Machine Learning: from Theory to Practice

Lecture 3: Learning in Reproducing Kernel Hilbert Spaces

F. d'Alché-Buc and E. Le Pennec
florence.dalche@telecom-paristech.fr

Fall 2016

Outline

Statistical Learning in a nutshell

- **Learning** $f_n = \mathcal{A}(\mathcal{S}_n, \mathcal{H}, \ell, \lambda)$ with
  - $\mathcal{A}$:learning/estimation/optimization algorithm
  - $\mathcal{S}_n$: training data
  - $\mathcal{H}$: class of functions
  - λ: some hyperparameter
  - ℓ : Local loss function
- **Prediction**: give me a new x, and compute $f_n(x)$

Example: Learning linear models

Linear models

$$f_{lin}(\mathbf{x}) = \beta^T \mathbf{x}$$

Learn β by minimizing:

$$J(\beta) = \sum_{i=1}^{n} (y_i - f_{lin}(x_i))^2 + \lambda \Omega(\beta)$$

Methodology

- Define
  - a representation space for data

Methodology

- Define
  - a representation space for data
  - a class of functions (a class of hypotheses) where to find the solution

Methodology

- Define
  - a representation space for data
  - a class of functions (a class of hypotheses) where to find the solution
  - a loss function to be minimized

Machine Learning

Methodology

- Define
  - a representation space for data
  - a class of functions (a class of hypotheses) where to find the solution
  - a loss function to be minimized
  - an optimization algorithm

Machine Learning

Methodology

- Define
  - a representation space for data
  - a class of functions (a class of hypotheses) where to find the solution
  - a loss function to be minimized
  - an optimization algorithm
  - a model selection method for hyperparameters

Goal:

- Study a general framework for learning (nonlinear) nonparametric functions

Learning in RKHS

- Work on a general class of functions called RKHS: Reproducing Kernel Hilbert Space (this answers to the Representation problem)
- Exhibit different loss functions that allows to solve various ML tasks
- Other properties we can get easily from working in RKHS: easier analysis of generalization bounds, consistency properties

Working in RKHS is as simple as working with linear models

Linear models

$$f_{lin}(\mathbf{x}) = \beta^T \mathbf{x}$$

Learn β by minimizing:

$$J(\beta) = \sum_{i=1}^{n} L(\mathbf{x}_i, y_i, f_{lin}(x_i)) + \lambda \Omega(\beta)$$

Working in RKHS is as simple as working with linear models

RKHS models

k positive definite and $\mathcal{H}_k$ the RKHS associated, $x_1, \ldots, x_n$. When a **representer theorem** applies:

$$f_{rep}(x) = \alpha^T k_x = \sum_{i=1}^{n} \alpha_i k(x, x_i),$$

with $k_x^T = [k(x, x_1), \ldots k(x, x_n)]$
Learn α by minimizing

$$J(\alpha) = \sum_{i=1}^{n} L(\mathbf{x}_i, y_i, f_\alpha(x_i)) + \lambda \Omega(f_\alpha)$$

Pb 1: predict the property of a molecule

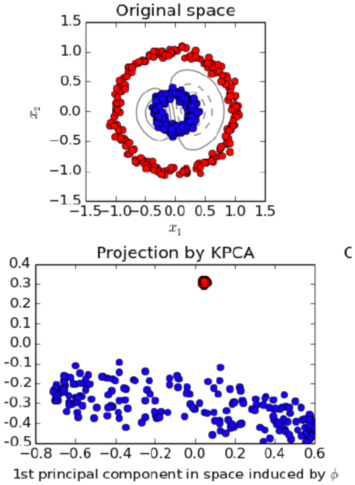
A supervised learning problem

- **Inputs** : molecule (drug candidate)
- **Output** : activity on a cancer line (or several cancer lines)

A regression problem from structured data.

