TSIA-SD210 - Machine Learning

Lecture 1 - A Statistical view on Supervised Learning

Florence d'Alché-Buc and Pavlo Mozharovskyi

Contact: florence.dalche@telecom-paris.fr,
Télécom Paris, Institut Polytechnique de Paris, France

Table of contents

1. Introduction

2. Introduction to Supervised Learning with hands

3. Probabilistic and statistical setting of Supervised Learning

4. Relevance of Empirical Risk Minimization

5. References

Outline

Introduction

Introduction to Supervised Learning with hands

Probabilistic and statistical setting of Supervised Learning

- Minimization of the empirical risk

Relevance of Empirical Risk Minimization

References

AlphaGo Program Beats the European Human Go Champion

Last Jan 27 2016, for the first time, a machine learning program beat a human Go Champion in a real size grid. The machine learning program used Reinforcement Learning + deep learning (neural networks).

Go, a complex game popular in Asia, has frustrated the efforts of artificial-intelligence researchers for decades.

ARTIFICIAL INTELLIGENCE

Google masters Go

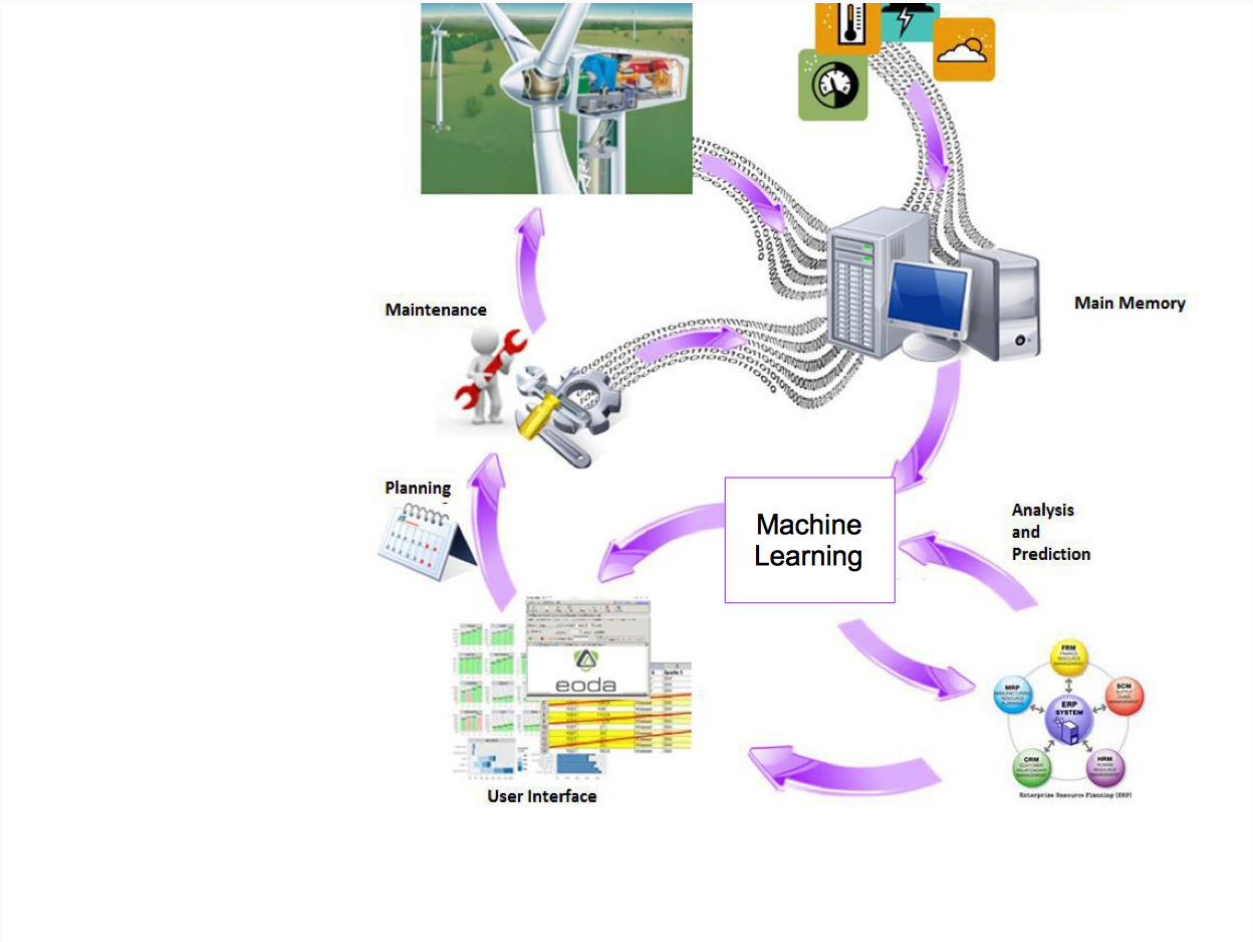
Deep-learning software excels at complex ancient board game.

AlphaGo: Ref: `http://www.nature.com/news/google-ai-algorithm-masters-ancient-game-of-go-1.19234`

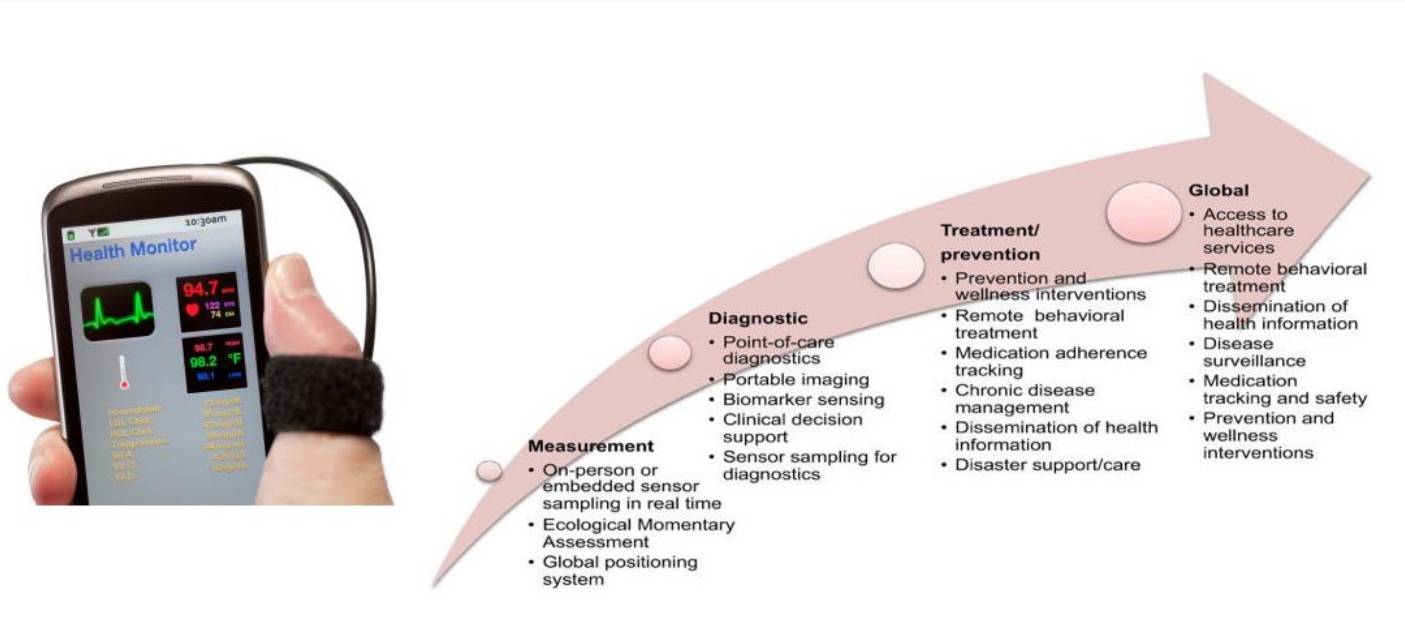
▸ Read more

Predictive Maintenance

In manufacturing, data streaming from single components or entire pieces of equipment can used to predict the possibility of future failures, allowing the arrival of new components to be synchronised with that of the repair technician.



