Introduction to Three Paradigms in Machine Learning

Julien Mairal

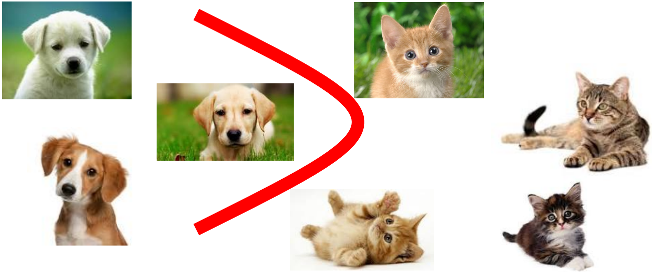
Inria Grenoble

Yerevan, 2018

Optimization is central to machine learning

In supervised learning, we learn a **prediction function** $h : \mathcal{X} \to \mathcal{Y}$ given labeled training data $(x_i, y_i)_{i=1,\ldots,n}$ with x_i in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

$$\min_{h \in \mathcal{H}} \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(y_i, h(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(h)}_{\text{regularization}} .$$



[Vapnik, 1995, Shalev-Shwartz and Ben-David, 2014, Bottou et al., 2016]...

Optimization is central to machine learning

In supervised learning, we learn a **prediction function** $h : \mathcal{X} \rightarrow \mathcal{Y}$ given labeled training data $(x_i, y_i)_{i=1,\ldots,n}$ with x_i in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

$$\min_{h \in \mathcal{H}} \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(y_i, h(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(h)}_{\text{regularization}} .$$

The labels y_i are in

- $\{-1, +1\}$ for **binary** classification.
- $\{1, \ldots, K\}$ for **multi-class** classification.
- $\mathbb{R}$ for **regression**.
- $\mathbb{R}^k$ for **multivariate regression**.
- any general set for **structured prediction**.

Optimization is central to machine learning

In supervised learning, we learn a **prediction function** $h : \mathcal{X} \to \mathcal{Y}$ given labeled training data $(x_i, y_i)_{i=1,\ldots,n}$ with x_i in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

$$\min_{h \in \mathcal{H}} \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(y_i, h(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda\Omega(h)}_{\text{regularization}} .$$

Example with linear models: logistic regression, SVMs, *etc.*

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

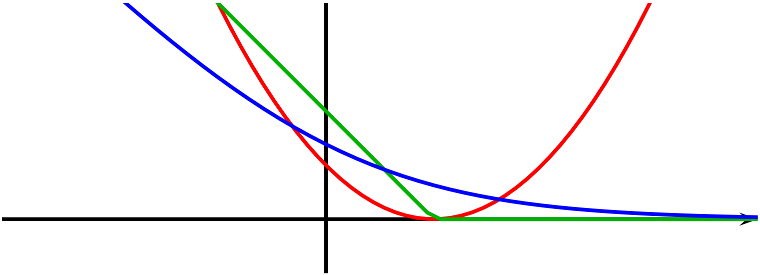
Optimization is central to machine learning

A few examples of linear models with no bias b:

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}(y_i - w^\top x_i)^2 + \lambda \|w\|_2^2.$$

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i w^\top x_i) + \lambda \|w\|_2^2.$$

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y_i w^\top x_i}\right) + \lambda \|w\|_2^2.$$



Optimization is central to machine learning

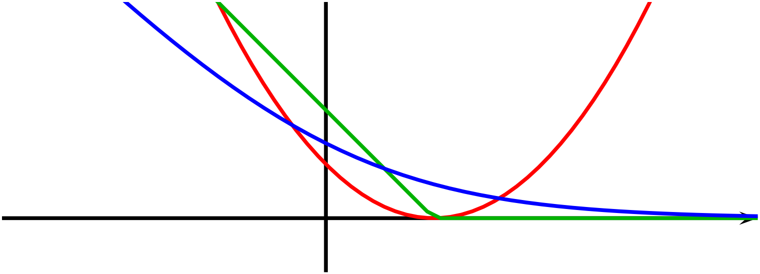
A few examples of linear models with no bias b:

Ridge regression:

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}(y_i - w^\top x_i)^2 + \lambda \|w\|_2^2.$$

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i w^\top x_i) + \lambda \|w\|_2^2.$$

