{H} but without the pain of working in it. The inner products are given to me by the means of $k(x, x')$

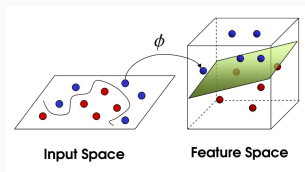


Let us define the model : $f(x) = \beta^T \varphi(x)$, I can now solve the linear regression problem.

Now let us go beyond the polynomial kernel.

How do we do that ?

Teaser about Reproducing kernel Hilbert space



If k is a symmetric, positive definite function, then there exists a unique feature space \mathcal{H}_k and feature map φ such that :

- $k(x, x') := \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$
- and $k(\cdot, x) \in \mathcal{H}$ satisfies the reproducing property :

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

$$\mathcal{H}_k = \overline{\text{Span} \{ \varphi(x) = k(\cdot, x) : x \in \mathcal{X} \}} \subset \mathcal{F}(\mathcal{X}, \mathbb{R}) \text{ and } \varphi(x) = k(\cdot, x).$$

Use kernels as a building block for your model f in order to work with f_{lin} :

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

as "easily" as working with

$$f_{lin}(x) = \langle \beta, x \rangle.$$

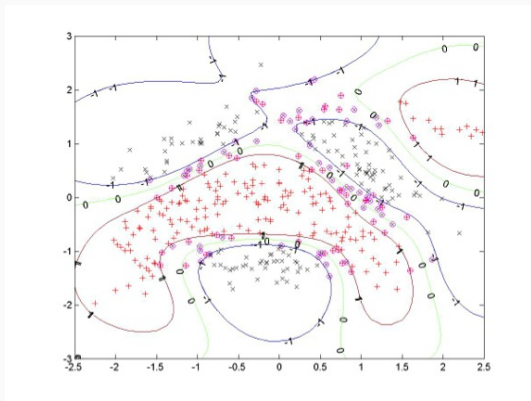
Reason 1 to know about Kernels and Reproducing Kernel Hilbert Spaces : building nonlinear model

Use linear methods to solve nonlinear problems :

- Linear regression (least-squares, ridge)
- Linear classification
- Linear dimension reduction, canonical correlation analysis
- Clustering : k-means
- Linear modeling of time-series
- Kernel ridge regression, Support Vector Regression
- Support Vector Classification
- Kernel PCA
- Clustering : kernel-k-means, spectral clustering
- Kernel Dynamic modeling

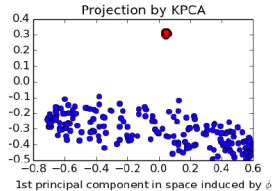
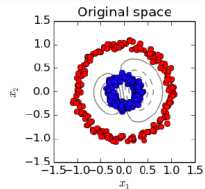
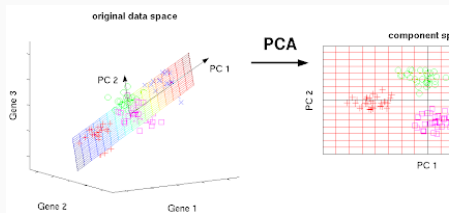
Gaussian kernel

Gaussian kernel over \mathbb{R}^d : $k(x, x') = \exp(-\gamma\|x - x'\|^2)$.



Results of a Support Vector Machine on a nonlinear problem using a Gaussian kernel

Teaser : Kernel Principal Component



Principal Component Analysis

Assume data are centered ($\sum_i x_i = 0$).

$$\begin{aligned} \max_{v \in \mathbb{R}^p} \quad & \frac{1}{n} \sum_{i=1}^n (x_i^T v)^2 \\ \text{s.t.} \quad & \|v\|^2 = 1 \end{aligned}$$

Kernel PCA

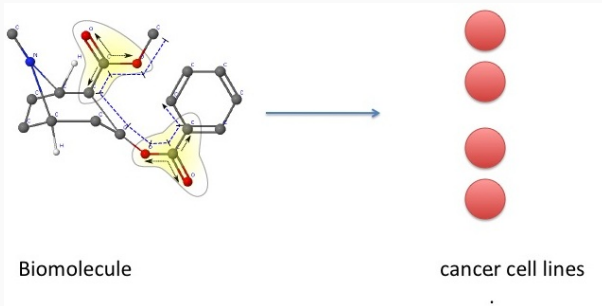
- Idea : replace the projection operator by a nonlinear function in the RKHS \mathcal{H}_k
- Let ϕ be a feature map associated to k
- Assume that $\sum_i \varphi(x_i) = 0$ (otherwise center it)
- Intuition : notice that $f(x_i) = \langle f, \varphi(x_i) \rangle_{\mathcal{H}_k}$
- We assume that : $\sum_{i=1}^n \varphi(x_i) = 0$

The first principal component in the feature space can be found by solving :

$$\begin{aligned} \max_{f \in \mathcal{H}_k} \sum_{i=1}^n f(x_i)^2 &= \langle f, \varphi(x_i) \rangle_{\mathcal{H}_k}^2 \\ \text{s.t. } \|f\|_{\mathcal{H}_k}^2 &= 1 \end{aligned}$$