Pb 2: dimension reduction

An unsupervised learning problem



Find a new data representation in a smaller dimension space

Outline

Outline

Outline

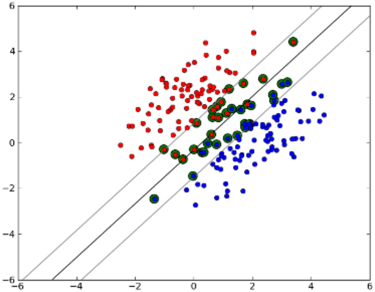
Minimizing a convex loss for all except for some outliers

Examples

- Example 1: Support Vector Machine (reminder), maximize the margin for all except a few training data points
- Example 2: Support Vector Regression, minimize the ϵ-insensitive for all except a few training data points

Example 1 : Linear SVM in $\mathbb{R}^p$

Input set: $\mathcal{X}$
Output set : $\{-1, +1\}$
$\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$



Example 1 : Linear SVM in $\mathbb{R}^p$

Maximizing the soft margin:

Solving the problem in the primal space

$$\min_{w,b,\xi} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{under the constraints} \quad y_i(w^T x_i + \mathrm{b}) \geq 1 - \xi_i \; i = 1, \ldots, n.$$

$$\xi_i \geq 0 \; i = 1, \ldots, n.$$

ξ_i: slack variable for each training data

Reference

Boser, B. E.; Guyon, I. M.; Vapnik, V. N. (1992). "A training algorithm for optimal margin classifiers". Proceedings of the fifth annual workshop on Computational learning theory - COLT '92. p. 144.

Optimization problem for SVM

Solving the pb in the dual

$$\max_{\alpha} \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j {x_i}^T x_j$$

$$\text{under the constraints} \quad 0 \leq \alpha_i \leq C \;\; i = 1, \ldots, n.$$

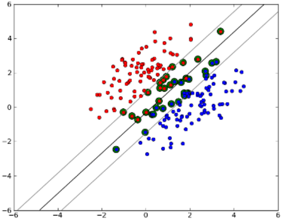
$$\sum_i \alpha_i y_i = 0 \;\; i = 1, \ldots, n.$$

Solution : Support Vector Machine

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i x^T x_i + b$$

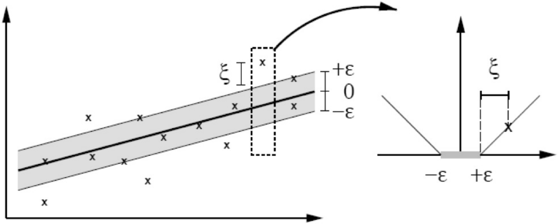
$$h_{SVM}(x) = \text{sign}(f(x))$$

The x_i such that $\alpha_i > 0$ are the so-called *support vectors*.



Support Vector Regression

- Extend the idea of maximal soft margin to regression: training data should be in the tube while the tube should be flat
- Impose an ϵ-tube : ϵ-sensitive loss , no penalty occurs if $\|y_i - f(x_i)\| \leq \epsilon$.
  $\ell_\epsilon(x, y, f(x)) = |y - f(x)|_\epsilon = max(0, |y - f(x)| - \epsilon)$



Support Vector Regression

SVR in the primal space

Given C and ϵ

$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_i(\xi_i + \xi_i^*)$

s.c.

$\forall i = 1, \ldots n, y_i - f(x_i) \leq \epsilon - \xi_i$

$\forall i = 1, \ldots n, f(x_i) - y_i \leq \epsilon - \xi_i^*$

$\forall i = 1, \xi_i \geq 0, \xi_i^* \geq 0$

with $f(x) = w^T\phi(x) + b$

Reference

Drucker H. Burfges, C. Kaufman L., Smola, A. V. Vapnik (1997). Support Vector Regression - NIPS'97. p. 144.

Solution in the dual

$$\min_{\alpha,\alpha^*} \sum_{i,j}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*){x_i}^T x_j + \epsilon \sum_i (\alpha_i + \alpha_i^*) - \sum_i y_i(\alpha_i - \alpha_i^*)$$

s.c. $\sum_i(\alpha_i - \alpha_i^*) = 0$ and $0 \leq \alpha_i \leq C$ and $0 \leq \alpha_i^* \leq C$

$w = \sum_{i=1}^n (\alpha_i - \alpha_i^*) x_i$

Solution

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*){x_i}^T x + b$$

Observe what is common to the two approaches

- A convex loss
- Notions of tube and geometric margin
- Insensitivity to some low errors: NB: could be useful for other algorithms as well
- Minimizing a term of complexity $||w||^2$ NB: minimize the Structural Risk and NOT the empirical risk
- Dual solution opens the door to the kernel trick

Kernelization of SVM and SVR

In the dual formulation, we notice : Each time the training data appear in the objective dual function, they appear as dot product:

- SVM : $\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j {x_i}^T x_j$
- SVR :
  $\sum_{i,j} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) {x_i}^T x_j + \epsilon \sum_i (\alpha_i + \alpha_i^*) - \sum_i y_i (\alpha_i - \alpha_i^*)$

Idea (credit: Isabelle Guyon):

- We just need to compute scalar product during the *learning phase* as well as the *prediction phase*
- Whatever the space / set (called $\mathcal{X}$) I am working in, if I had a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that k computes inner products.