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y_i w^\top x_i}\right) + \lambda \|w\|_2^2.$$



Optimization is central to machine learning

A few examples of linear models with no bias b:

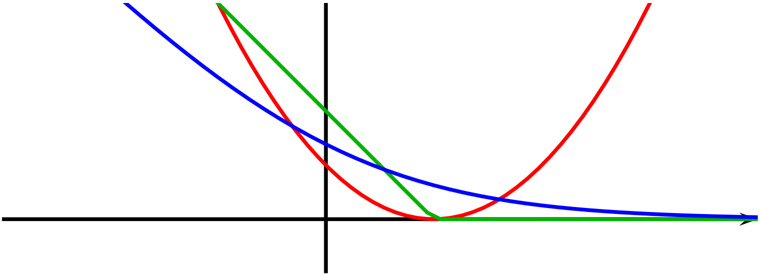
Ridge regression:

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}(y_i - w^\top x_i)^2 + \lambda \|w\|_2^2.$$

Linear SVM:

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i w^\top x_i) + \lambda \|w\|_2^2.$$

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y_i w^\top x_i}\right) + \lambda \|w\|_2^2.$$



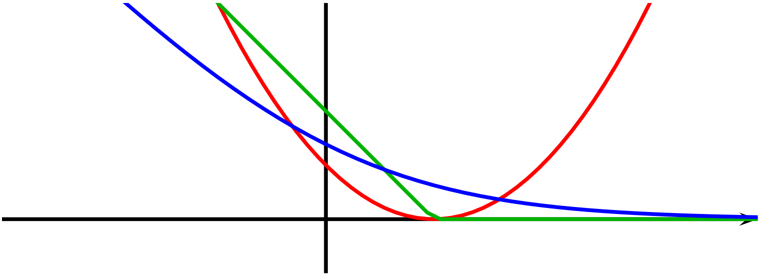
Optimization is central to machine learning

A few examples of linear models with no bias b:

Ridge regression: $$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}(y_i - w^\top x_i)^2 + \lambda \|w\|_2^2.$$

Linear SVM: $$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i w^\top x_i) + \lambda \|w\|_2^2.$$

Logistic regression: $$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y_i w^\top x_i}\right) + \lambda \|w\|_2^2.$$



Optimization is central to machine learning

The previous formulation is called *empirical risk minimization*; it follows a classical scientific paradigm:

1. **observe** the world (gather data);
2. **propose models** of the world (design and learn);
3. **test** on new data (estimate the generalization error).

Optimization is central to machine learning

The previous formulation is called *empirical risk minimization*; it follows a classical scientific paradigm:

1. **observe** the world (gather data);
2. **propose models** of the world (design and learn);
3. **test** on new data (estimate the generalization error).

A general principle

It underlies many paradigms:

- deep neural networks,
- kernel methods,
- **sparse estimation.** (main topic of this sequence of lectures)
- . . .

Optimization is central to machine learning

The previous formulation is called *empirical risk minimization*; it follows a classical scientific paradigm:

1. **observe** the world (gather data);
2. **propose models** of the world (design and learn);
3. **test** on new data (estimate the generalization error).

Even with simple linear models, it leads to challenging problems in optimization:

- **scaling** both in the problem size n and dimension p;
- **exploiting the problem structure** (sum, composite);
- obtaining **convergence and numerical stability** guarantees;
- obtaining **statistical guarantees**.

Optimization is central to machine learning

The previous formulation is called *empirical risk minimization*; it follows a classical scientific paradigm:

1. **observe** the world (gather data);
2. **propose models** of the world (design and learn);
3. **test** on new data (estimate the generalization error).

It is not limited to supervised learning

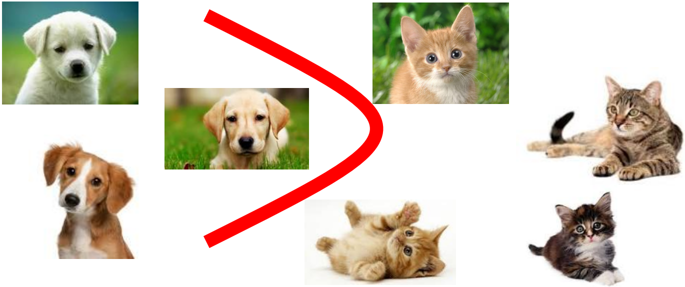
$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(f(x_i)) + \lambda \Omega(f).$$

- L is not a classification loss any more;
- K-means, PCA, EM with mixture of Gaussian, matrix factorization,... can be expressed that way.

