Advanced Machine Learning
Lecture 2
Learning with kernel methods

F. d'Alché-Buc
email: florence.dalche@telecom-paristech.fr

Fall 2015

Outline

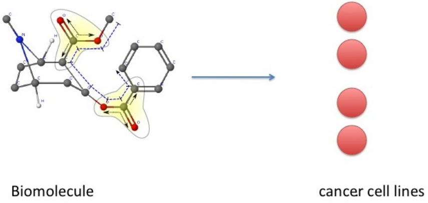
1. Learning in RKHS

2. Kernel design

3. Exercises

Learning in RKHS

Motivation : 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.

Learning in RKHS

A to-do do list

1. Define a PDS kernel : $k(\cdot, \cdot)$
2. Define a (unique) 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 : $\sum_i \alpha_i h(\cdot, x_i)$ **We learn this today**
5. Solve the optimization problem with this minimizer **we do that for kernel ridge regression today**

Learning in RKHS

To solve the molecular property pb

- $S_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- Assume we have defined a kernel over labeled graphs k ad its associated RKHS $\mathcal{F}$
- Choose and minimize the loss function :
    - $\arg\min_{f \in \mathcal{F}} \frac{1}{2} \sum_{i=1}^{n} ||y_i - f(x_i)||^2 + \lambda ||f||_{\mathcal{F}}^2$

Learning in RKHS

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 $\mathrm{L} : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, any minimizer of :

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

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.

Learning in RKHS

Proof of the Representer theorem

Let us define : $\mathcal{H}_1 = \text{span }\{k(x_i, \cdot), i = 1, \ldots, n\}$
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, $\|f\|^2 = \|f_1\|^2 + \|f_1^\perp\|^2$
Hence, by property of Ω,
$\Omega(\|f\|^2) = \Omega(\|f_1\|^2) + \Omega(\|f_1^\perp\|^2) \geq \Omega(\|f_1\|^2)$
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, $L(f(x_1), \ldots, f(x_n)) = L(f_1(x_1), \ldots, f_1(x_n))$ 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.

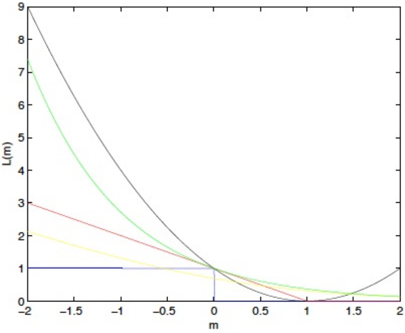
Learning in RKHS

Applications

- $L(f(x_1), \ldots, f(x_n)) = \sum_i (y_i - h(x_i))^2$ and $\Omega(||f||) = ||f||^2$
  - Kernel ridge regression : $\hat{\alpha} = (K + \lambda Id)^{-1}\mathbf{y}$ (proved during course)
- SVM without bias b
- $L(f(x_1), \ldots, f(x_n)) = max(0, 1 - y_i f(x_i))$ (hinge loss) and $\Omega(||f||) = ||f||^2$
  - If you want to introduce b, you need the refer to the semi-parametric representer theorem.

Learning in RKHS

Losses



Various convexifications of the $0 - 1$ loss, including the hinge loss.

Learning in RKHS

Solving our problem with labeled graphs (molecule) in the input space

A molecule x is seen as a labeled graph defined on an alphabet $\mathcal{A}$, a finite set of symbols.
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 set (m is the sum of the number of all possible combinations of $\ell \leq L$ symbols of $\mathcal{A}$). Let us build a dictionary of possible paths of length less or equal to L.

Learning in RKHS
Kernel for labeled graphs

For a graph x, define the feature map :
$\varphi(x) = (\varphi_1(x), \ldots, \varphi_m(x))^T$ where $\varphi_i(x)$ is 1 if the i^{th} path appears in the labeled graph G, and 0 otherwise. **Definition 1** :

$$k_L(x, x') =< \varphi(x), \varphi(x') >$$

Tanimoto kernel

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

idea : k_L^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

Learning in RKHS

Solution for Kernel Ridge regression

Let $K_L = (k_L(x_i, x_{j\,ij})$ be the Gram matrix of our training molecules $\{x_1, \ldots, x_n\}$. Let $\mathbf{y}$ the vector of n dimensions with the target values. Using the representer theorem for the ridge loss, and using the proof on the blackboard we get :

$$\hat{\alpha} = (K + \lambda Id)^{-1}\mathbf{y}$$

We are now able to predict for a new molecule its score with respect with a given cancer (cell line) by using :

$$f(\cdot) = \sum_{i=1}^{n} \hat{\alpha}_i k_L(x, x_i)$$

Learning in RKHS

Outline

1. Learning in RKHS

2. Kernel design

3. Exercises

- 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 $k_1, \dots, k_m$, learn a convex combination $\sum_i \beta_i k_i$ of kernels (see SimpleMKL Rakotomamonjy et al. 2008, unifying view in Kloft et al. 2010)

Kernel design

Kernel closure properties : a few rules

- The sum of two kernels is a kernel
- The product of two kernels is a kernel
- $k(x, x') = \exp(-\gamma D(x, x')^2)$ is a kernel if $D : \mathcal{X} \times \to \mathbb{R}^+$ is a distance.