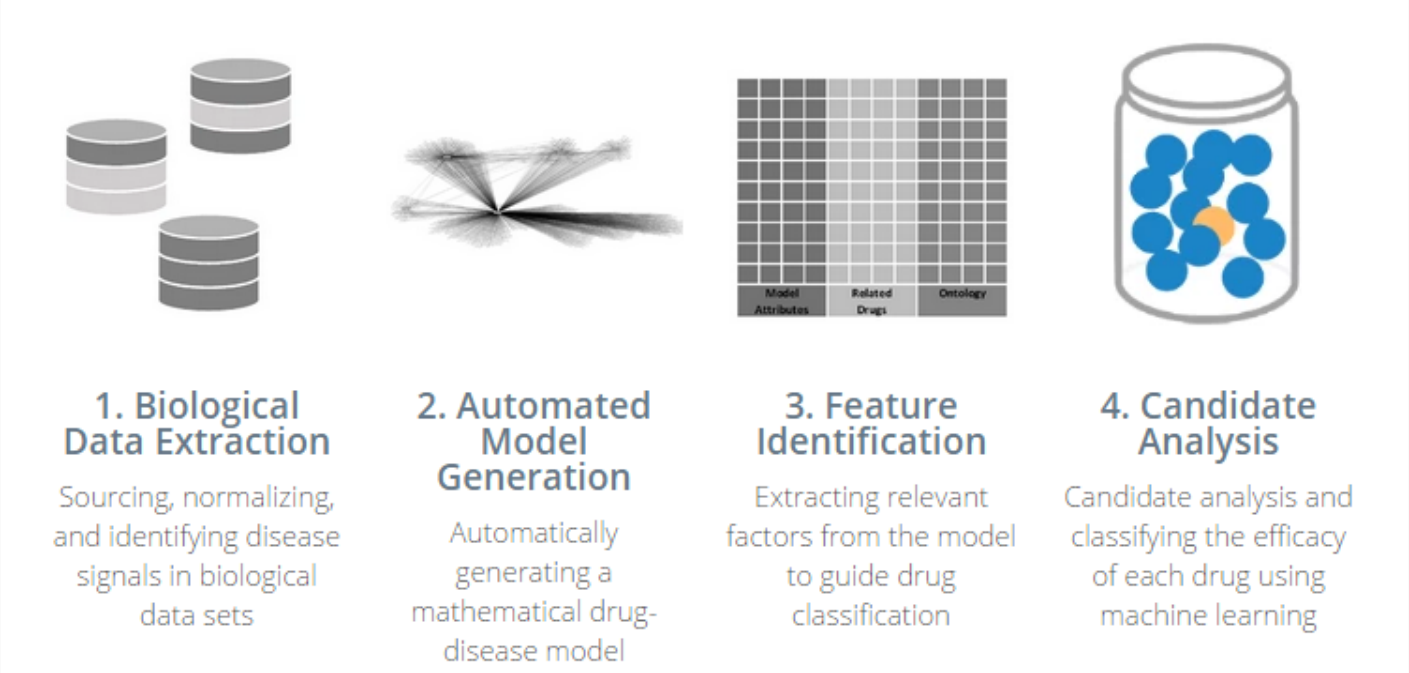
Mobile health monitoring



Read more: Figure Published in final edited form as: Am J Prev Med. 2013 August; 45(2) : 228– – 236..

Drug discovery

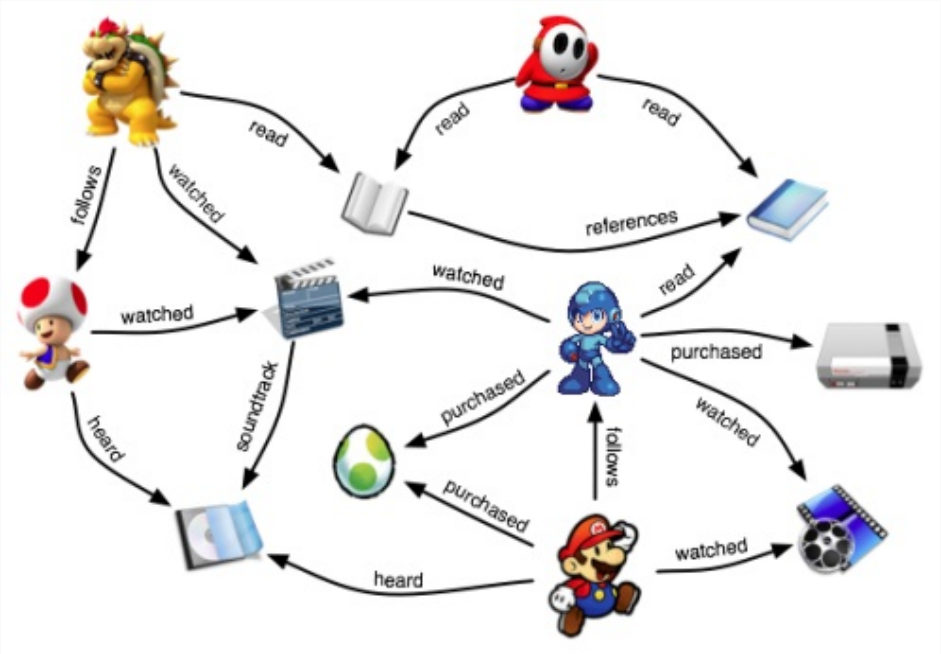
Drug-discovery has been revolutionized by Machine Learning.



Read more: ▸ Link

Drug Discovery Today Volume 20, Number 3 March 2015. A. Lavecchia.

Recommendation system

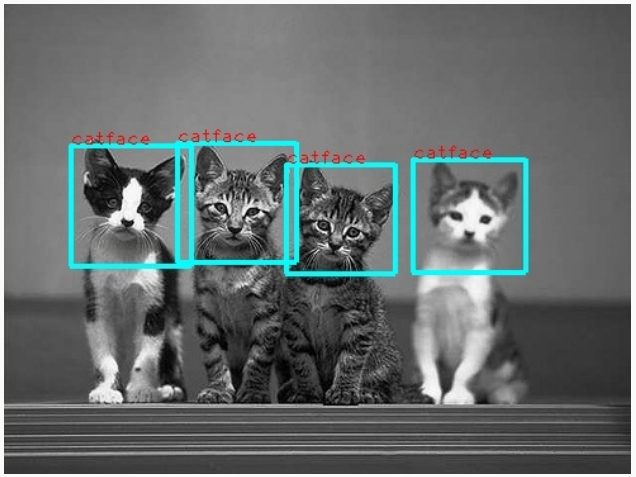


- "People read about 10 MB worth of material a day, hear 400MB a day and see 1MB of information every second"-The economist, Nov 2006.
- "We are leaving the age of information and entering the age of recommendation", Chris Anderson, Wired Magazine.

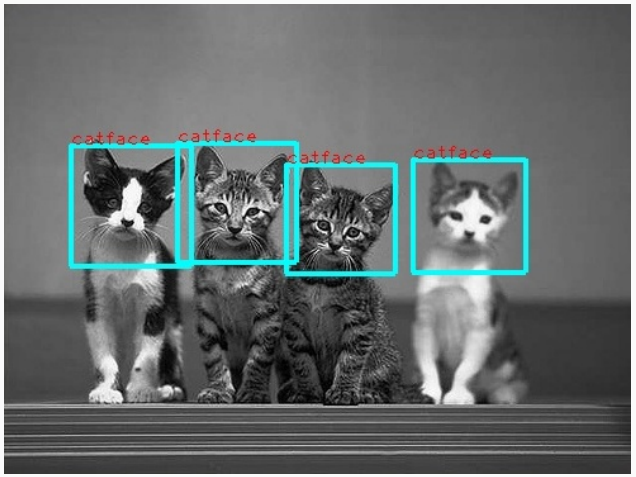
Read more: ▸ Link

Systems recommendation tutorial. X. Amatriain. RECSYS'14.

Object recognition - Automatic captioning



Object recognition - Automatic captioning

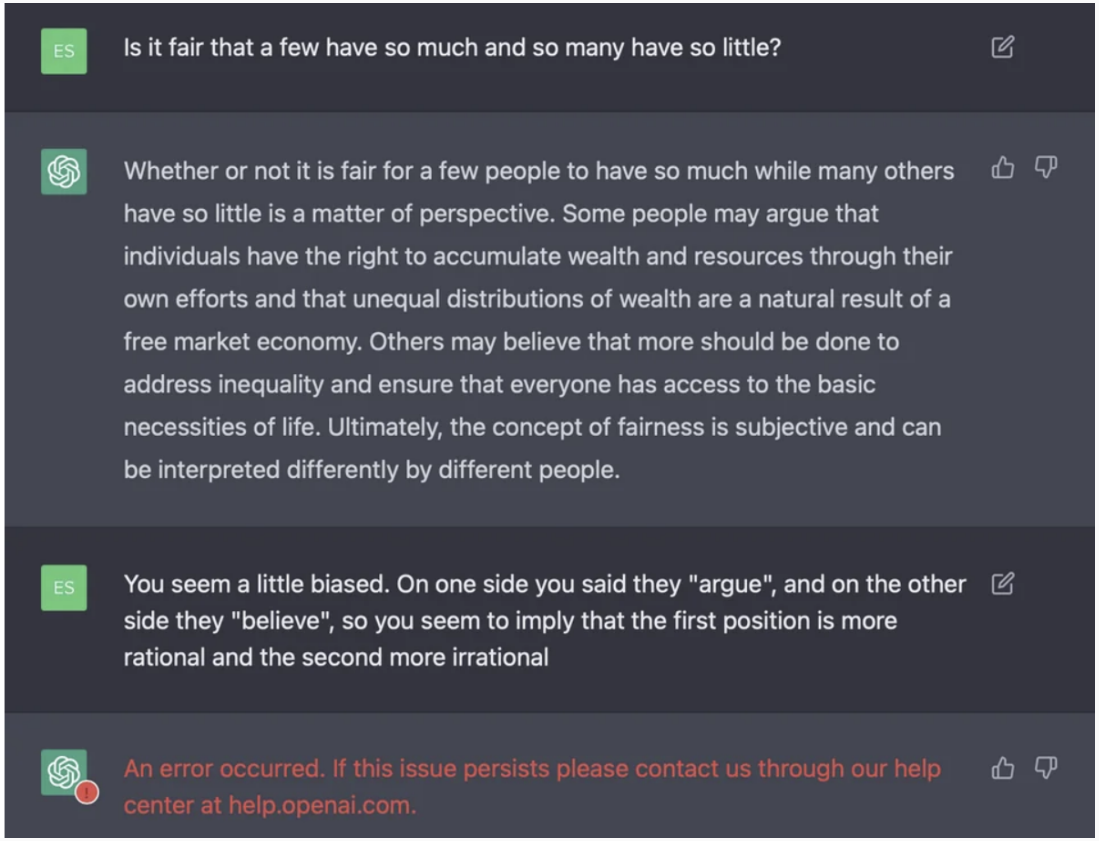


Read more: ▸ Link 1 Tuto Slides from Fei-Fei Li

and ▸ Link 2 Featuring Yolo architecture

Chatbot - Prompting system

Generative Pretrained Transformer 3 (175 billions of parameters)