Paradigm 1: Deep neural networks

The goal is to learn a **prediction function** $f : \mathbb{R}^p \to \mathbb{R}$ given labeled training data $(x_i, y_i)_{i=1,\ldots,n}$ with x_i in $\mathbb{R}^p$, and y_i in $\mathbb{R}$:

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



Paradigm 1: Deep neural networks

The goal is to learn a **prediction function** $f : \mathbb{R}^p \to \mathbb{R}$ given labeled training data $(x_i, y_i)_{i=1,\ldots,n}$ with x_i in $\mathbb{R}^p$, and y_i in $\mathbb{R}$:

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

What is specific to multilayer neural networks?

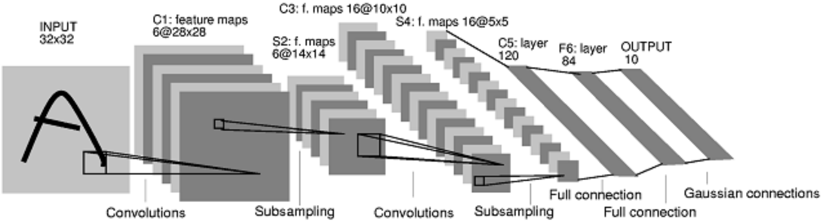
- The "neural network" space $\mathcal{F}$ is explicitly parametrized by:

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

- Linear operations are either unconstrained (fully connected) or involve parameter sharing (e.g., convolutions).
- Finding the optimal $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k$ yields a **non-convex** optimization problem in **huge dimension.**

Paradigm 1: Deep neural networks

Picture from LeCun et al. [1998]

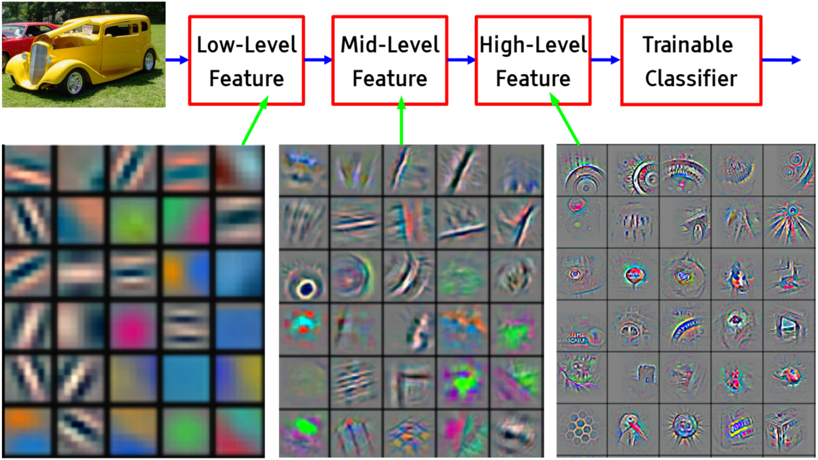


What are the main features of CNNs?

- they capture **compositional** and **multiscale** structures in images;
- they provide some **invariance**;
- they model **local stationarity** of images at several scales;
- they are **state-of-the-art** in many fields.