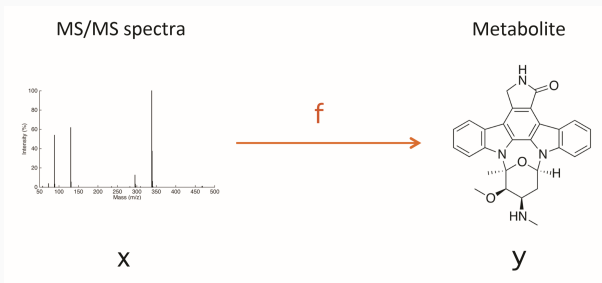
Reason 2 to know about Kernel Machines : complex inputs

Use non-vectorial data as input :



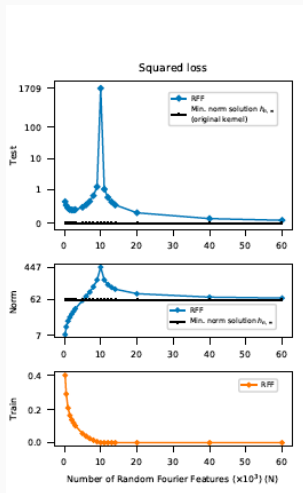
Reason 3 : complex outputs

Use vectorial and non data as outputs to solve structured output prediction



Reason 4 : understanding neural networks

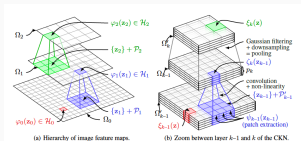
- Equivalence between a shallow kernel machine and a specific neural layer : Random Fourier Feature (Rahimi and Recht, 2007)
- "overfitted" networks / interpolated classifiers : understanding the near-zero generalization error for deep/overparametrized models (Belkin et al. 2017, 18, 19)



Reason 5 : understanding neural networks

- Understanding/approximating convolutional networks with RKHS : ► CKN, Mairal et al. (2014)
- Neural Tangent Kernel :

Jacot et al. (2018)



Introduction to Kernels and Reproducing Kernel Hilbert Spaces

How to learn with kernels ?



fig/pillar-rkhs-small.png

- We introduce the minimal set of theoretical tools to work in a principled manner and solve machine learning problems with kernels

Kernel Definition

Let \mathcal{X} be a non-empty set.

Definition 1 (Positive Definite Symmetric Kernel)

A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be a Positive Definite Symmetric (PDS) Kernel if

- $\forall (x, x') \in \mathcal{X} \times \mathcal{X}, \quad k(x, x') = k(x', x)$ (symmetry)
- $\forall (x_1, \dots, x_n) \in \mathcal{X}^n, \forall \alpha \in \mathbb{R}^n, \sum_{i,j=1}^n \alpha_i k(x_i, x_j) \alpha_j = \alpha^T K \alpha \geq 0$ (positiveness)

where $K \in \mathbb{R}^{n \times n}$ is a matrix whose entries are : $K_{ij} = k(x_i, x_j)$.

N.B. We ask that any Gram matrix built from a finite number of elements in \mathcal{X} has its eigenvalues greater or equal to 0, e.g. to be semi-definite positive.

Our classic kernels on \mathbb{R}^d

- Linear kernel : $k(x, x') = x^T x'$
- Gaussian kernel : $k(x, x') = \exp(-\gamma \|x - x'\|^2)$
- Polynomial kernels : $k(x, x') = (x^T x' + 1)^m$

Closure properties of kernels

closure property	feature space representation
a) $K_1(x, y) + K_2(x, y)$	$\Phi(x) = (\Phi_1(x), \Phi_2(x))^T$
b) $\alpha K_1(x, y)$ for $\alpha > 0$	$\Phi(x) = \sqrt{\alpha} \Phi_1(x)$
c) $K_1(x, y) K_2(x, y)$	$\Phi(x)_{ij} = \Phi_1(x)_i \Phi_2(x)_j$ (tensor product)
d) $f(x)f(y)$ for any f	$\Phi(x) = f(x)$
e) $x^T A y$ for $A \succeq 0$ (i.e. psd)	$\Phi(x) = L^T x$ for $A = L L^T$ (Cholesky)

From those properties, we conclude that a polynomial of kernels is still a kernel.
the pointwise limit of kernels is also a kernel.

Exercise: : prove these properties.

Cauchy-Schwartz inequality

Cauchy-Schwartz inequality

Let k be a PDS kernel then $\forall (x, z) \in \mathcal{X}^2$, we have :

$$k(x, z)^2 \leq k(x, x)k(z, z)$$

Proof : consider the matrix :

$$K = \begin{pmatrix} k(x, x) & k(x, z) \\ k(z, x) & k(z, z) \end{pmatrix}$$

then, $\det(K) = k(x, x)k(z, z) - k(x, z)^2$. We know that K is semi-definite positive so $\det(K) \geq 0$.

Reminder about Hilbert space

Hilbert spaces are the analogous of euclidean spaces, they are especially useful for functional analysis and quantum physics.

Definition 2 (Hilbert Space)

A **Hilbert space** is a complete inner-product space ; that is, an inner product space in which every Cauchy sequence is convergent for the metric induced by the inner product(norm).

N.B. We will note the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, the associated norm $\| \cdot \|$ satisfies : $\|h\|^2 = \langle h, h \rangle_{\mathcal{H}}$.

Definition 3 (Reproducing Kernel Hilbert space - RKHS)

Let \mathcal{H} be a Hilbert space of \mathbb{R} -valued functions on non-empty set \mathcal{X} . A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a **reproducing kernel** of \mathcal{H} , and \mathcal{H} is a reproducing kernel Hilbert space if :

- $\forall x \in \mathcal{X}, k(\cdot, x) \in \mathcal{H},$
- $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}, \langle f, k(\cdot, x) \rangle_{\mathcal{H}} = f(x)$ (**reproducing property**).