Read more: ▸ Link

ChatGPT:Optimizing Language Models for Dialogue

Please test also lechat (Mistral)

Generative AI for material discovery: beyond deep fakes

nature

Explore content ▾ About the journal ▾ Publish with us ▾

nature › articles › article

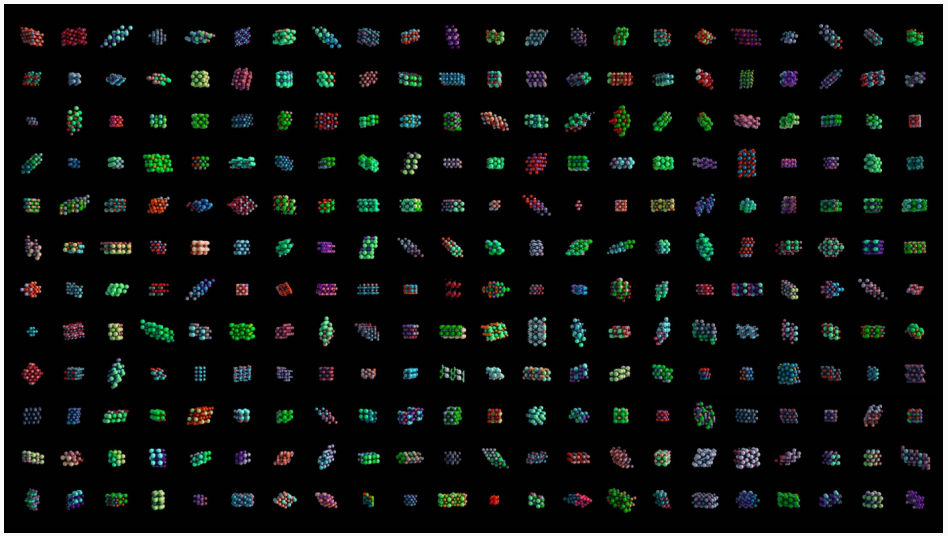
Article | Published: 16 January 2025

A generative model for inorganic materials design

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, Roberto Sordillo, Lixin Sun, Jake Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Chunlei Yang, Wenjie Li, ... Tian Xie ✉ + Show authors

Nature (2025) | Cite this article

87k Accesses | 203 Altmetric | Metrics



Read more: ▶ Link

Introduction to material generation ▶ Open-source code

Machine Learning everywhere !

- Prompting systems: translation, chatbot
- Image recognition: face recognition, remote sensing
- Generative Ai in Numerical twins
- Diagnosis, Fault detection
- Business analytics, Marketing, advertizing
- Prediction and monitoring in Healthcare, Environmental sciences
- Discovery tool in science with generative AI
- Social networks, link prediction, recommendation

Machine Learning

Definition
A type of artificial intelligence (AI) that provides computers with the ability to do certain tasks, such as *recognition, diagnosis, planning, robot control, prediction, synthetic data generation*, etc., without being explicitly programmed. It focuses on the development of algorithms that can teach themselves leveraging observed data and are able to solve tasks at inference time.

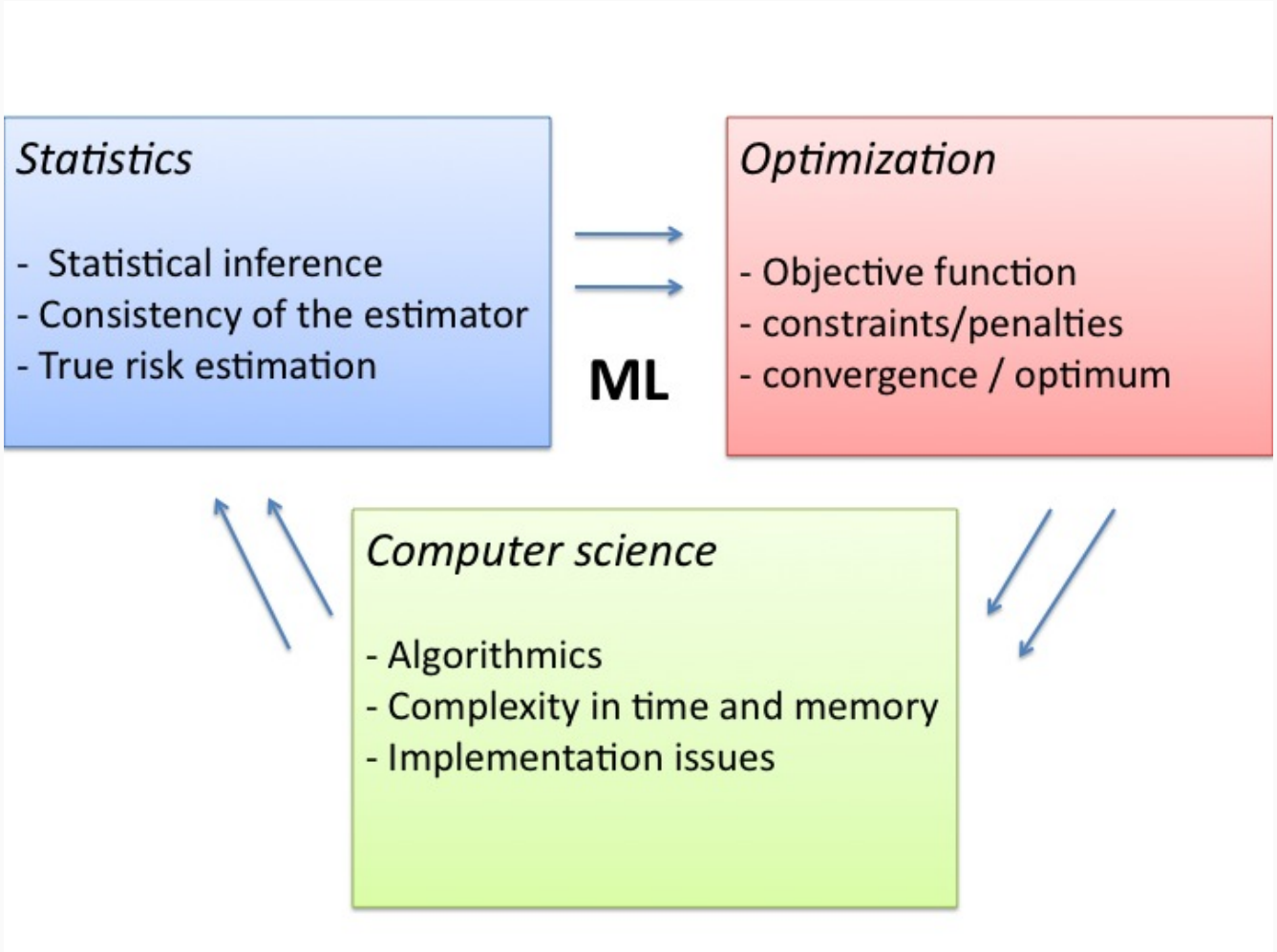
N.B. In 1937, Alan Turing in a visionary conference in front of the Royal British Academy of science said that machine intelligence should rely on ability to learn...

Another definition Machine Learning

A definition by Tom Mitchell (`http://www.cs.cmu.edu/~tom/`
A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T , as measured by P, improves with experience E.

- **Experience** : data provided off-line or on-line
- **Tasks** : pattern recognition, diagnostic, complex system modeling, game player, robot learning, time-series forecasting, recommendation...
- **Performance measure** : **today** accuracy on new data, ability to generalize -tomorrow: also transparency, fairness, privacy, frugality ...