Paradigm 1: Deep neural networks

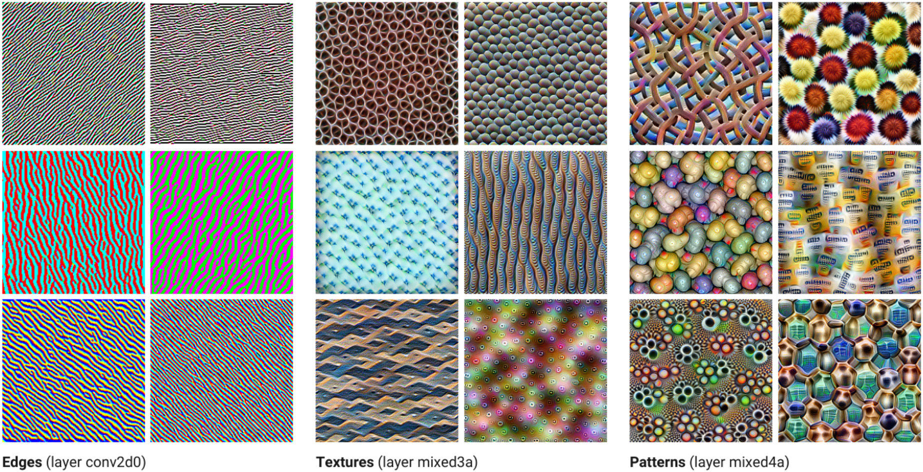
The keywords: **multi-scale, compositional, invariant, local features**.
Picture from Y. LeCun's tutorial:



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Paradigm 1: Deep neural networks

Picture from Olah et al. [2017]:



Paradigm 1: Deep neural networks

Picture from Olah et al. [2017]:

Patterns (layer mixed4a) **Parts** (layers mixed4b & mixed4c) **Objects** (layers mixed4d & mixed4e)

Paradigm 1: Deep neural networks

ImageNet: 1000 image categories, 10M hand-labeled images.
Picture from unknown source:

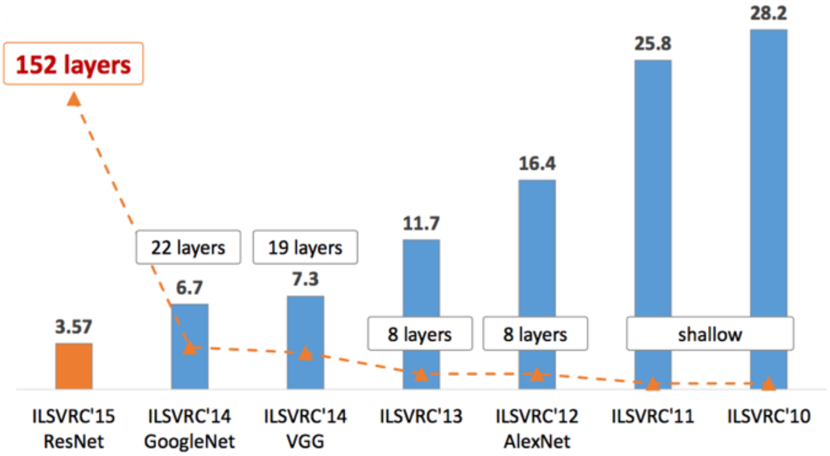


Figure: Top-5 error rate

Paradigm 1: Deep neural networks

What are current high-potential problems to solve?

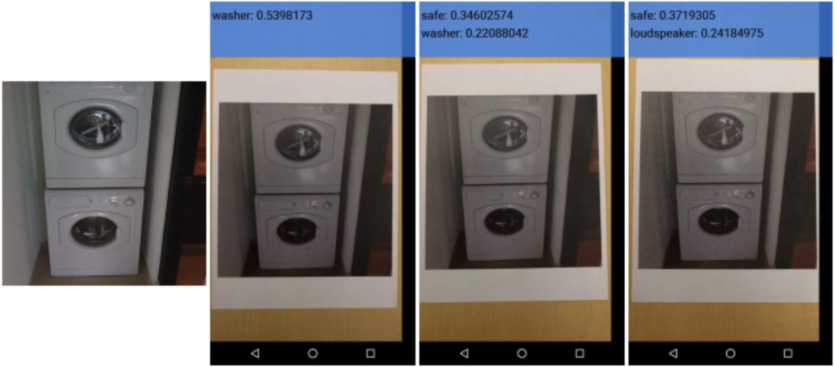
1. lack of **stability** (see next slide).
2. learning with **few labeled data**.
3. learning with **no supervision** (see Tab. from Bojanowski and Joulin, 2017).

Method	Acc@1
Random (Noroozi & Favaro, 2016)	12.0
SIFT+FV (Sánchez et al., 2013)	55.6
Wang & Gupta (2015)	29.8
Doersch et al. (2015)	30.4
Zhang et al. (2016)	35.2
[1]Noroozi & Favaro (2016)	38.1
BiGAN (Donahue et al., 2016)	32.2
NAT	36.0

Table 3. Comparison of the proposed approach to state-of-the-art unsupervised feature learning on ImageNet. A full multi-layer perceptron is retrained on top of the features. We compare to several self-supervised approaches and an unsupervised approach.