In particular, for any $x, y \in \mathcal{X},$

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

Building a RKHS from a PDS kernel k

Theorem 4 (Reproducing Kernel Hilbert space induced by a kernel (Aronszajn, 1950))

Let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a positive definite symmetric kernel. Then, there exists a Hilbert space \mathcal{H} and a function $\varphi : \mathcal{X} \rightarrow \mathcal{H}$ such that :

$$\forall (x, x') \in \mathcal{X} \times \mathcal{X}, k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$$

Furthermore, \mathcal{H} is the Reproducing Kernel Hilbert Space associated to k and k is its reproducing kernel, e.g. has the following reproducing property :

$$\forall f \in \mathcal{H}, \forall x \in \mathcal{X}, f(x) = \langle f, k(\cdot, x) \rangle$$

Constructive Proof 1/5

Let us define $\mathcal{H}_0 = \{\sum_{i \in I} \alpha_i k(\cdot, x_i), x_i \in \mathcal{X}, |I| < \infty\}$.

\mathcal{H}_0 is the set of finite linear combinations of functions $x \rightarrow k(\cdot, x_i)$.

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

$$\begin{aligned}\forall f, g, \in \mathcal{H}_0^2, f(\cdot) &= \sum_{i \in I} \alpha_i k(\cdot, x_i) \\ g(\cdot) &= \sum_{j \in J} \beta_j k(\cdot, z_j)\end{aligned}$$

by

$$\langle f, g \rangle_{\mathcal{H}_0} := \sum_{i \in I, j \in J} \alpha_i \beta_j k(x_i, z_j)$$

We notice that :

$$\langle f, g \rangle = \sum_{j \in J} \beta_j f(z_j) = \sum_{i \in I} \alpha_i g(x_i)$$

meaning that this product between f and g does not depend on the expansions of f or g .

It also verifies $\langle f, f \rangle_{\mathcal{H}_0} \geq 0$ and

$$\langle f, f \rangle_{\mathcal{H}_0} = 0$$

if and only if $f = 0$.

We define a norm from this product :

$$\|f\|_{\mathcal{H}_0}^2 := \langle f, f \rangle_{\mathcal{H}_0} = \sum_{i \in I, j \in I} \alpha_i K_{ij} \alpha_j = \alpha^T K \alpha$$

where K is the Gram matrix associated to k .

Remark : we have a Cauchy-Schwartz inequality for PDS kernels (that we will use).

Constructive Proof 3/5

We need to prove that we have the reproducing property :

$$\begin{aligned}\langle f, k(\cdot, x) \rangle_{\mathcal{H}_0} &= \langle \sum_i \alpha_i k(\cdot, x_i), k(\cdot, x) \rangle \\ &= \sum_i \alpha_i k(x, x_i) \\ &= f(x)\end{aligned}$$

Now \mathcal{H}_0 is named a pre-Hilbert space and we need to complete it with the limits of Cauchy sequences to get a **Hilbert space**.

Let $(f_n)_n$, a Cauchy sequence of functions of \mathcal{H}_0 .

$$\forall \epsilon > 0, \exists N \in \mathbb{N}, \forall p, q > N, \|f_p - f_q\|^2 < \epsilon$$

Let us consider $\mathcal{H} = \{\text{lim of Cauchy sequences from } \mathcal{H}_0\}$.

note that $\mathcal{H}_0 \subset \mathcal{H}$.

To ensure the reproducing property for these new functions, we need to have the pointwise convergence of Cauchy Sequences $(f_n(x))_n$ for $x \in \mathcal{X}$.

Constructive Proof 4/6

Proof of pointwise convergence of $(f_n(x))_n$ for $x \in \mathcal{X}$

$\forall x \in \mathcal{X}, \forall (p, q) \in \mathbb{N}^2,$

$$\begin{aligned} |f_p(x) - f_q(x)| &= | \langle f_p, k(\cdot, x) \rangle - \langle f_q, k(\cdot, x) \rangle | \\ &= | \langle f_p - f_q, k(\cdot, x) \rangle | \\ &\leq \sqrt{\langle f_p - f_q, f_p - f_q \rangle} \sqrt{k(x, x)} \\ &\leq \|f_p - f_q\| \sqrt{k(x, x)} \end{aligned}$$

Then it comes that $(f_n(x))_n$ is a Cauchy Sequence in \mathbb{R} and thus has a limit.

now $f(x) := \lim_{n \rightarrow \infty} f_n(x)$.

Now let us consider the space \mathcal{H} of functions that are pointwise limits of Cauchy Sequences in \mathcal{H}_0 . Note that $\mathcal{H}_0 \subset \mathcal{H}$.

Our goal is to define an inner product on \mathcal{H} , and show the reproducing property, i.e. that \mathcal{H} is a RKHS with k as reproducing kernel.

Intermediate result : Any Cauchy Sequence $(f_n)_n \in \mathcal{H}_0$ that converges pointwise satisfies : $\lim_{n \rightarrow \infty} \|f_n\|_{\mathcal{H}_0} = 0..$

Proof by using Cauchy-Schwartz.

Proof 5/6

We then show that if we have two Cauchy sequences (f_n) and (g_n) in \mathcal{H}_0 , then their inner product $\langle f_n, g_n \rangle_{\mathcal{H}_0}$ converges (using Cauchy-Schwartz and reproducing property) and the limit only depends on the limits of the functions, f and g .

