Machine Learning with Kernel Methods

Julien Mairal (Inria)

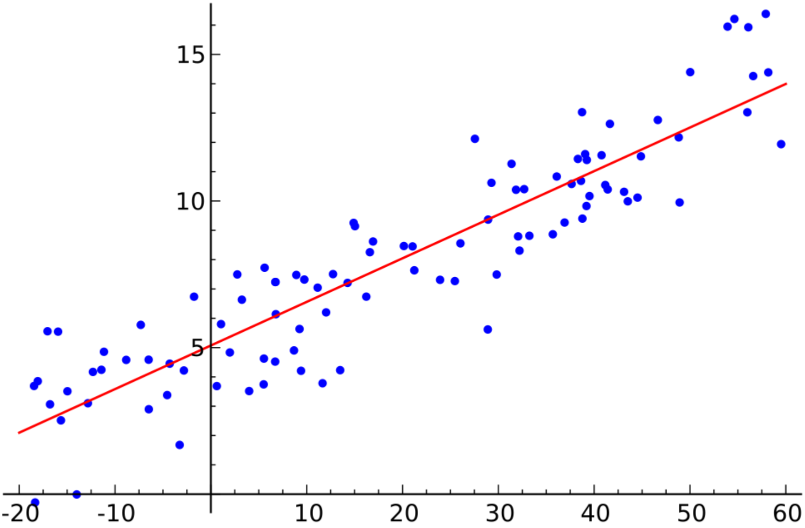
Jean-Philippe Vert (Institut Curie, Mines ParisTech)

History of the course

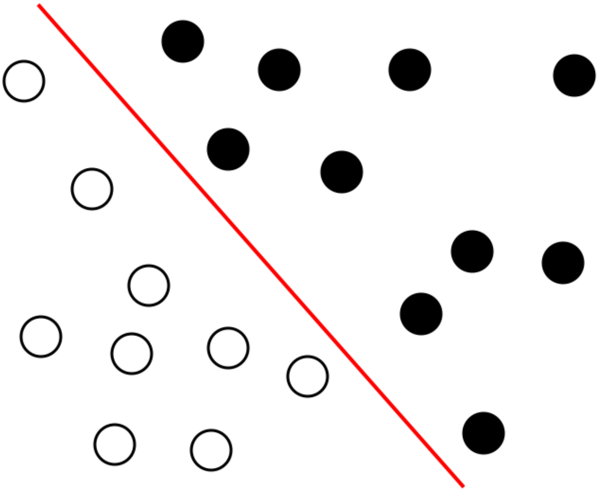
A large part of the course material is due to Jean-Philippe Vert, who gave the course from 2004 to 2015 and who is on sabbatical at UC Berkeley in 2016.

- Along the years, the course has become more and more exhaustive and the slides are probably one of the best reference available on kernels.
- This is a course with a fairly large amount of math, but still accessible to computer scientists who have heard what is a Hilbert space (at least once in their life).

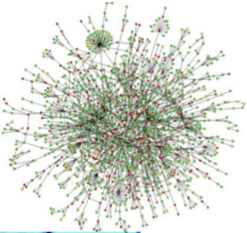
Starting point: what we know is how to solve

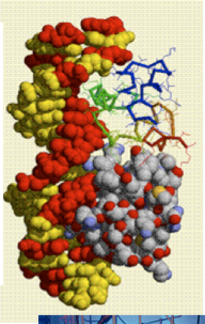


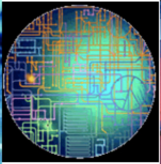
Or



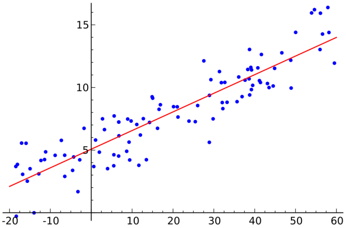
But real data is often more complicated...

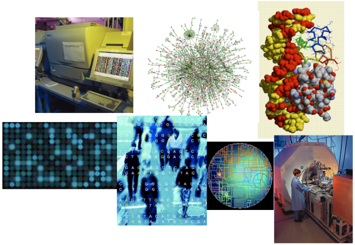






Main goal of this course





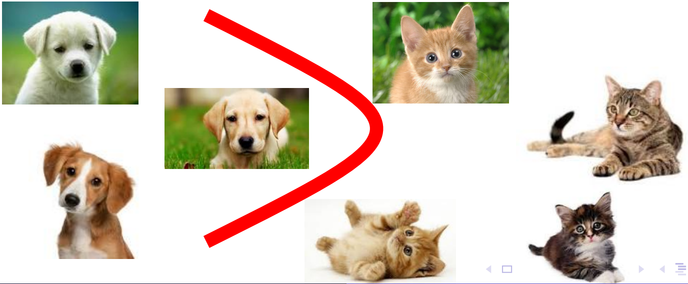
- Extend well-understood, linear statistical learning techniques to real-world, complicated, structured, high-dimensional data (images, texts, time series, graphs, distributions, permutations...)

A concrete supervised learning problem

Regularized empirical risk formulation

The goal is to learn a prediction function $f : \mathcal{X} \to \mathcal{Y}$ given labeled training data $(\mathbf{x}_i \in \mathcal{X}, \mathbf{y}_i \in \mathcal{Y})_{i=1,\ldots,n}$:

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



A concrete supervised learning problem

Unfortunately, linear models often perform poorly unless the problem features are well-engineered or the problem is very simple.

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

First approach to work with a non-linear functional space $\mathcal{F}$

- The "deep learning" space $\mathcal{F}$ is parametrized:

$$f(\mathbf{x}) = \sigma_k(\mathbf{A}_k \sigma_{k-1}(\mathbf{A}_{k-1} \ldots \sigma_2(\mathbf{A}_2 \sigma_1(\mathbf{A}_1 \mathbf{x})) \ldots)).$$

- Finding the optimal $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k$ yields an (intractable) **non-convex** optimization problem in **huge dimension.**

A concrete supervised learning problem

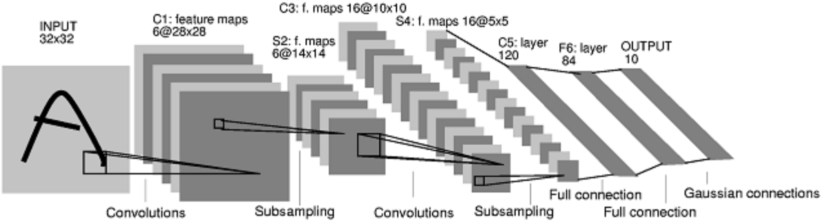


Figure: Exemple of convolutional neural network from **?**

What are the main limitations of neural networks?

- Poor theoretical understanding.
- They require cumbersome hyper-parameter tuning.
- They are hard to regularize.

Despite these shortcomings, they have had an enormous success, thanks to large amounts of labeled data, computational power and engineering.

A concrete supervised learning problem

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

Second approach based on kernels

- Works with possibly infinite-dimensional functional spaces $\mathcal{F}$;
- Works with non-vectorial structured data sets $\mathcal{X}$ such as graphs;
- Regularization is natural and easy.

Current limitations (and open research topics)

- Lack of scalability with n (traditionally $O(n^2)$);
- Lack of adaptivity to data and task.

Organization of the course

Content

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

Practical

- Course homepage with slides, schedules, homework's etc...: `http://lear.inrialpes.fr/people/mairal/teaching/2015-2016/MVA/`.
- Evaluation: 50% homework + 50% data challenge.