Paradigm 1: Deep neural networks

Illustration of instability. Picture from Kurakin et al. [2016].



(a) Image from dataset (b) Clean image (c) Adv. image, $\epsilon = 4$ (d) Adv. image, $\epsilon = 8$

Figure: Adversarial examples are generated by computer; then printed on paper; a new picture taken on a smartphone fools the classifier.

Paradigm 1: Deep neural networks

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

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**?

$$|f(x) - f(x')| \text{ should be close if } x \text{ and } x' \text{ are “similar”}.$$

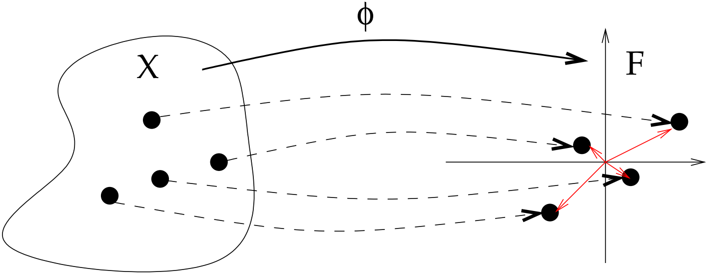
- what does it mean for x and x' to be "similar"?
- what should be a good **regularization function** Ω?

Paradigm 2: Kernel methods

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

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

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



[Shawe-Taylor and Cristianini, 2004, Schölkopf and Smola, 2002]...

Paradigm 2: Kernel methods

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

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: for all x, x' in $\mathcal{X}$,

$$|f(x) - f(x')| \leq \|f\|_{\mathcal{H}} \|\phi(x) - \phi(x')\|_{\mathcal{H}}.$$

Paradigm 2: Kernel methods

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

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: for all x, x' in $\mathcal{X}$,

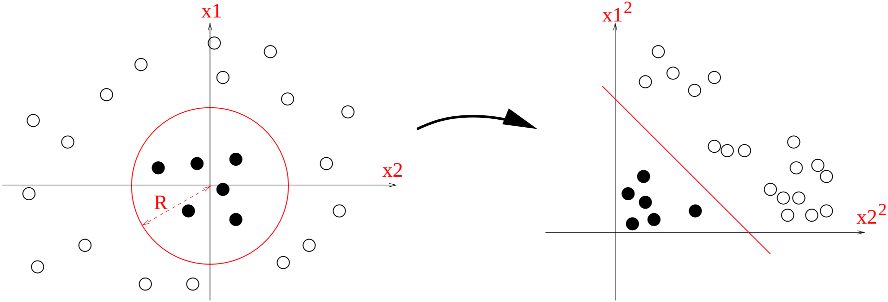
$$|f(x) - f(x')| \leq \|f\|_{\mathcal{H}} \|\phi(x) - \phi(x')\|_{\mathcal{H}}.$$

The principle is **generic** and does not assume anything about the nature of the set $\mathcal{X}$ (vectors, sets, graphs, sequences).

Paradigm 2: Kernel methods

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 $f(x) = \langle \varphi(x), f \rangle_{\mathcal{H}}$ in $\mathcal{H}$ may correspond to a **non-linear** model in $\mathcal{X}$.

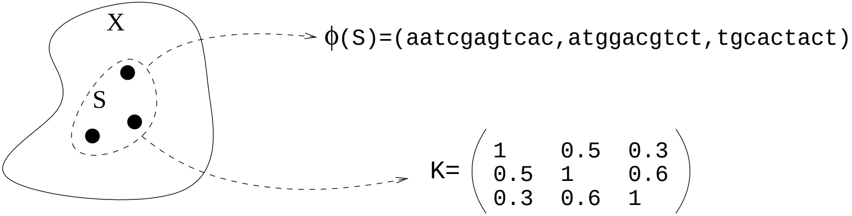


Paradigm 2: Kernel methods (technical parenthesis)