We then can define for two functions f and g in \mathcal{H} that are pointwise limit of $(f_n) \in \mathcal{H}_0$ and $(g_n) \in \mathcal{H}_0$:

$$\langle f, g \rangle_{\mathcal{H}} = \lim_{n \rightarrow \infty} \langle f_n, g_n \rangle_{\mathcal{H}_0} .$$

This is a bilinear form. For the norm, defined as : $\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}}$, we need to verify that :

if $\|f\|_{\mathcal{H}}^2 = 0$ then $f = 0$.

Use the fact that f is a pointwise limit of a Cauchy sequence in \mathcal{H}_0

We have the reproducing property as well. If we want to compute $\langle f, k(\cdot, x) \rangle_{\mathcal{H}}$, using the reproducing property in the pre-hilbert space : we have $\lim_{n \rightarrow \infty} \langle f_n, k(\cdot, x) \rangle = \lim_{n \rightarrow \infty} f_n(x) = f(x)$.

Unicity theorem

Theorem

Let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a positive definite symmetric kernel and \mathcal{H}_k be a RKHS built from k and \mathcal{X} , then \mathcal{H}_k is unique.

Exercise: : prove it.

Feature Space and feature map

Any Hilbert space \mathcal{H} such that there exists $\varphi : \mathcal{X} \rightarrow \mathcal{H}$ with :

$$\forall (x, x') \in \mathcal{X} \times \mathcal{X}, k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$$

is called a feature space associated with k and φ is called a feature map.

Learning with kernels

Machine Learning : a statistical viewpoint

Supervised Learning

Let $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$, a i.i.d. sample drawn from μ a joint probability distribution defined on the random pair (X, Y) : X takes its values in \mathbb{R}^d and Y is real-valued.

Supervised learning problem

Given a loss function $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$, the goal is to find a solution of :

$$\min_{f \in \mathcal{H}} \mathbb{E}_{\mu}[\ell(Y, f(X))], \quad (2)$$

using the help of the training labeled sample \mathcal{S} .

Examples of losses : $\ell(y, f(x)) = 1_{y \neq f(x)}$ for binary classification,
 $\ell(y, f(x)) = (y - f(x))^2$ for regression.

Supervised learning problem

Given a loss function $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$, the goal is to find a solution of :

$$\arg \min_{f \in \mathcal{H}} \mathbb{E}_{\mu}[\ell(Y, f(X))], \quad (3)$$

using the help of a training labeled sample \mathcal{S} .

N.B.1/ We cannot solve exactly this problem so we use instead of the true risk $R(f) := \mathbb{E}[\ell(Y, f(X))]$ the empirical counter part :

$$R_n(f) := \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i)).$$

2/ Moreover, when n is not too large, we avoid overfitting by controlling the complexity of the model f and minimizing the empirical risk.

Machine Learning : a statistical viewpoint

Supervised Learning

Let $\mathcal{S}_n = \{(x_i, y_i)\}_{i=1}^n$, a i.i.d. sample drawn from μ a joint probability distribution defined on (X, Y) : X takes its values in \mathbb{R}^d and Y is real-valued.

Regularized empirical risk minimization

Given a loss function $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$, $\Omega : \mathcal{H} \rightarrow \mathbb{R}^+$, the goal is now to find a solution of :

$$\arg \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, h(x_i)) + \lambda \Omega(h) \quad (4)$$

Variant :

$$\begin{aligned} \arg \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, h(x_i)) \\ \text{s.t. } \Omega(h) \leq C. \end{aligned}$$

Machine Learning : two tasks

Let $\mathcal{S}_n = \{(x_i, y_i)\}_{i=1}^n$, a i.i.d. sample drawn from μ a joint probability distribution defined on (X, Y) : X takes its values in \mathbb{R}^d and Y is real-valued.

- **Learning** : get $h_n = \mathcal{A}(\mathcal{S}_n, \mathcal{H}, \ell, \lambda, \Omega) \in \mathcal{H}$ with
 - \mathcal{S}_n : training data
 - \mathcal{H} : class of functions
 - λ : some hyperparameter
 - ℓ : Local loss function
 - Ω : regularizing function
 - \mathcal{A} : learning algorithm
- **inference (prediction)** : given x , and compute $h_n(x)$

learning from data within a RKHS

Let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a positive definite symmetric kernel and \mathcal{H}_k , its corresponding RKHS.

In the following, we have a we want to solve a supervised learning problem where the **hypothesis space** is a RKHS \mathcal{H}_k .

For that purpose, we are going to make use of a Representer Theorem that says that the solution of our problem takes always the form :

$$f_n(\cdot) = \sum_{i=1}^n \alpha_i k(x_i, \cdot).$$

Representer theorem

Theorem (Wahba, 1978 ; Schoelkopf et al. 2001)

Let $\{x_1, \dots, x_n\}$ be a set of points in \mathcal{X} . Assume λ is a strictly positive scalar. Let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a positive definite symmetric kernel and \mathcal{H}_k , its corresponding RKHS, then, for a strictly monotonically increasing function $g : [0, \infty[\rightarrow \mathbb{R}$ and any cost function $c : (\mathcal{X} \times \mathbb{R})^n \rightarrow \mathbb{R} \cup \{+\infty\}$, any function $f \in \mathcal{H}_k$ minimizing

$$J(f) = c(x_1, \dots, x_n, f(x_1), \dots, f(x_n)) + \lambda g(\|f\|_{\mathcal{H}_k}) \quad (5)$$

admits a solution that is an expansion of the form :

$$f_n(\cdot) = \sum_{i=1}^n \alpha_i k(x_i, \cdot).$$

Proof of the Representer theorem

Let us define : $\mathcal{H}_1 = \text{span} \{k(x_i, \cdot), i = 1, \dots, n\}$

Any $f \in \mathcal{H}$ writes as : $f = f_1 + f_1^\perp$, with $f_1 \in \mathcal{H}_1$ and $f_1^\perp \in \mathcal{H}_1^\perp$
where $\mathcal{H} =$ direct sum of \mathcal{H}_1 and \mathcal{H}_1^\perp .