Exercise : prove it....

Kernel design

Examples of kernels for non-vectorial data and applications

- Convolution kernels
- Graph kernel
- Fisher kernels
- Sequence kernels

Kernel design

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 design

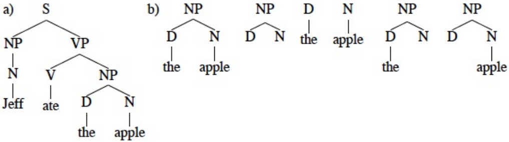
Kernels for Natural Language Processing

Learning task :

- **Input** : sentence → syntax tree
- **Output** : question class
- For instance, in economical news articles, classes are ORGANIZATION, LOCATION,

Kernel design

Kernel for Natural Language Processing



Let us first enumerate all tree fragments that occur in the training data. Let m be the size of this set. For a tree, define $\varphi(T) = (\varphi_1(T), \ldots, \varphi_m(T))^T$ where $\varphi_i(T)$ is the number of occurrences of the i^{th} subtree.

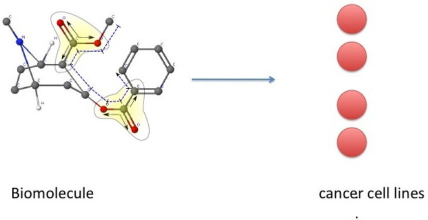
Definition :

$$k_{conv}(T, T') = k(\varphi(T), \varphi(T'))$$

NB : the kernel can be normalized. In NLP, k is often chosen as the linear kernel. Efficient implementations are available.
Sequences can be processed in the same way.
References : Collins and Duffy, 2001 ; Suzuki et al. 2003

Kernel design
Kernel for labeled graphs

Motivation : 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 design

Kernel for labeled graphs

A molecule is seen as a labeled graph defined on an alphabet $\mathcal{A}$, a finite set of symbols.
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 set (m is the sum of the number of all possible combinations of $\ell \leq L$ symbols of $\mathcal{A}$). Let us build a dictionary of possible paths of length less or equal to L.

Kernel design

Kernel for labeled graphs

For a graph G, define the feature map : $\varphi(G) = (\varphi_1(G), \ldots, \varphi_m(T))^T$ where $\varphi_i(T)$ is 1 if the i^{th} path appears in the labeled graph G, and 0 otherwise. **Definition 1** :

$$k_m(G, G') =< \varphi(G), \varphi(G') >$$

Tanimoto kernel

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

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

Kernel design

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

Kernel design

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

Outline

1. Learning in RKHS

2. Kernel design

3. Exercises

Exercises

Exercise 1

We would like to use an appropriate theorem for learning classification function of the following form :

$$f(x) = \sum_{i} \alpha_i k(x, x_i) + b \tag{2}$$

whose sign will give the class. For that purpose, the RKHS associated to a given PDS kernel k is not exactly the functional space we want to work in. We would like to learn two functions : $f \in$ the RKHS $\mathcal{F}_k$ and g a constant function such that $f(x) = h(x) + g(x)$

- Write a Representer theorem for semi-parametric models for models of the form $f(\cdot) + \sum_{\beta} {}_{j}\psi_j(\cdot)$ where the family of functions $(\psi_j(\cdot))_{j=1,\ldots,m}$ is given.
- Apply it with the hinge loss : $L(f(x_1), \ldots, f(x_n)) = \sum_{i=1}^{n} max(0, 1 - y_i f(x_i))$ and $Omega(z) = z^2$

Exercises

Exercise 2

- Prove that $k(x, y) = \exp(-\gamma D(x, y)^2)$ is a kernel for D a distance function over a set $\mathcal{X} \times \mathcal{X}$
- Prove other simple rules of construction of kernels (product, sum, ...)

- Foundations of Machine Learning, Morhi, Rostamizadeh, Talwalkar, MIT Press, 2012.
- A tutorial review in RKHS methods in Machine Learning, Hofman, Schoelpkof, Smola. (online pdf)
- Papers
  - Convolution kernels on discrete structure, D. Haussler, UCSC-CRL-99 (public technical report)
  - Convolution kernels for natural language, Collins and Duffy, NIPS 2001
  - Graph Kernels for chemical informatics, Ralaivola et al. 2005.Preprint Elsevier.
  - Structured output prediction of anti-cancer drug activity, Su et al. 2010, PRB 2010, online pdf.
  - Improving Fisher kernel for large scale image Classification, Perronin et al. 2010, PSM2010,online pdf.
  - Rademacher and gaussian complexities : risk bounds and structural results, Bartlett and Mendelson, JMLR 3(2002), online