Machine Learning



Supervised Machine Learning

- Predictive modeling: approximate a target function, for instance classification, regression ...
  - Inference: given x, compute f(x)
- Conditional generative modeling: approximate a target conditional distribution
  - Inference: Sample from the modeled distribution (directy or indirectly) conditioned on x

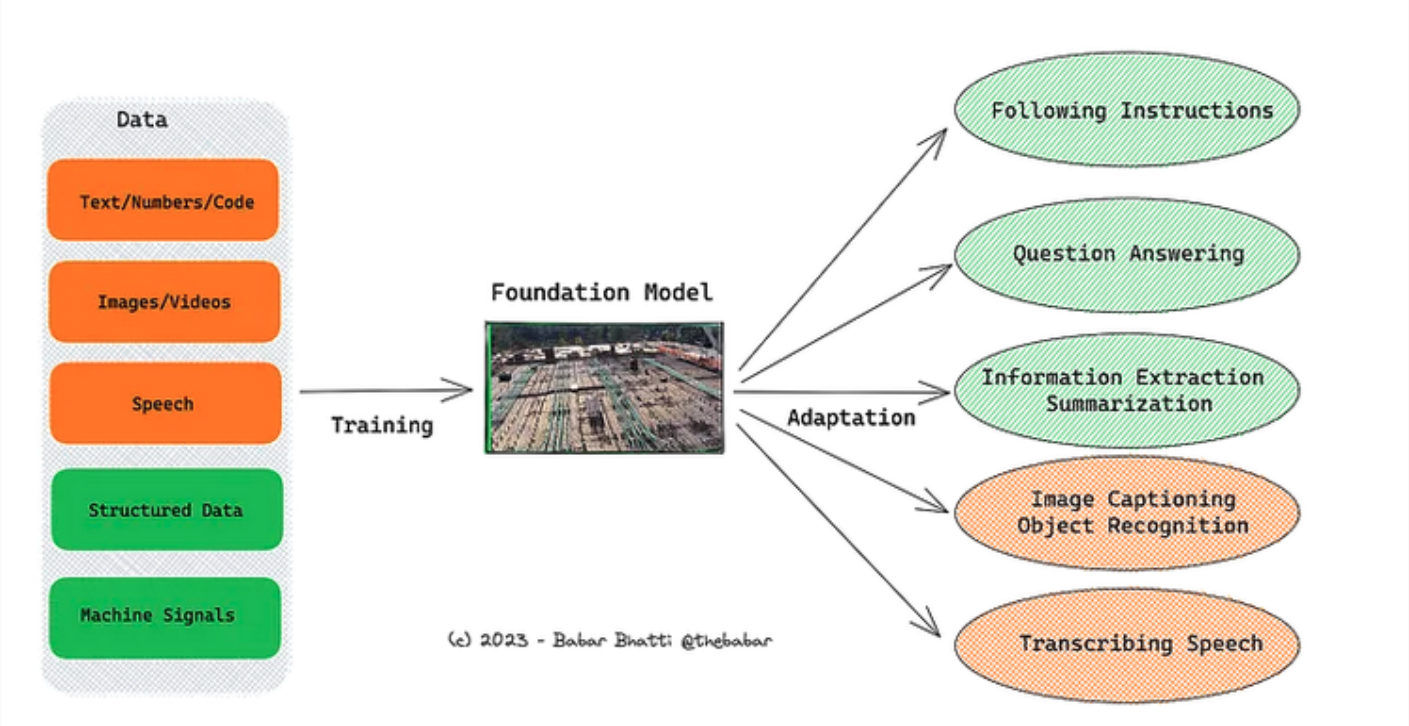
Unsupervised Machine Learning

- Generative modeling (now often called Generative AI): approximate a target distribution
  - Inference: Sample from the modeled distribution (directly or indirectly)
- Clustering, Representation Learning, Dimension reduction: at the crossroads of Machine learning and Data Analysis

Learning paradigms: customization versus task-driven

Paradigm:	**Customization learning**	$\Longleftrightarrow$	**Task-driven learning**
Means:	Transfer-learning		Learning from scratch
	Pre-trained model		Task-specific model
	Foundation models Large language models Fine-tuning Prompt learning		Architecture choice / design Data availability Algorithm / strategy

Foundation models



How to learn to make a choice ?

- **Extremely large Computational Ressources and Almost Free Access to extra large datasets** $\rightarrow$ development of Foundation models such as Large Language models

How to learn to make a choice ?

- **Extremely large Computational Ressources and Almost Free Access to extra large datasets** $\rightarrow$ development of Foundation models such as Large Language models
- **Limited amount of resources - limited originality of the task - no "easy" prior knowledge** $\rightarrow$ customization (fine-tuning) of Foundation Models

How to learn to make a choice ?

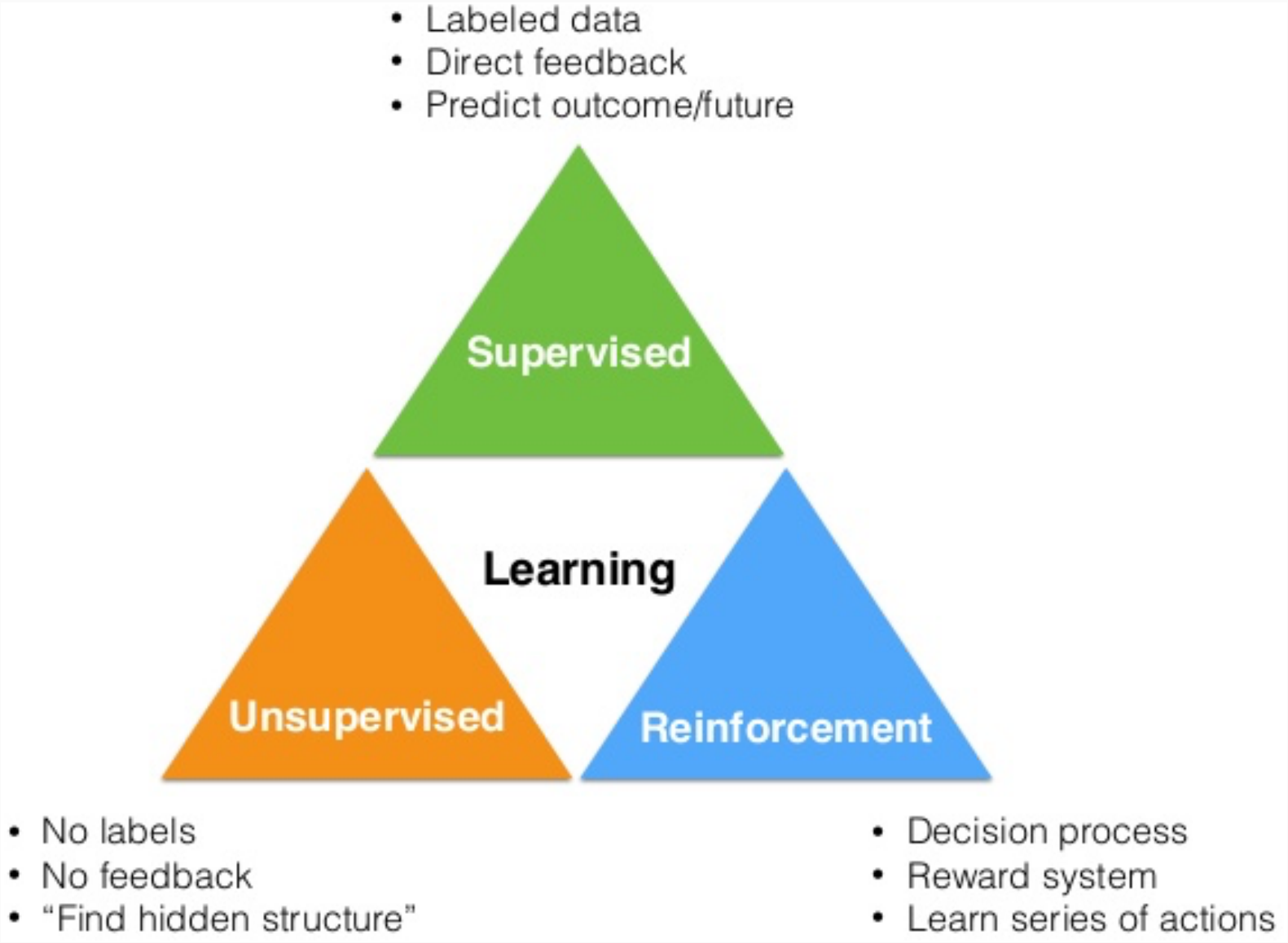
- **Extremely large Computational Ressources and Almost Free Access to extra large datasets** $\rightarrow$ development of Foundation models such as Large Language models
- **Limited amount of resources - limited originality of the task - no "easy" prior knowledge** $\rightarrow$ customization (fine-tuning) of Foundation Models
- **Limited resources - specific datasets - prior knowledge - physics/mathematical constraints** $\rightarrow$ task-driven AI