By the reproducing property, we get :

$$f(x_i) = \langle f_1(\cdot) + f_1^\perp(\cdot), k(x_i, \cdot) \rangle = \langle f_1(\cdot), k(x_i, \cdot) \rangle = f_1(x_i)$$

Hence, $c(f(x_1), \dots, f(x_n)) = c(f_1(x_1), \dots, f_1(x_n))$ By orthogonality,

$$\|f\|^2 = \|f_1\|^2 + \|f_1^\perp\|^2$$

Similarly we have :

$$g(\|f\|) = g(\sqrt{\|f_1\|^2 + \|f_1^\perp\|^2}) \geq g(\|f_1\|)$$

due to the fact that g is strictly monotonic. Equality occurs if and only if $f_1^\perp = 0$.

If f is a minimizer of $J(f)$, then f_1 is also a minimizer of J . if f_1^{perp} was not equal to 0 Moreover if g is strictly increasing, $J(f_1) < J(f)$, then any $f = f_1 + f_1^\perp$ exactly equals to f_1 .

A to-do do list

1. Define a **PDS kernel** : $k(\cdot, \cdot)$ not only as a good similarity between your input data but also define a RKHS, \mathcal{H}_k from k with an appropriate norm $\|\cdot\|_{\mathcal{H}_k}$ that depends on k .
2. Define a **loss functional** with two terms : a local loss function ℓ and a penalty function g
3. Either Prove/use a **representer theorem** to get the form of the minimizer of this functional : $\sum_i \alpha_i k(\cdot, x_i)$ and go to 5, or go to 6
4. Solve the **optimization problem** in \mathbb{R}^n instead of \mathcal{H}_k (practical session), i.e. define a learning algorithm
5. *Alternative solution : define the pb in the primal space and then dualize the primal formulation, and a finite representation is obtained as well*

The example of Kernel Ridge Regression

Machine Learning : a statistical viewpoint

Supervised Learning

Let $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$, a i.i.d. sample drawn from μ a joint probability distribution defined on the random pair (X, Y) : X takes its values in \mathbb{R}^d and Y is real-valued.

Supervised learning problem

Given a loss function $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$, the goal is to find a solution of :

$$\min_{f \in \mathcal{H}} \mathbb{E}_{\mu}[\ell(Y, f(X))], \quad (6)$$

using the help of the training labeled sample \mathcal{S} .

Examples of losses : $\ell(y, f(x)) = 1_{y \neq f(x)}$ for binary classification,
 $\ell(y, f(x)) = (y - f(x))^2$ for regression.

Supervised learning problem

Given a loss function $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$, the goal is to find a solution of :

$$\arg \min_{f \in \mathcal{H}} \mathbb{E}_{\mu}[\ell(Y, f(X))], \quad (7)$$

using the help of a training labeled sample \mathcal{S} .

N.B.1/ We cannot solve exactly this problem so we use instead of the true risk $R(f) := \mathbb{E}[\ell(Y, f(X))]$ the empirical counter part :

$$R_n(f) := \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i)).$$

2/ Moreover, when n is not too large, we avoid overfitting by controlling the complexity of the model f and minimizing the empirical risk.

Machine Learning : a statistical viewpoint

Supervised Learning

Let $\mathcal{S}_n = \{(x_i, y_i)\}_{i=1}^n$, a i.i.d. sample drawn from μ a joint probability distribution defined on (X, Y) : X takes its values in \mathbb{R}^d and Y is real-valued.

Regularized empirical risk minimization

Given a loss function $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$, $\Omega : \mathcal{H} \rightarrow \mathbb{R}^+$, the goal is now to find a solution of :

$$\arg \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, h(x_i)) + \lambda \Omega(h) \quad (8)$$

Variant :

$$\begin{aligned} \arg \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, h(x_i)) \\ \text{s.t. } \Omega(h) \leq C. \end{aligned}$$

Machine Learning : two tasks

Let $\mathcal{S}_n = \{(x_i, y_i)\}_{i=1}^n$, a i.i.d. sample drawn from μ a joint probability distribution defined on (X, Y) : X takes its values in \mathbb{R}^d and Y is real-valued.

- **Learning** : get $h_n = \mathcal{A}(\mathcal{S}_n, \mathcal{H}, \ell, \lambda, \Omega) \in \mathcal{H}$ with
 - \mathcal{S}_n : training data
 - \mathcal{H} : class of functions
 - λ : some hyperparameter
 - ℓ : Local loss function
 - Ω : regularizing function
 - \mathcal{A} : learning algorithm
- **inference (prediction)** : given x , and compute $h_n(x)$

A classic example : linear ridge regression

Comparison of ridge regression with ordinary least squares and lasso.