Does there exist such functions ?

Definition of Positive Definite Symmetric **kernel**, PDS kernels

Let $\mathcal{X}$ be a non-empty set. Let k:$\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric function. k is a positive definite kernel *if and only if* for any finite set $\{x_1, \ldots, x_m\}$ de $\mathcal{X}$ and the column vector c of $\mathbb{R}^m$,

$$c^T K c = \sum_{i,j} c_i c_j k(x_i, x_j) \geq 0$$

Be careful: each matrix needs to be semi-definite positive while we call the kernel Positive definite (improperly)

Kernel properties

A simplified version of Moore-Aronzajn theorem, 1950

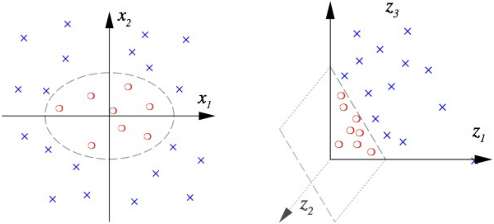
Let k:$\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a PDS kernel then there exists a Hilbert space $\mathcal{H}$, called *feature space* and a *feature map*: $\phi : \mathcal{X} \rightarrow \mathcal{H}$ such that: $k(x, x') =< \phi(x), \phi(x') >_{\mathcal{H}}$,
where $< \cdot, \cdot >_{\mathcal{H}}$ is the dot product associated with $\mathcal{H}$.

We will come back in a few slides to the constructive proof of the full theorem and the Reproducing Kernel Hilbert Space Theory.

- There always exists at least one feature map and one feature space such that: $\phi(x) = k(\cdot, x)$
- Givne a kernel, the pairs (feature map, feature space) are not unique !

Example: Polynomial kernel

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$



Example: Polynomial kernel

Kernel trick

Notice that $\phi(\mathbf{x}_1)^T\phi(\mathbf{x}')$ cna be computed without working directly in $\mathbb{R}^3$

We know how to compute k without needing this specific feature map ϕ: $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}') = (\mathbf{x}^T\mathbf{x}')^2$

Some other kernels on vectors

$\mathcal{X} = \mathbb{R}^p$

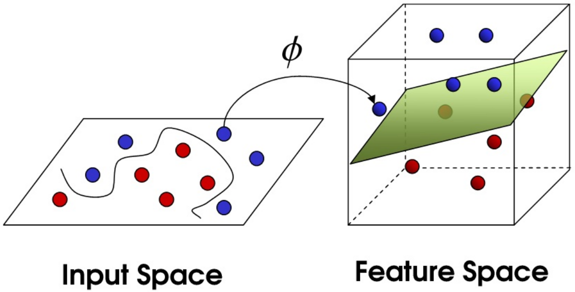
- Linear kernels: $k(x, x')$
- **Gaussian kernels**: $k(x, x') = \exp(-\gamma ||x - x'||^2)$ (no finite dimensional feature map)
- Polynomial kernels: $k(x, x') = (x^T x' + c)^d$ (there exists a finite dimensional feature map)
- Sigmoidal kernels: $k(x, x') = \tanh(a x^T x' + b)$

Back to the kernelization of SVM and SVR

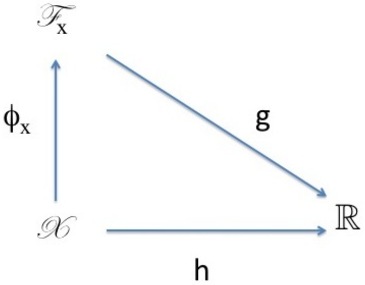
In the dual formulation, we replace ${x_i}^T x_j$ by $k(x_i, x_j)$ where k is a Positive Definite Symmetric kernel

- SVM : $\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$
- SVR : $\sum_{i,j} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(x_i, x_j) + \epsilon \sum_i (\alpha_i + \alpha_i^*) - \sum_i y_i (\alpha_i - \alpha_i^*)$