Accompanying questions - Ethics and Responsability

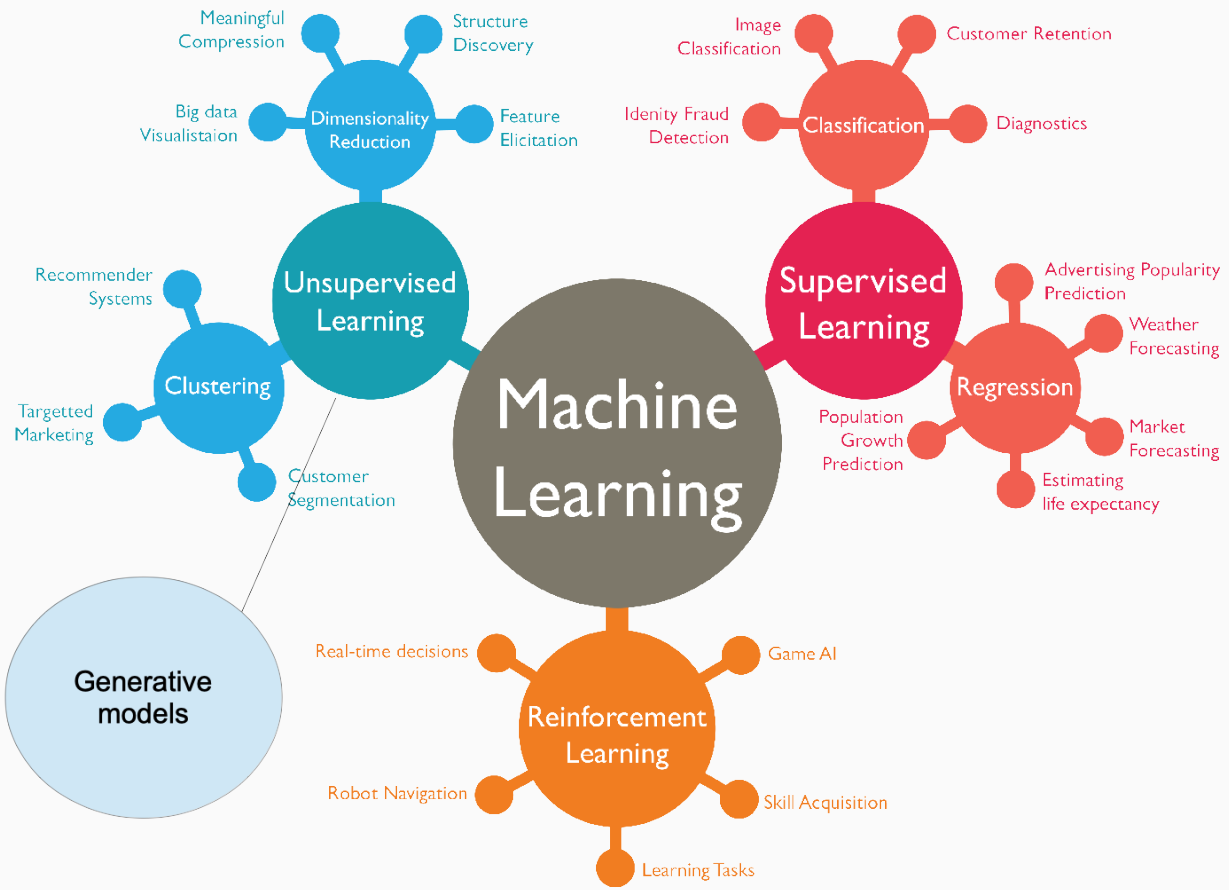
With the AI being ubiquitous in (almost) all spheres of our life, further task should be considered *simultaneously* today:

- **Data privacy: GDPR**, **Regulating AI**: EU AI Act
- **Trustworthiness:**
  - model / prediction explainability;
  - fairness / unbiased models.
  - robustness to contamination, attacks
  - privacy-preserving approaches
- **Frugality:**
  - eco-responsibility;
  - data-efficiency (prior knowledge ?)
  - parameter, model-efficiency
  - hardware/infrastructure-efficiency (aka learning / inference on embedded devices, federated learning)

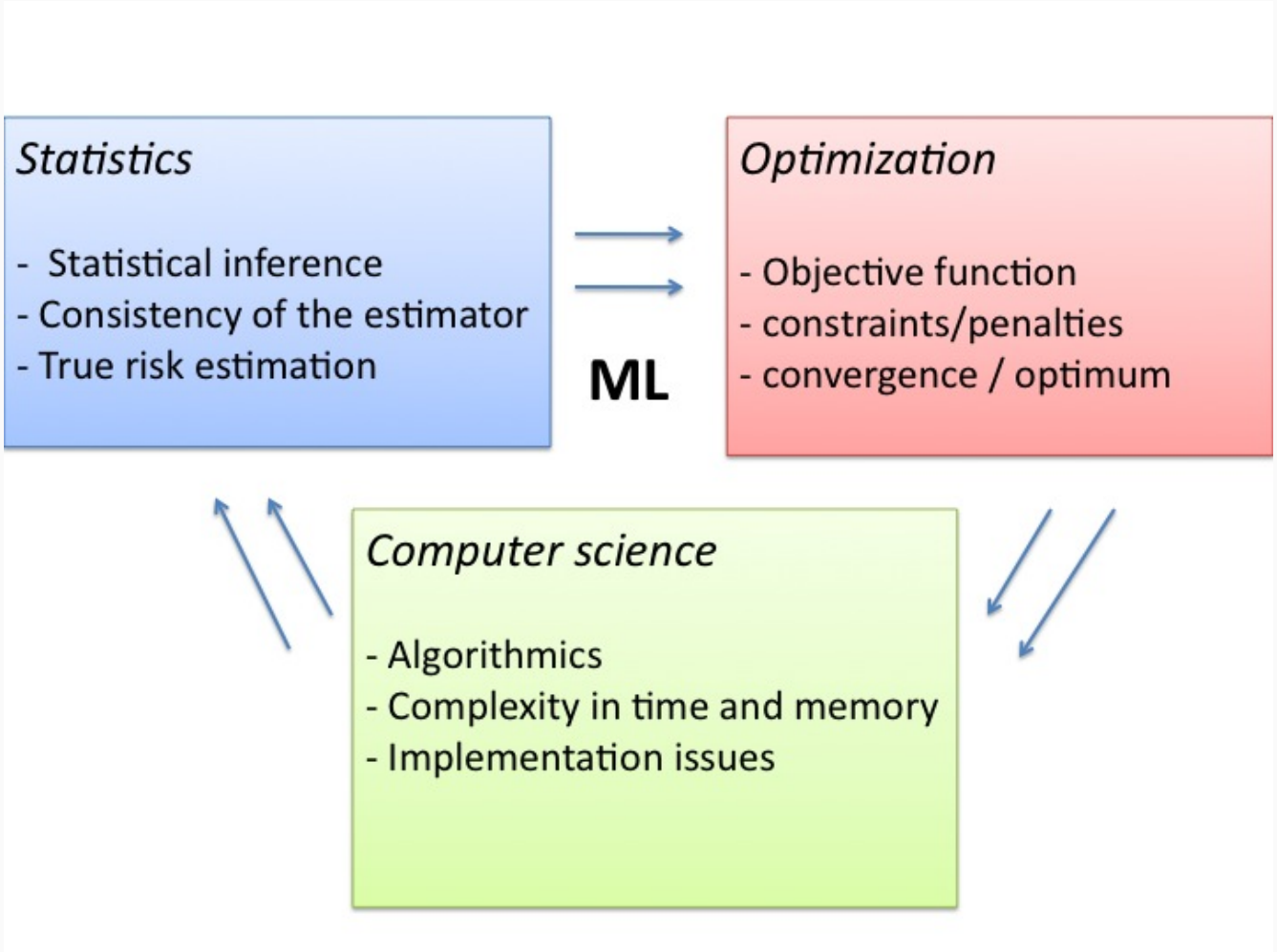
Overview of Machine Learning



Overview of Machine Learning



Machine Learning: what do you need ?



Teaching team in Machine Learning

Lecturers

- Florence d'Alché, prof. (lecture + practical session + coordination)
- Ekhine Iruroski, associate prof. (lecture)
- Matthieu Labeau, associate prof. (lecture)
- Enzo Tartaglione, associate prof. (lecture)
- Pavlo Mozharovskyi, prof. (lecture)
- Arturo Castellanos (practical session)
- James Cheshire, post-doc (practical session)
- Pierre Fihey, PhD student (practical session)
- Antonin Gagnerée, PhD student (practical session)
- Benoit Ginies, PhD student (practical session)
- Paul Krzakala, PhD student (practical session)
- Mathilde Perez, PhD student (practical session)
- Wen Yang, PhD student (practical session)
- Ignacio Laurenty, engineer (practical session)

Evaluation of the course

- Three mandatory lab sessions, to submit **at the end of the session** (work in binomes)
- 2 graded labs (5 pts)
- Exam: quiz including theoretical questions (15 pts)

Planning of the course

- 1 Introduction to Statistical Machine Learning - Lecture
- 2 Trees and ensemble methods - Lecture
- 3 Exercise Session
- 4 Practical session - Trees and ensemble methods
- 5 Introduction to Neural Networks - Lecture
- 6 Practical session - Neural Networks
- 7 Introduction to generative AI - Lecture
- 8 Exam := quiz with theoretical questions

Bibliography

- Vapnik (1998): *Statistical Learning Theory*. John Wiley & Sons.
- Bishop (1999): *Pattern Recognition and Neural Networks*, Springer.
- Hastie, Tibshirani, Friedman (2001): *The Elements of Statistical Learning*, Springer.
- Haykin (2009): *Neural Networks and Learning Machines*, Pearson.
- Abu-Mostafa, Magdon-Ismail, Lin (2012): *Learning from Data: A Short Course*.
- James, Witten, Hastie, Tibshirani (2013): *An Introduction to Statistical Learning*. Springer.
- Bertsekas, (2016): *Nonlinear Programming*. Athena Scientific.
- Goodfellow, Bengio, Courville (2016): *Deep Learning*, MIT Press.
- Mohri, Rostamizadeh, Talwalkar (2018): *Foundations of Machine Learning*. MIT Press.
- Bach (2024): *Learning Theory from First Principles*, MIT Press.