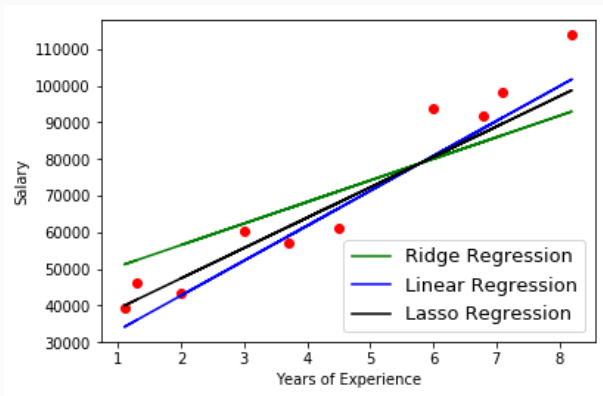


Figure (M. Bajwa).

A classic example : linear ridge regression

Linear Regression in \mathbb{R} with a ℓ_2 regularization/

$$\arg \min_{\beta \in \mathbb{R}^d} L(\beta) := \frac{1}{n} \sum_{i=1}^n (y_i - \beta^T x_i) + \lambda \|\beta\|^2. \quad (9)$$

which also writes as : (with \mathbf{y} the vector of y_i 's)

$$\arg \min_{\beta \in \mathbb{R}^d} \frac{1}{n} (\mathbf{y} - X\beta)^T (\mathbf{y} - X\beta) + \lambda \beta^T \beta \quad (10)$$

First order condition gives :

$$\frac{\partial L(\beta)}{\partial \beta} = 0 \iff \beta_{\text{ridge}} = (X^T X + \lambda n I)^{-1} X^T \mathbf{y}. \quad (11)$$

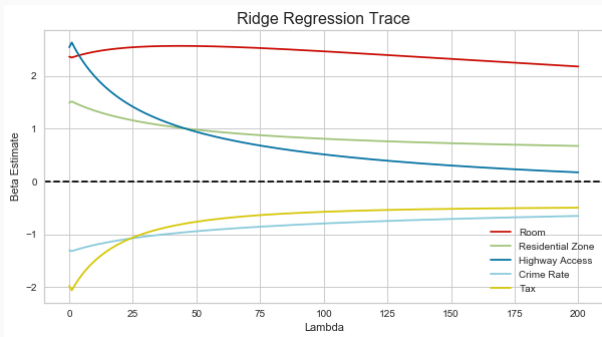
Note that :

$$\beta_{\text{ridge}} = (X^T X + \lambda n I)^{-1} X^T \mathbf{y} = X^T (X X^T + \lambda n I_n)^{-1} \mathbf{y} := X^T \alpha_{\text{ridge}}$$

with $\alpha_{\text{ridge}} := (X X^T + \lambda n I_n)^{-1} \mathbf{y}$.

Effect of lambda

Boston House price prediction in function of 5 features.



Application of kernel principles to ridge regression

Let us solve the kernel ridge regression problem in \mathcal{H}_k :

$$\arg \min_{f \in \mathcal{H}_k} L(f) := \frac{1}{n} \sum_i (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}_k}^2 \quad (12)$$

The representer theorem applies :

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

Denoting K the $n \times n$ Gram matrix, we can write :

$$\begin{aligned} L(\alpha) &= \frac{1}{n} \|Y - K\alpha\|^2 + \lambda \|f\|^2 \\ &= \frac{1}{n} (Y - K\alpha)^T (Y - K\alpha) + \lambda \alpha^T K \alpha, \end{aligned}$$

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

Application to kernel ridge regression

First order conditions :

$$\begin{aligned}\frac{\partial L}{\partial \alpha} &= -\frac{1}{n}(Y - K\alpha)^T K + \lambda \alpha^T K \\ &= -\frac{1}{n}K(Y - K\alpha) + \lambda K\alpha \\ &= -\frac{1}{n}KY + \frac{1}{n}K^2\alpha + \lambda K\alpha\end{aligned}$$

We have : $\frac{\partial L}{\partial \alpha} = 0 \iff K(K\alpha + n\lambda I) = KY$.

$$K((K + n\lambda I)\alpha - Y) = 0$$

Let us note : $z = (K + n\lambda I)\alpha - Y$.

We have 2 ways to search for a solution : $Kz = 0$ means that $z \in \text{Ker}K$.

However let us look first at the case $z = 0$. NB : $(K + \lambda I)$ is invertible if λ is strictly positive

In this case, $z = 0$ implies $\alpha - (K + n\lambda I)^{-1}Y = 0$

Then, $\alpha = (K + n\lambda I)^{-1}Y$ is a solution.

Kernel ridge regression : cont'd

Do we have other solutions with $z \neq 0$?

We take : $\alpha' = \alpha + \epsilon$ with $K\epsilon = 0$.