How does it work? representation by pairwise comparisons

- Define a "comparison function": $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$.
- Represent a set of n data points $\mathcal{S} = \{x_1, \ldots, x_n\}$ by the $n \times n$ **matrix**:

$$\mathbf{K}_{ij} := K(x_i, x_j).$$

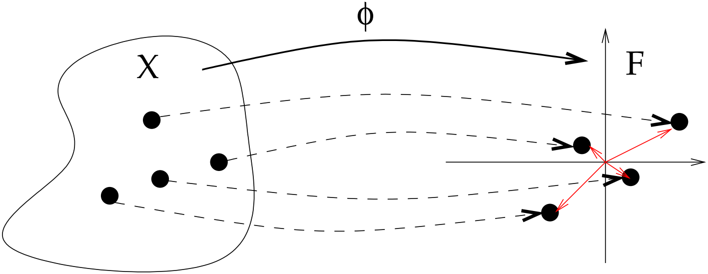


Paradigm 2: Kernel methods (technical parenthesis)

Theorem (Aronszajn, 1950)

$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping $\varphi : \mathcal{X} \to \mathcal{H}$, such that

$$\text{for any } x, x' \text{ in } \mathcal{X}, \qquad K(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}.$$



Paradigm 2: Kernel methods (technical parenthesis)

Mathematical details

- the only thing we require about K is **symmetry** and **positive definiteness**

$$\forall x_1, \ldots, x_n \in \mathcal{X}, \alpha_1, \ldots, \alpha_n \in \mathbb{R}, \quad \sum_{ij} \alpha_i \alpha_j K(x_i, x_j) \geq 0.$$

- then, there exists a Hilbert space $\mathcal{H}$ of functions $f : \mathcal{X} \to \mathbb{R}$, called the **reproducing kernel Hilbert space (RKHS)** such that

$$\forall f \in \mathcal{H}, x \in \mathcal{X}, \quad f(x) = \langle \varphi(x), f \rangle_{\mathcal{H}},$$

and the mapping $\varphi : \mathcal{X} \to \mathcal{H}$ (from Aronszajn's theorem) satisfies

$$\varphi(x) : y \mapsto K(x, y).$$

Paradigm 2: Kernel methods (technical parenthesis)

Why mapping data in $\mathcal{X}$ to the functional space $\mathcal{H}$?

- it becomes feasible to learn a prediction function $f \in \mathcal{H}$:

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

(why? the solution lives in a finite-dimensional hyperplane).

- **non-linear** operations in $\mathcal{X}$ become **inner-products** in $\mathcal{H}$ since

$$\forall f \in \mathcal{H}, x \in \mathcal{X}, \quad f(x) = \langle \varphi(x), f \rangle_{\mathcal{H}}.$$

- the norm of the RKHS is a **natural regularization function**:

$$|f(x) - f(x')| \leq \|f\|_{\mathcal{H}} \|\varphi(x) - \varphi(x')\|_{\mathcal{H}}.$$

Paradigm 2: Kernel methods (non-technical parenthesis)

What are the main features of kernel methods?

- builds **well-studied functional spaces** to do machine learning;
- **decoupling** of data representation and learning algorithm;
- typically, **convex optimization problems** in a supervised context;
- **versatility**: applies to vectors, sequences, graphs, sets,. . . ;
- **natural regularization function** to control the learning capacity;

[Shawe-Taylor and Cristianini, 2004, Schölkopf and Smola, 2002, Müller et al., 2001]

Paradigm 2: Kernel methods (non-technical parenthesis)

What are the main features of kernel methods?

- builds **well-studied functional spaces** to do machine learning;
- **decoupling** of data representation and learning algorithm;
- typically, **convex optimization problems** in a supervised context;
- **versatility**: applies to vectors, sequences, graphs, sets,...;
- **natural regularization function** to control the learning capacity;

But...

- **decoupling** of data representation and learning may not be a good thing, according to recent **supervised** deep learning success.
- requires **kernel design**.
- $O(n^2)$ **scalability problems**.

[Shawe-Taylor and Cristianini, 2004, Schölkopf and Smola, 2002, Müller et al., 2001]

Paradigm 3: The sparsity principle

Let us consider again the classical scientific paradigm:

1. **observe** the world (gather data);
2. **propose models** of the world (design and learn);
3. **test** on new data (estimate the generalization error).