Kernel trick and feature map 1/2



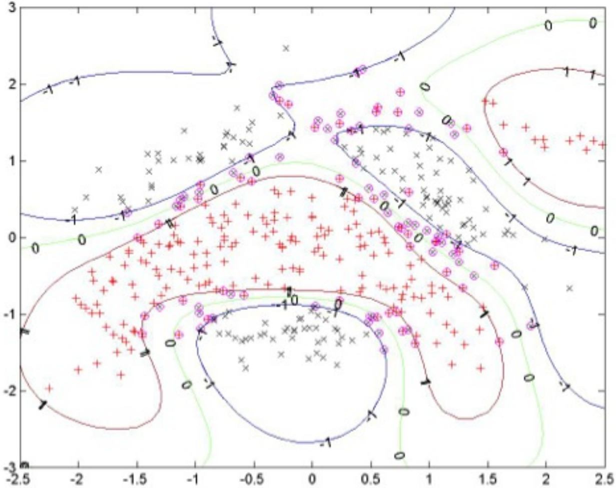
Kernel trick and feature map 2/2



Cas of SVM:

- $f(x) = \sum_{i=1}^{n} \alpha_i y_i < \phi(x), \phi(x_i) >_{\mathcal{F}} = \sum_{i=1}^{n} \alpha_i y_i k(x, x_i)$,
- $g(z) = (\sum_i \alpha_i \phi(x_i))^T z$
- $f(x) = g \circ \phi(x)$
- SVM : $h(x) = \text{sign}(f(x) + b)$

Non linear SVM : on simulated data



Outline

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) $x^T Ay$ for $A \succeq 0$ (i.e. psd)	$\Phi(x) = L^T x$ for $A = LL^T$ (Cholesky)

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

Much more interesting: kernels for complex objects

Kernels for

- **Complex (unstructured) objects**: texts, images, documents, signal, biological objects (gene, mRNA,protein, ...), functions, histograms
- **Structured objects**: sequences, trees, graphs, any composite objects

This made the success of kernels in computational biology, information retrieval (categorization for instance), but also in unexpected areas such as software metrics ....

Example: predict the property of a molecule

- **Inputs** : molecule (drug candidate)
- **Output** : activity on a cancer line (or several cancer lines)

A regression problem from structured data.

Kernel for labeled graphs

For a given length L, let us first enumerate all the paths of length $\ell \leq L$ in the training dataset (data are molecule = labeled graphs). Let m be the size of this (huge) set. For a graph, define $\phi(G) = (\phi_1(G), \ldots, \phi_m(G))^T$ where $\phi_m(T)$ is 1 if the m^{th} path appears in the labeled graph G, and 0 otherwise.

Kernel for labeled graphs

Definition 1:

$$k_L(G, G') =< \phi(G), \phi(G') >$$

Tanimoto kernel

$$k_L^t(G, G') = \frac{k_m(G, G')}{k_m(G, G) + k_m(G', G') - k_m(G, G')}$$

idea: k_m^t calculates the ratio between the number of elements of the intersection of the two sets of paths (G and G' are seen as bags of paths) and the number of elements of the union of the two sets.
Reference: Ralaivola et al. 2005, Su et al. 2011

Convolution kernels

Definition:
Suppose that $x \in \mathcal{X}$ is a **composite structure** and $x_1, \ldots, x_D$ are its "parts" according a relation R such that $(R(x, x_1, x_2, \ldots, x_D)$ is true, with $x_d \in \mathcal{X}_d$ for each $1 \leq d \leq D$, D being a positive integer. k_d be a PDS kernel on a set $\mathcal{X} \times \mathcal{X}$, for all (x,x'), we define:

$$k_{conv}(x, x') = \sum_{(x_1,\ldots,x_d)\in R^{-1}(x),(x'_1,\ldots,x'_d)\in R^{-1}(x')} \prod_{d=1}^{D} k_d(x_d, x'_d)$$

$R^{-1}(x) =$ all decompositions $(x_1, \ldots, x_D)$ such that $(R(x, x_1, x_2, \ldots, x_D)$. k_{conv} is a PDS kernel as well. Intuitive kernel, used as a building principle for a lot of other kernels. Next, we will see two examples.