Now, what if we compare f_α and $f_{\alpha'}$:

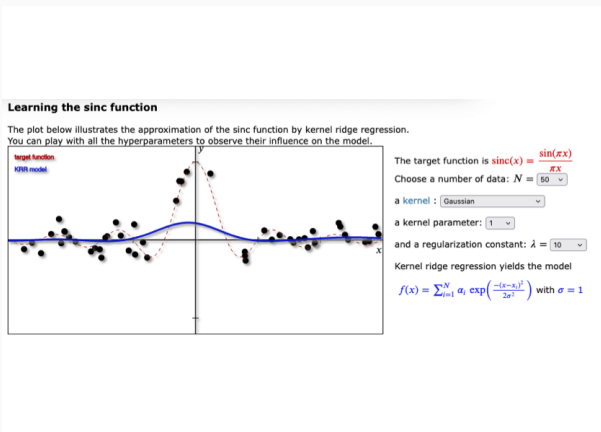
$$\begin{aligned}\|f_{\alpha'} - f_\alpha\|^2 &= (\alpha' - \alpha)^T K (\alpha' - \alpha) \\ &= \epsilon^T K \epsilon \\ &= 0\end{aligned}$$

so the solution is unique and writes as : $\alpha = (K + \lambda I)^{-1} Y$

N.B. In practice we prefer **not to inverse a $n \times n$ matrix**, either we use **low rank approximation** or we use **stochastic gradient descent** algorithm to find the minimum. See practical session.

Kernel ridge regression : the effect of regularization

$$\lambda = 10$$



See illustration here (Fabien Lauer, Loria) :

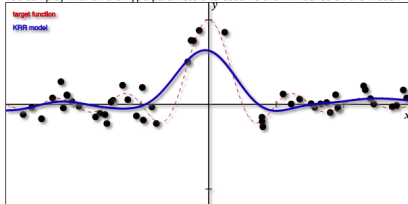
<https://mlweb.loria.fr/book/en/kernelridgeregression.html>.

Kernel ridge regression : the effect of regularization

$$\lambda = 1$$

Learning the sinc function

The plot below illustrates the approximation of the sinc function by kernel ridge regression. You can play with all the hyperparameters to observe their influence on the model.



The target function is $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$

Choose a number of data: $N = 50$

a kernel : Gaussian

a kernel parameter: 1

and a regularization constant: $\lambda = 1$

Kernel ridge regression yields the model

$$f(x) = \sum_{i=1}^N \alpha_i \exp\left(\frac{-(x-x_i)^2}{2\sigma^2}\right) \text{ with } \sigma = 1$$

See illustration here (Fabien Lauer, Loria) :

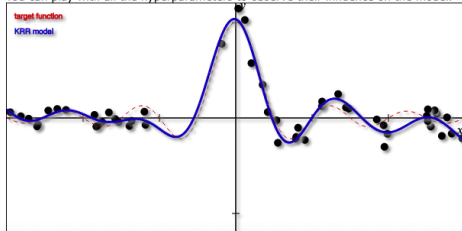
<https://mlweb.loria.fr/book/en/kernelridgeregression.html>.

Kernel ridge regression : the effect of regularization

$$\lambda = 0.01$$

Learning the sinc function

The plot below illustrates the approximation of the sinc function by kernel ridge regression. You can play with all the hyperparameters to observe their influence on the model.



The target function is $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$

Choose a number of data: $N = 50$

a kernel : Gaussian

a kernel parameter: 1

and a regularization constant: $\lambda = 0.01$

Kernel ridge regression yields the model

$$f(x) = \sum_{i=1}^N \alpha_i \exp\left(\frac{-(x-x_i)^2}{2\sigma^2}\right) \text{ with } \sigma = 1$$

See illustration here (Fabien Lauer, Loria) :

<https://mlweb.loria.fr/book/en/kernelridgeregression.html>.

Kernel ridge regression : hyperparameter selection

hyperparameter : λ (and σ if the kernel is chosen to be Gaussian).

- Cross-validation
- Special case leave-one out (LOO)

The **Generalized Cross-validation** estimator was introduced in 1978 (Grace Wahba) as an approximation of the LOO estimator :

$$GCV_{\lambda}(K, y) = n \frac{y^T (K + \lambda I)^{-2} y}{\text{Tr}((K + \lambda I)^{-1})^2}.$$

This estimator has been proven to be uniform consistent and the convergence is non-asymptotic. See very recent work (March 2024) of Misiakiewicz and Saaed (<https://arxiv.org/pdf/2403.08938>).

What if the kernel is linear : $k(x, x') = x^T x'$, i.e. $K = XX^T$. we have :

$$\alpha_{ridge} = (XX^T + n\lambda I)^{-1} Y \quad (13)$$

Revisiting SVM

Binary classification

- Let $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$, a i.i.d. sample drawn from μ a joint probability distribution defined on (X, Y) .
- X takes its values in \mathcal{X} and Y takes its values in $\mathcal{Y} = \{-1, 1\}$
- Choose a PDS kernel k over \mathcal{X} (for instance when $\mathcal{X} = \mathbb{R}^d$, use the Gaussian kernel).
- Define \mathcal{H}_k the RKHS associated to k .
- Consider binary models of the form :

$$f(x) = \text{sign}(h(x)),$$

with $h \in \mathcal{H}_k$.

Hinge loss

Pick up the hinge loss function :

$\ell_{\text{hinge}}(y, h(x)) = \max(1 - yh(x), 0)$,
which is a convex upper bound of the
non-continuous 0-1 loss.

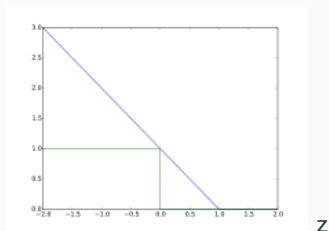


Figure 1 – Green : 0-1 loss as a function of the functional *margin* $z = yh(x)$, and Blue : hinge loss as a function of $z = yh(x)$

Revisiting SVM at the lens of the representer theorem

Regularized empirical risk minimization with hinge loss

Given a loss function $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$, , the goal is now to find a solution of :

$$\arg \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \max(1 - y_i h(x_i), 0) + \lambda \|h\|_{\mathcal{H}_k}^2 \quad (14)$$

Representer theorem applies.