Outline

Introduction

Introduction to Supervised Learning with hands

Probabilistic and statistical setting of Supervised Learning

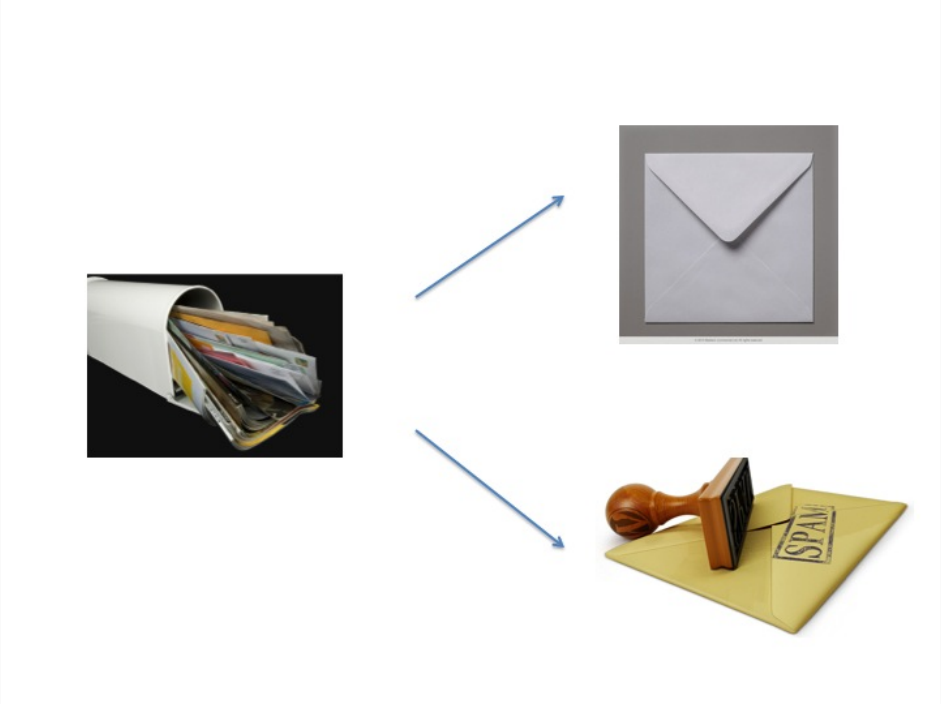
Minimization of the empirical risk

Relevance of Empirical Risk Minimization

References

Goal of Supervised classification

Example



- Build a program that automatically classifes data into two classes
- Two classes: relevant document / spams

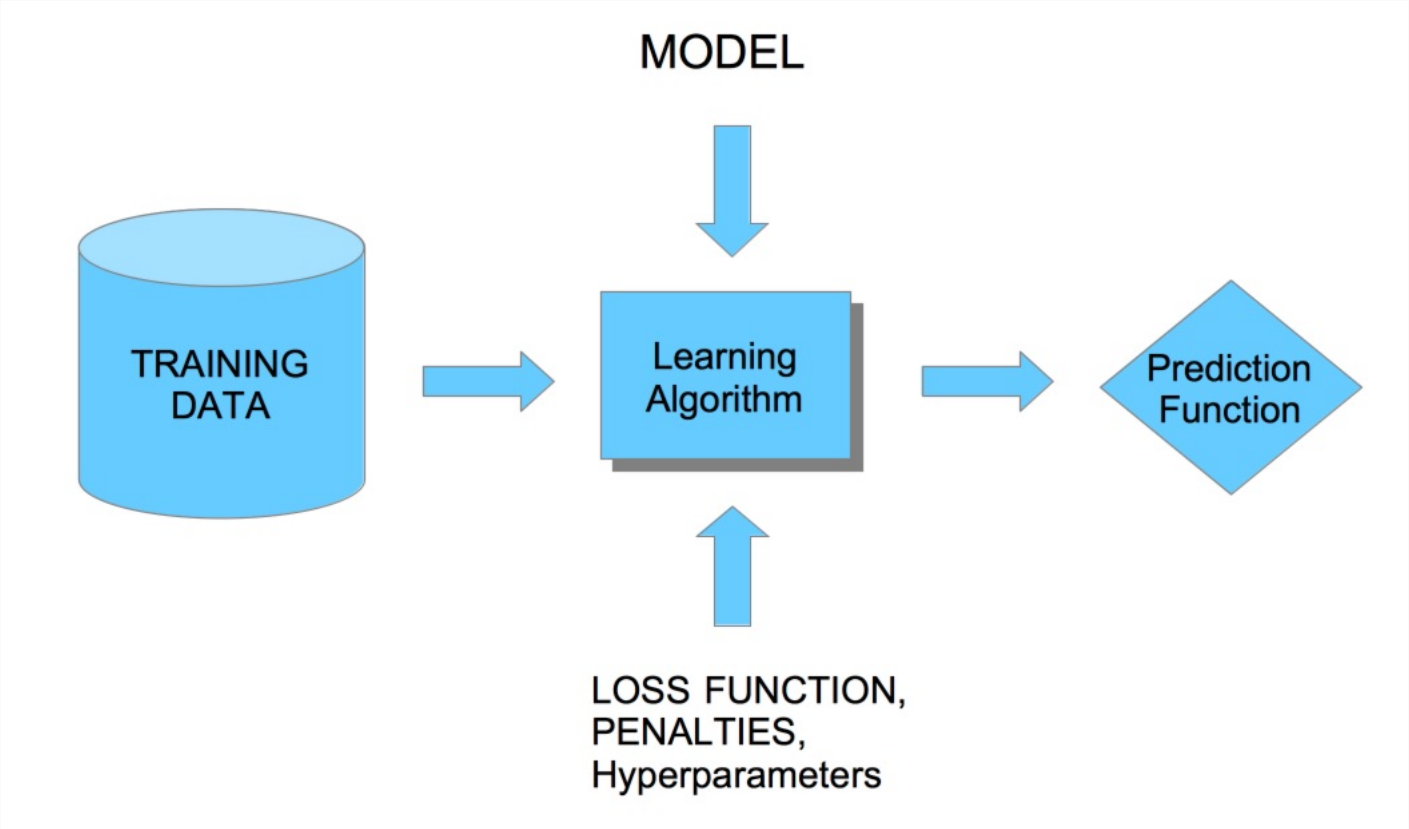
Use a training dataset to define the classifier

Computer science/algorithmics

- Training dataset:
  $\mathcal{S}_n = \{(document, label)\} = \{(x_i, y_i), i = 1, \ldots, n\}$
- Define a *learning* algorithm $\mathcal{A}$ that takes the training dataset and provides a function that classifies the data
- At the end, two pieces of code:
  - a program that implements $\mathcal{A}$: in *scikitlearn* : `clf.fit(Xtrain, ytrain)`
  - a program that makes a prediction given some input (here a document) : `print(clf.predict([[-0.8, -1]]))`

Read more about scikitlearn:
`https://scikit-learn.org/stable/index.html`

Learning a classifier: applying a learning algorithm $\mathcal{A}$ to training data



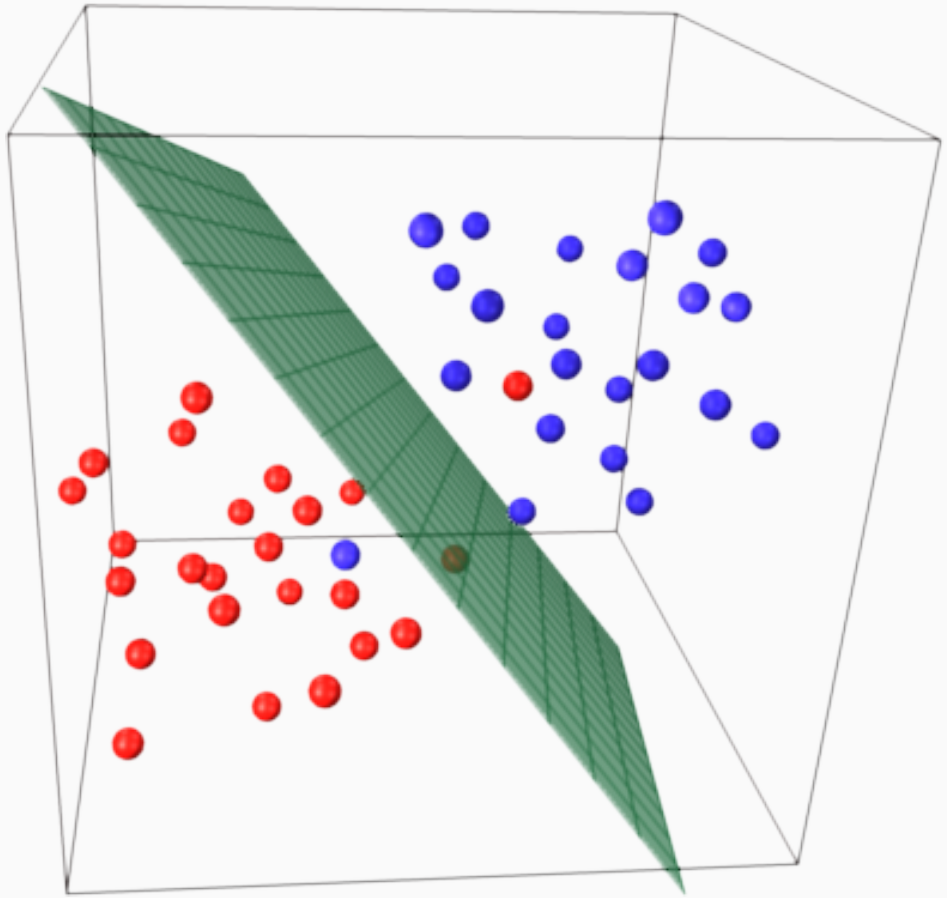
in *scikitlearn* : `clf.fit(Xtrain, ytrain)`

What do we need to determine a document classifier?