Kernel between vertices in a graph

Let $x_1, \dots, x_n$, n objects associated with a non oriented graph of size n and adjacency matrix W. Define the graph Laplacian :
$L = D - W$, D is the diagonal matrix of degrees

$$K = \exp(-\lambda L)$$

We will see applications of this kernel in the unsupervised course.
Reference: Kondor and Lafferty, 2003

Fisher kernel

Combine the advantages of graphical models and discriminative methods

Let $\mathbf{x} \in \mathbb{R}^p$ be the input vector of a classifier.

- Learn a generative model $p_\theta(\mathbf{x})$ from unlabeled data $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- Define the Fisher vector as : $\mathbf{u}_\theta(\mathbf{x}) = \nabla_\theta \log p_\theta(\mathbf{x})$
- Estimate the Fisher Information matrix of p_θ: $F_\theta = \mathbb{E}_{\mathbf{x} \sim p_\theta}[\mathbf{u}_\theta(\mathbf{x})\mathbf{u}_\theta(\mathbf{x})^T]$
- **Definition**: $k_{Fisher}(\mathbf{x}, \mathbf{x}') = \mathbf{u}_\theta(\mathbf{x})^T F_\theta \mathbf{u}_\theta(\mathbf{x})$

Applications

Classification of secondary structure of proteins, topic modeling in documents, image classification and object recognition, audio signal classification ...

Kernel design

- Use closure properties to build new kernels from existing ones
- Kernels can be defined for various objects:
  - **Structured objects**: (sets), graphs, trees, sequences, ...
  - Unstructured data with underlying structure: texts, images, documents, signal, biological objects (gene, mRNA,protein, ...)
- **Kernel learning**:
  - Hyperparameter learning: see Chapelle et al. 2002
  - Multiple Kernel Learning: given $k_1, \ldots, k_m$, learn a convex combination $\sum_i \beta_i k_i$ of kernels (SimpleMKL Rakotomamonjy et al. 2008, unifying view in Kloft et al. 2010)

Outline

Definition

Definition (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 $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a **reproducing kernel** of $\mathcal{H}$, and $\mathcal{H}$ is a reproducing kernel Hilbert space if:

- $\forall x \in \mathcal{X}, k(\cdot, x) \in \mathcal{X}$
- $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}, \langle f(\cdot), k(\cdot, x)\rangle_{\mathcal{H}} = f(x)$ (**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 (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 $\phi : \mathcal{X} \rightarrow \mathcal{H}$ such that:

$$\forall (x, x') \in \mathcal{X} \times \mathcal{X}, k(x, x') =< \phi(x), \phi(x') >_{\mathcal{H}}$$

Furthermore, $\mathcal{H}$ has the following reproducing property:

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

Constructive Proof 1/4

Let us define $\mathcal{H}_0 = \text{span}\{\sum_{i \in I} \alpha_i k(\cdot, x_i), x_i \in \mathcal{X}, |I| < \infty\}$.
$\mathcal{H}_0$ is the set of finite linear combinations of functions $x \to k(\cdot, x_i)$.
Introduce the operation $<, \cdot, \cdot >_{\mathcal{H}_0}$:

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

by

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

We notice that:

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

meaning that this product between f and g does not depend on the expansions of f or g. This last equation also shows that this product is bilinear. It is also trivially symmetric. $< \cdot, \cdot >_{\mathcal{H}_0}$ is a dot product on functions of $\mathcal{H}_0$

We define a norm from this dot product:

$$\|f\|_{\mathcal{H}_0}^2 =< f, f >_{\mathcal{H}_0}= \sum_{i \in I, j \in I} \alpha_i^T K \alpha_j$$

where K is the Gram matrix associated to k.
Remark: we have a Cauchy-Schwartz inequality for PDS kernels (that we will use).
Proposition: 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 $\det(K) \geq 0$.

Constructive Proof 3/4

We need to prove that we have the reproducing property:

$$\begin{aligned} < f, k(\cdot, x) >_{\mathcal{H}_0} &= < \sum_i \alpha_i k(\cdot, x_i), k(\cdot, x_i) > \\ &= \sum_i \alpha_i k(x, x_i) \\ &= f(x) \end{aligned}$$

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 $\mathcal{H} = \mathcal{H}_0 \cup \{\lim \text{of Cauchy sequences from} \mathcal{H}_0\}$.

Let us call $f = \lim_{n \to \infty} f_n$.