$h(x) = \sum_{i=1}^n \alpha_i k(x, x_i) = \sum_{i=1}^n \alpha_i \varphi(x_i)$. The optimization problem now writes :

$$\arg \min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \max(1 - y_i (K\alpha)_i, 0) + \lambda \alpha^T K \alpha. \quad (15)$$

Finite dimensional optimization problem but still a difficulty remains with max

Revisiting SVM at the lens of the representer theorem

Let us introduce slack variables $\xi_1, \dots, \xi_n \in \mathbb{R}$ s.t. the problem rewrites :

$$\min_{\alpha \in \mathbb{R}^n, \xi \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \xi_i + \lambda \alpha^T K \alpha$$

and $\xi_i \geq 1 - y_i(K\alpha)_i$ and $\xi_i \geq 0$.

Once you get α , you have $h(x)$ from $h(x) = \sum_i \alpha_i k(x, x_i)$.

Solving the problem

We recognize a quadratic optimization problem with affine constraints for which the saddle point theorem applies -

- Define the Lagrangian and the Lagrangian coefficients
- Apply first order optimality condition
- Re-write the Lagrangian as a function of the Lagrangian coefficients
- The optimization problem can be easily solved by a solver (convex optimization with affine constraints)
- Insights on the support vector by applying the whole set of Karush-Kuhn-Tucker conditions

The problem we solved is not exactly the original SVM

Usually we take : $f(x) = \text{sign}(h(x) + b)$, with $b \in \mathbb{R}$.

We need a semi-parametric representer theorem to treat this case.

Semi-parametric representer theorem

Theorem 5 (Semi-parametric Representer Theorem (Schoelkopf et al. 2001))

Suppose that in addition to the conditions of the General Representer Theorem we are given a set of M real-valued functions $\{\psi_p\}_{p=1}^M$ on \mathcal{X} , with the property that the $n \times M$ matrix $(\psi_p(x_i))_{ip}$ has rank M . Then any $\tilde{f} := f + h$, with $f \in \mathcal{H}_k$ and $h \in \text{span}(\psi_p)$, minimizing the regularized risk :

$$c((x_1, x_2, \dots, x_n, \tilde{f}(x_1), \tilde{f}(x_2), \dots, \tilde{f}(x_n))) + g(\|f\|_{\mathcal{H}_k}),$$

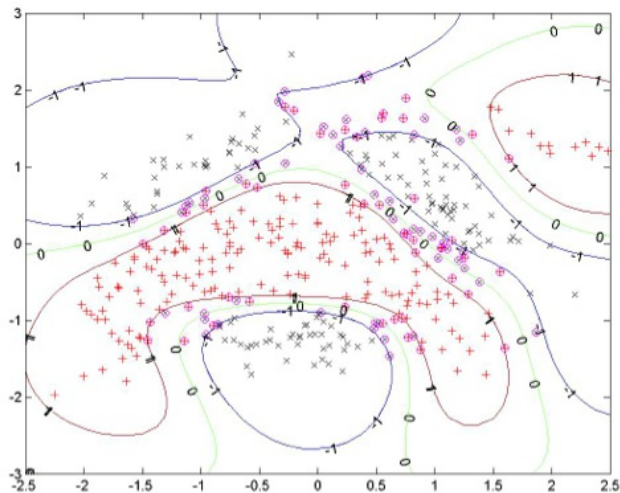
admits a representation of the form :

$$\tilde{f}_n(\cdot) = \sum_{i=1}^n \alpha_i k(\cdot, x_i) + \sum_{p=1}^M \beta_p \psi_p(\cdot),$$

with unique coefficients β_p , $p = 1, \dots, M$.

Proof : the same arguments can be used as $\{\psi_p\}_{p=1}$ is fixed.

Support Vector Machine : nonlinear frontiers in 2D space



References

- Thomas Hofmann, Bernhard Schoelkopf, Alexander J. Smola. "Kernel methods in machine learning." Ann. Statist. 36 (3) 1171 - 1220, June 2008.
<https://doi.org/10.1214/009053607000000677>
- Mehriar Mohri, Afshin Rostamizadeh, Ameet Tlawaakar, Foundations of Machine Learning, The MIT Press, Chapt 5,6 and 11, second edition, 2018.
- Ingo Steinwart, Andreas Christmann, Support Vector Machines, Springer, 2008.
- Misiakiewicz, T., & Saaed, B. (2024). A non-asymptotic theory of Kernel Ridge Regression : deterministic equivalents, test error, and GCV estimator. arXiv preprint arXiv :2403.08938.
- Schoelkopf, B., Herbrich, R., Smola, A. J. (2001, July). A generalized representer theorem. In International conference on computational learning theory (COLT) (pp. 416-426). Berlin, Heidelberg : Springer Berlin Heidelberg.
- B. Schoelkopf, A. J. Smola, Learning with kernels : support vector machines, regularization, optimization, and beyond. MIT press, 2002.
- Grace Wahba, Spline models for observational data, Society for industrial and applied mathematics, 1978.
- G. Wahba, Y. Wang, Representer theorem, Wiley StatsRef : Statistics Reference Online, 1-11, 2019.
- Alexander Wei, Wei Hu, and Jacob Steinhardt, More than a toy : Random matrix models predict how real-world neural representations generalize, arXiv :2203.06176 (2022).