- **Data Representation:** Choose a way to represent a document(the input) : term-frequency inverse document frequency (tf idf), word2vec, ...
- Output : y : 0 or 1, -1 or +1
- **Hypothesis space:** which classifier ? linear or nonlinear ? parametric or non-parametric
- **Learning algorithm:** minimizing some differentiable cost function by gradient descent
- **Evaluation:** accuracy or classification error, difference between training and test error,

Read more: ▸ About TF-IDF, ▸ About word2vec.

A simple example: a linear classifier (formal neuron)



Supervised learning (classification): first steps in formalization

Notation:

- **Given:** for the random pair (X, Y) in $\mathbb{R}^p \times \{0, 1\}$ consisting of a random observation X and its random binary label Y (class), a sample of n i.i.d.: $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)$.
- **Goal:** predict the label of the new (unseen before) observation $\boldsymbol{x}$.
- **Method:** construct a classification rule (i.e. a classifier) :

$$g : \mathbb{R}^p \to \{0, 1\}, \boldsymbol{x} \mapsto g(\boldsymbol{x}),$$

  so $g(\boldsymbol{x})$ is the prediction of the label for observation $\boldsymbol{x}$.
- **Criterion:** for instance, performance of g is measured by the **error probability**:

$$\mathbb{P}\big(g(X) \neq Y\big) = \mathbb{E}\big[\mathbf{1}\big(g(\boldsymbol{x}) \neq Y\big)\big].$$

Building a document classifier?

- n documents available at the "training phase"
- document i $\rightarrow$ a vector $\mathbf{x}_i \in \mathbb{R}^p, i = 1, \ldots, n$
- Label: $y_i \in \{0, 1\}$
- A linear classifier: $f(\mathbf{x}) = s(w_0 + w^T x)$
- with $s(z) = \frac{1}{1+exp(-\frac{1}{2}z)}$, $z \in \mathbb{R}$
- Simple example: minimization of $\mathcal{L}(w; \mathbf{x_1}, \ldots, \mathbf{x_n}) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2$
- Find w such that $\mathcal{L}(w; \mathbf{x_1}, \ldots, \mathbf{x_n})$ be minimal

Outline

Introduction

Introduction to Supervised Learning with hands

Probabilistic and statistical setting of Supervised Learning

- Minimization of the empirical risk

Relevance of Empirical Risk Minimization

References

A probabilistic setting for the learning problem

- Let us call X a random vector that takes its value in $\mathcal{X} = \mathbb{R}^p$
- X describes the properties (we say , features) of the objects to be predicted
- Y a random variable that takes its value in $\mathcal{Y}$: Y encodes some output property
- Let us call p is the joint probability distribution of the random pair (X, Y)
- $\mathcal{Y} = \mathbb{R}$ in case of regression
- $\mathcal{Y} = \{1, -1\}$ in case of binary supervised classification

Risk of a predictive model

(local) loss function:= $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$: for instance, the zero-one prediction loss $\ell(g(x), y) := 1_{y \neq g(x)}$

$$R(g) = \mathbb{E}_{(X,Y)\sim p}[\ell(Y, g(X))]$$

Minimization of the true risk

Imagine for the moment that we do not observe data and that we know ρ.

What is the best solution of the following problem ?

$$\arg \min_{g:\mathcal{X}\to\mathcal{Y}} R(g)$$

This of course depends on the nature of ℓ.

Finding the best binary classification rule

Now $\ell(g(x), y) := 1_{y \neq g(x)}$

$$\arg \min_{g: \mathcal{X} \to \mathcal{Y}} \mathbb{P}(g(X) \neq Y)$$

Note that:

$$\begin{aligned} R(g) &= \sum_{y=-1,1} P(Y = y) \int_{\mathbb{R}^p} \ell_{0,1}(g(x), y) p(x|Y = y) dx \\ &= \sum_{y=-1,1} P(Y = y) \int_{\mathbb{R}^p} 1_{g(x) \neq y} p(x|Y = y) dx \\ &= P(Y = -1) \int_{\mathbb{R}^p} 1_{g(x) \neq -1} p(x|Y = -1) dx + P(Y = +1) \int_{\mathbb{R}^p} 1_{g(x) \neq +1} p(x|Y = +1) dx \end{aligned}$$

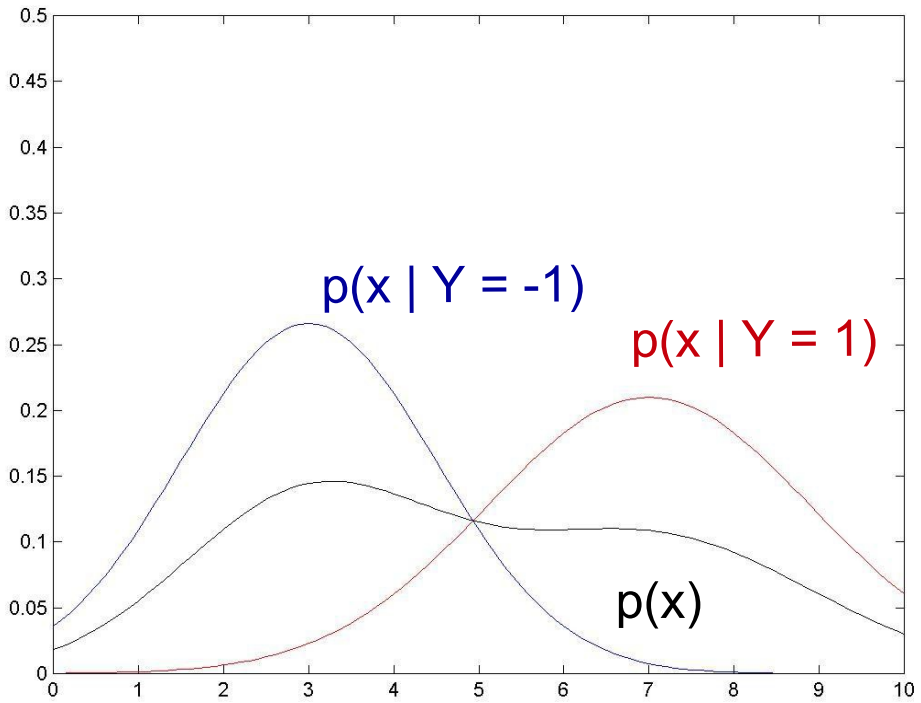
Bayes Rule

Bayes rule

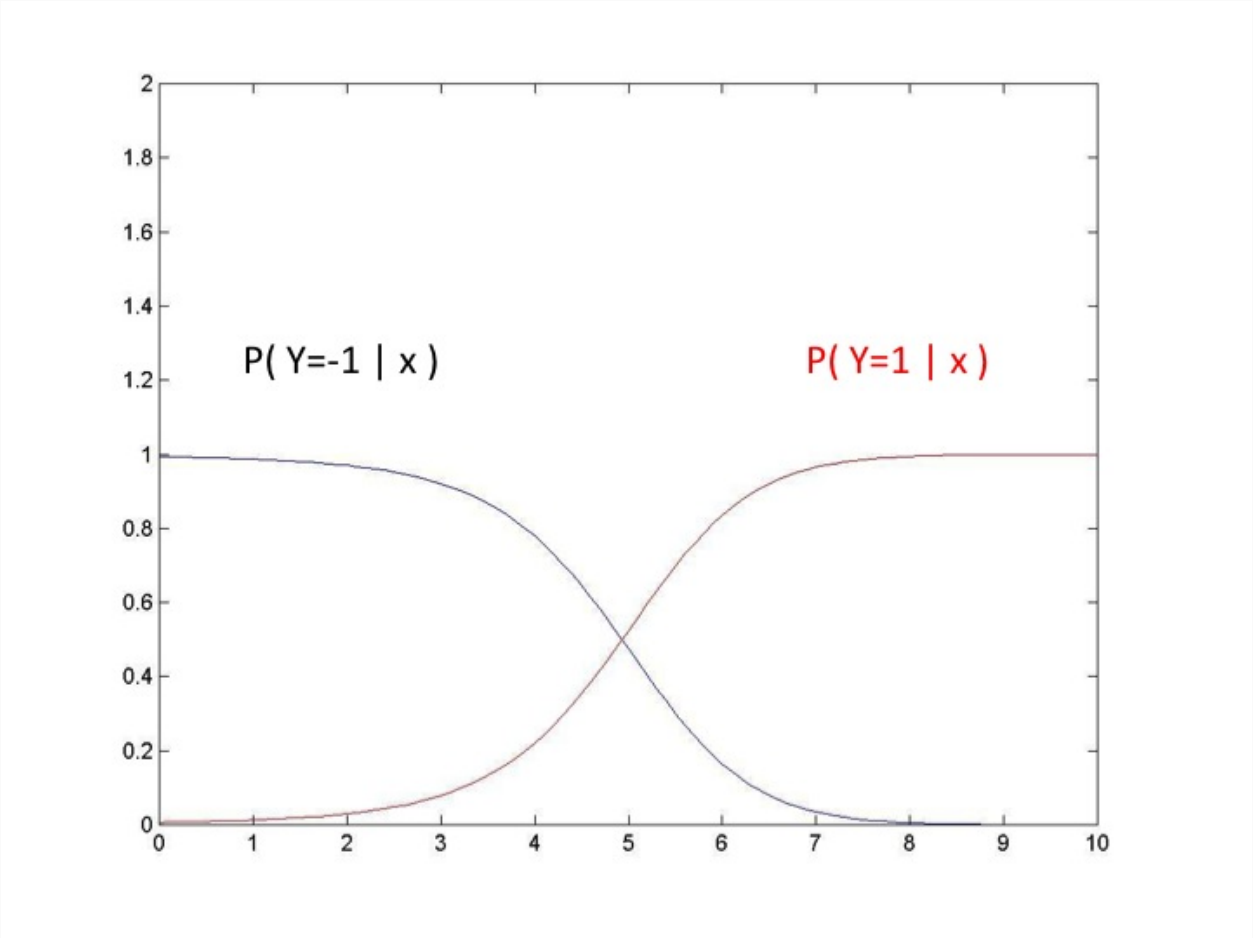
$$P(Y = k|x) = \frac{p(x|Y = k)P(Y = k)}{p(x|Y = -1).P(Y = -1) + p(x|Y = 1).P(Y = 1)}$$