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

Constructive Proof 4/4

Proof of pointwise convergence of $(f_n(x))_n$ for $x \in \mathcal{X}$
$\forall x \in \mathcal{X}, \forall (p, q) \in \mathbb{N}^2,$

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

Then it comes that $(f_n(x))_n$ is a Cauchy Sequence in $\mathbb{R}$ and thus has a limit.
now $f(x) = \lim_{n\to\infty} f_n(x)$.
We want to compute $< \lim f_n, k(\cdot, x) >$. Let us first compute:
$\lim_{n\to\infty} < f_n, k(\cdot, x) >= \lim f_n(x) = f(x)$.
We now define the dot product between a limit of Cauchy Sequence and the function $k(\cdot, x)$ from $\mathcal{H}_0$ as: $< lim f_n, k(\cdot, x) >:= \lim f_n(x) = f(x)$. The dot product can be also defined between two limits of Cauchy sequences and also benefit from the reproducing property.

Unicity theorem

Theorem

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

Feature Space and feature map

Any Hilbert space $\mathcal{H}$ such that there exists $\phi : \mathcal{X} \to \mathcal{H}$ with:
$\forall (x, x') \in \mathcal{X} \times \mathcal{X}, k(x, x') =< \phi(x), \phi(x') >_{\mathcal{H}}$
is called a feature space associated with k and ϕ is called a feature map.

Representer theorem

Theorem

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite symmetric kernel and $\mathcal{H}_k$, its corresponding RKHS, then, for any non-decreasing function $\Omega : \mathbb{R} \to \mathbb{R}$ and any loss function L: $\mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, any minimizer of :

$$J(f) = L(f(x_1), \ldots, f(x_n)) + \lambda \Omega(\|f\|^2_{\mathcal{H}}) \tag{1}$$

admits an expansion of the form:

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

Moreover if Ω is strictly increasing, then any minimizer of 1 has exactly this form.

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: $f = f_1 + f^{\perp}$, with $f_1 \in \mathcal{H}_1$ and $f^{\perp} \in \mathcal{H}_1^{\perp}$
where $\mathcal{H}$= direct sum of$\mathcal{H}_1$ and $\mathcal{H}_1^{+}$.
By orthogonality, $\|f\|^2 = \|f_1\|^2 + \left\|f_1^{\perp}\right\|^2$
Hence, by property of Ω,
$\Omega(\|f\|^2) = \Omega(\|f_1\|^2) + \Omega(\left\|f_1^{\perp}\right\|^2) \geq \Omega(\|f_1\|^2)$
By the reproducing property, we get:
$f(x_i) =< f_1(\cdot) + f_1^{\perp}(\cdot), k(x_i, \cdot) >=< f_1(\cdot), k(x_i, \cdot) >= f_1(x_i)$
Hence, $L(f(x_1), \dots, f(x_n)) = L(f_1(x_1), \dots, f_1(x_n))$ and
$J(f_1) \leq J(f)$
To recap, if f is a minimizer of $J(f)$, then f_1 is also a minimizer of J. Moreover if Ω is stricly 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: $k(\cdot, \cdot)$
2. Define a RKHS, $\mathcal{H}$ from k with an appropriate norm $||\cdot||_{\mathcal{H}}$
3. Define a loss functional with two terms: a local loss function ℓ and a penalty function Ω
4. Prove/use a representer theorem to get the form of the minimizer of this functional: $\sum_i \alpha_i k(\cdot, x_i)$
5. Solve the optimization problem with this minimizer

Outline

Application to kernel ridge regression

- $L(f(x_1), \ldots, f(x_n)) = \sum_i (y_i - f(x_i))^2$ and $\Omega(||f||) = ||f||^2$

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

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

First order conditions:

$$\frac{\partial L}{\partial \alpha} = -(Y - K\alpha)^T K + \lambda\alpha^T K$$

$$\begin{aligned} &= -K(Y - K\alpha) + \lambda K\alpha \\ &= -KY + K^2\alpha + \lambda K\alpha \end{aligned}$$

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

Kernel ridge regression

$$
\begin{aligned}
& K\left((K+\lambda I)\alpha - Y\right) = 0 \\
\iff & \left((K+\lambda I)\alpha - Y\right) \in \text{Ker } K
\end{aligned}
$$

NB: $(K+\lambda I)$ is invertible if λ is positive

Therefore, (2) $\iff \alpha - (K+\lambda I)^{-1}Y \in \text{Ker } K$

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

As well as any $\alpha' = \alpha + \epsilon$ with $K\epsilon = 0$.