[Corfield et al., 2009].

Paradigm 3: The sparsity principle

Let us consider again the classical scientific paradigm:

1. **observe** the world (gather data);
2. **propose models** of the world (design and learn);
3. **test** on new data (estimate the generalization error).

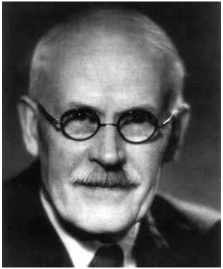
But...

- it is not always possible to distinguish the generalization error of various models based on available data.
- when a complex model A performs slightly better than a simple model B, should we prefer A or B?
- generalization error requires a predictive task: what about unsupervised learning? which measure should we use?
- we are also leaving aside the problem of non i.i.d. train/test data, biased data, testing with counterfactual reasoning...

[Corfield et al., 2009, Bottou et al., 2013, Schölkopf et al., 2012].

Paradigm 3: The sparsity principle

(a) Dorothy Wrinch
1894–1980



(b) Harold Jeffreys
1891–1989

The existence of simple laws is, then, apparently, to be regarded as a quality of nature; and accordingly we may infer that it is justifiable to prefer a simple law to a more complex one that fits our observations slightly better.

[Wrinch and Jeffreys, 1921].

Paradigm 3: The sparsity principle

Remarks: sparsity is...

- appealing for experimental sciences for **model interpretation**;
- (too-)**well understood** in some mathematical contexts:

$$\min_{w \in \mathbb{R}^p} \underbrace{\frac{1}{n}\sum_{i=1}^{n} L\left(y_i, w^\top x_i\right)}_{\text{empirical risk, data fit}} + \underbrace{\lambda \|w\|_1}_{\text{regularization}}.$$

- extremely powerful for **unsupervised learning** in the context of matrix factorization, and **simple to use**.

[Olshausen and Field, 1996, Chen, Donoho, and Saunders, 1999, Tibshirani, 1996]...

Paradigm 3: The sparsity principle

Remarks: sparsity is...

- appealing for experimental sciences for **model interpretation**;
- (too-)**well understood** in some mathematical contexts:

$$\min_{w \in \mathbb{R}^p} \underbrace{\frac{1}{n}\sum_{i=1}^{n} L\left(y_i, w^\top x_i\right)}_{\text{empirical risk, data fit}} + \underbrace{\lambda \|w\|_1}_{\text{regularization}}.$$

- extremely powerful for **unsupervised learning** in the context of matrix factorization, and **simple to use**.

Today's challenges

- Develop sparse and **stable** (and **invariant?**) models.
- Go beyond clustering / low-rank / union of subspaces.

[Olshausen and Field, 1996, Chen, Donoho, and Saunders, 1999, Tibshirani, 1996]...

Some references

On kernel methods

- B. Schölkopf and A. J. Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. 2002.
- J. Shawe-Taylor and N. Cristianini. An introduction to support vector machines and other kernel-based learning methods. 2004.
- 635 slides: `http://members.cbio.mines-paristech.fr/~jvert/svn/kernelcourse/course/2017mva/index.html`

On sparse estimation

- M. Elad. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. 2010.
- J. Mairal, F. Bach, and J. Ponce. Sparse Modeling for Image and Vision Processing. 2014. **free online**.

References I

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14 (1):3207–3260, 2013.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1999.

David Corfield, Bernhard Schölkopf, and Vladimir Vapnik. Falsificationism and statistical learning theory: Comparing the popper and vapnik-chervonenkis dimensions. *Journal for General Philosophy of Science*, 40(1):51–58, 2009.

Alexey Kurakin, Ian Goodfellow, and Samy 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. *P. IEEE*, 86(11):2278–2324, 1998.

References II

K-R Müller, Sebastian Mika, Gunnar Ratsch, Koji Tsuda, and Bernhard Scholkopf. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, 12(2):181–201, 2001.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. 2017.

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381: 607–609, 1996.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

References III

John Shawe-Taylor and Nello Cristianini. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2004.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1995.

D. Wrinch and H. Jeffreys. XLII. On certain fundamental principles of scientific inquiry. *Philosophical Magazine Series 6*, 42(249):369–390, 1921.