Now, 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 writes as:

$$\alpha = (K+\lambda I)^{-1}Y$$

Note that in practise we prefer not to inverse a $n \times n$ matrix and use a stochastic gradient descent algorithm to find the minimum.

Application to the hinge loss

- SVM without bias b
- $L(f(x_1), \ldots, f(x_n)) = max(0, 1 - y_i f(x_i))$ (hinge loss) and $\Omega(\|f\|) = ||f||^2$
- $\min_\alpha \sum_{i=1}^n \max(0, 1 - y_i \sum_j \alpha_j k(x_i, x_j)) + \lambda \alpha^T K \alpha$
  - NB: If you want to introduce b, you need the refer to the semi-parametric representer theorem.

Example: predict the property of a molecule

- **Inputs** : molecule (drug candidate)
- **Output** : activity on a cancer line (or several cancer lines)

A regression problem from structured data.

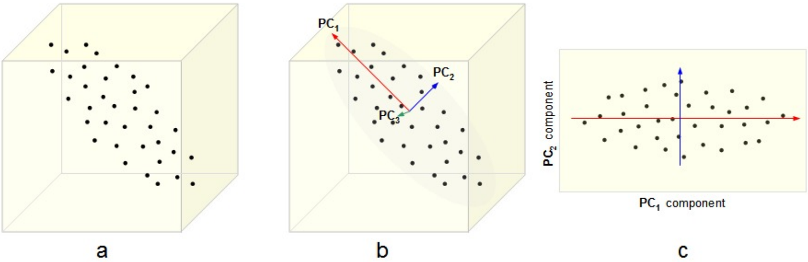
To solve the molecular property pb

- $\mathcal{S}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- Each x_i is a labeled graph, each y_i is a scalar
- Assume we have defined a kernel over labeled graphs
- Different loss functions for different methods
  1. $\arg\min_{f \in \mathcal{F}} \frac{1}{2} \sum_{i=1}^{n} ||y_i - f(x_i)||^2 + \lambda ||f||^2_{\mathcal{H}}$: KRR
  2. $\arg\min_{f \in \mathcal{F}} \frac{1}{2} \sum_{i=1}^{n} max(0, |y_i - f(x_i)|_\epsilon + \lambda ||f||^2_{\mathcal{H}}$: SVR

  See exercise in Datalab 1.

Outline

Principal component analysis



What for ?

- dimension reduction
- denoising

Principal Component Analysis

$Y = X - 1g^T$, centered data with g the center of gravity.

$$\max_{\mathbf{v}\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n}(y_i^T\mathbf{v})^2$$
$$s.t. \|\mathbf{v}\|^2 = 1$$

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
- Intuition: notice that $f(x_i) =< f, \phi(x_i) >_{\mathcal{H}_k}$
- We assume that: $\sum_{i=1}^{n} \phi(x_i) = 0$

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

$$\max_{f \in \mathcal{H}_k} \sum_{i=1}^{n} f(x_i)^2$$

$$s.t. \, \|f\|_{\mathcal{H}_k}^2 = 1$$

Representer theorem applies for Kernel PCA

We have to solve:

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

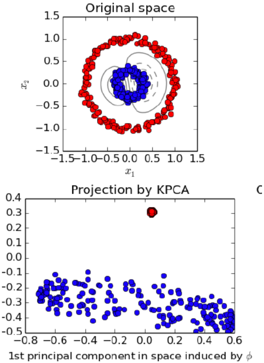
Any solution admits an expansion: $f(x) = \sum_i \alpha_i k(x, x_i)$
Now the problem writes as:

$$\min_{\alpha \in \mathbb{R}^n} -(K\alpha)^T (K\alpha) + \lambda(\|f\|^2_{\mathcal{H}_k} - 1)$$

Any minimum α^* satisfies $K\alpha^* = \lambda\alpha^*$
We find n eigenvectors of K that we re-order by decreasing order using the corresponding eigenvalues.

Compute the projection of a new data on the first component

$$\mathbf{v}_1^T \phi(x) = \sum_i \alpha_i^1 k(x_i, x)$$



Outline

References

- A tutorial review of RKHS, Hoffman, Scholpkoft,Smola, 2005 (first part).
- Foundations of Machine Learning, Mohri, MIT Press, 2012.