$P(Y = k)$: prior probability
$P(Y = k|x)$: posterior probability of $Y = k$ given x
$p(x|Y = k)$: likelihood or probability density of x conditionally to $Y = k$
Note that $P(Y = 1) + P(Y = -1) = 1$

A 1D example with simple Gaussian probability distribution



Simple Gaussian probability distributions

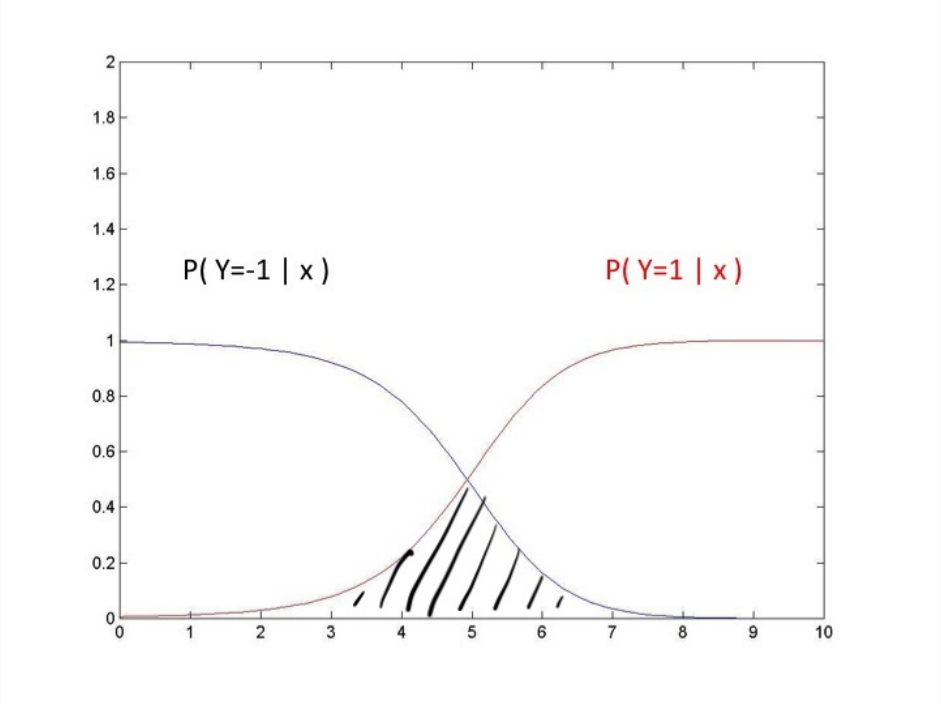


Bayes Classifier

$$g_{Bayes}(x) = \mathbf{1}\big(\mathbb{P}[Y = 1|X = x] > 0.5\big)$$

Here for sake of simplicity, $\mathbf{1}$ gives $+1$ or -1 (indicator function).

Risk of the Bayes classifier: case of 1D Gaussian probability distribution



The Bayes classifier achieves the minimal risk for the classification loss (exercise: prove it)

IMPORTANT ! The Bayes risk is characteristic of the "complexity" of the joint probability distribution P and the **loss**.

First take-home message

- The target function in supervised classification is the Bayes classifier for the $0-1$loss
- The target function in regression is $h(x) = \mathbb{E}[Y|x]$ for the square loss
- More generally, the nature of the target function depends heavily on the loss

Goal of learning:

Find a proxy of the target function g^* using an i.i.d. training sample $\mathcal{S}_n$ without the entire knowledge of p. Go further: see examples of other losses `https://arxiv.org/pdf/1612.03663.pdf`

Statistical supervised learning problem

Supervised learning
In supervised learning we wish to find a classifier (a regressor) in some hypothesis space $\mathcal{G}$ that minimizes

$$R(g) = \mathbb{E}_{(X,Y)\sim p}[\ell(g(X), Y)],$$

without having access to the probability distribution but only a **finite training sample**: $S_n := \{(x_i, y_i)_{i=1}^n\}$ containing n identical independent realizations of (X, Y).

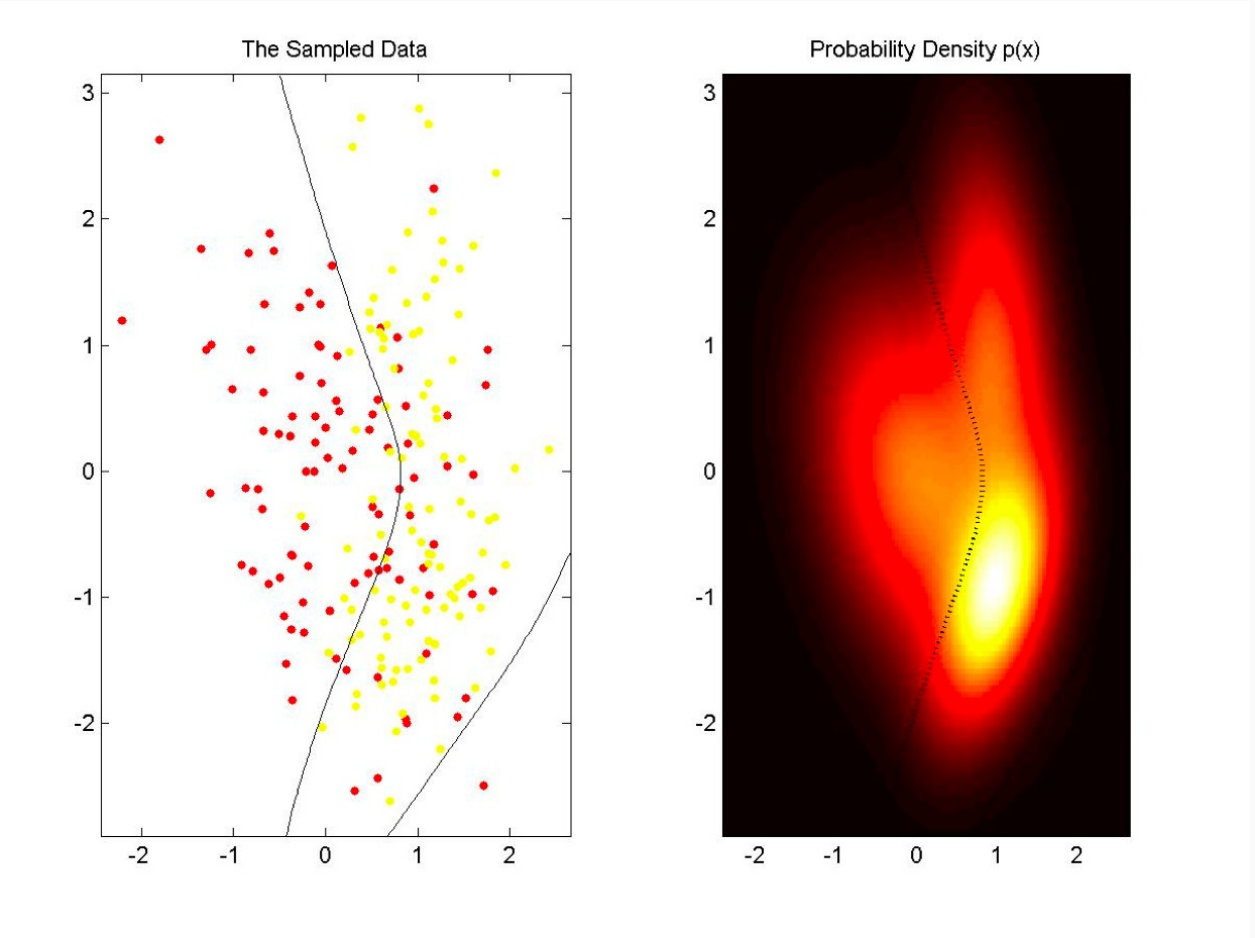
Statistical supervised learning problem: a functional estimation problem

Supervised learning
Suppose there exists g^* a minimizer

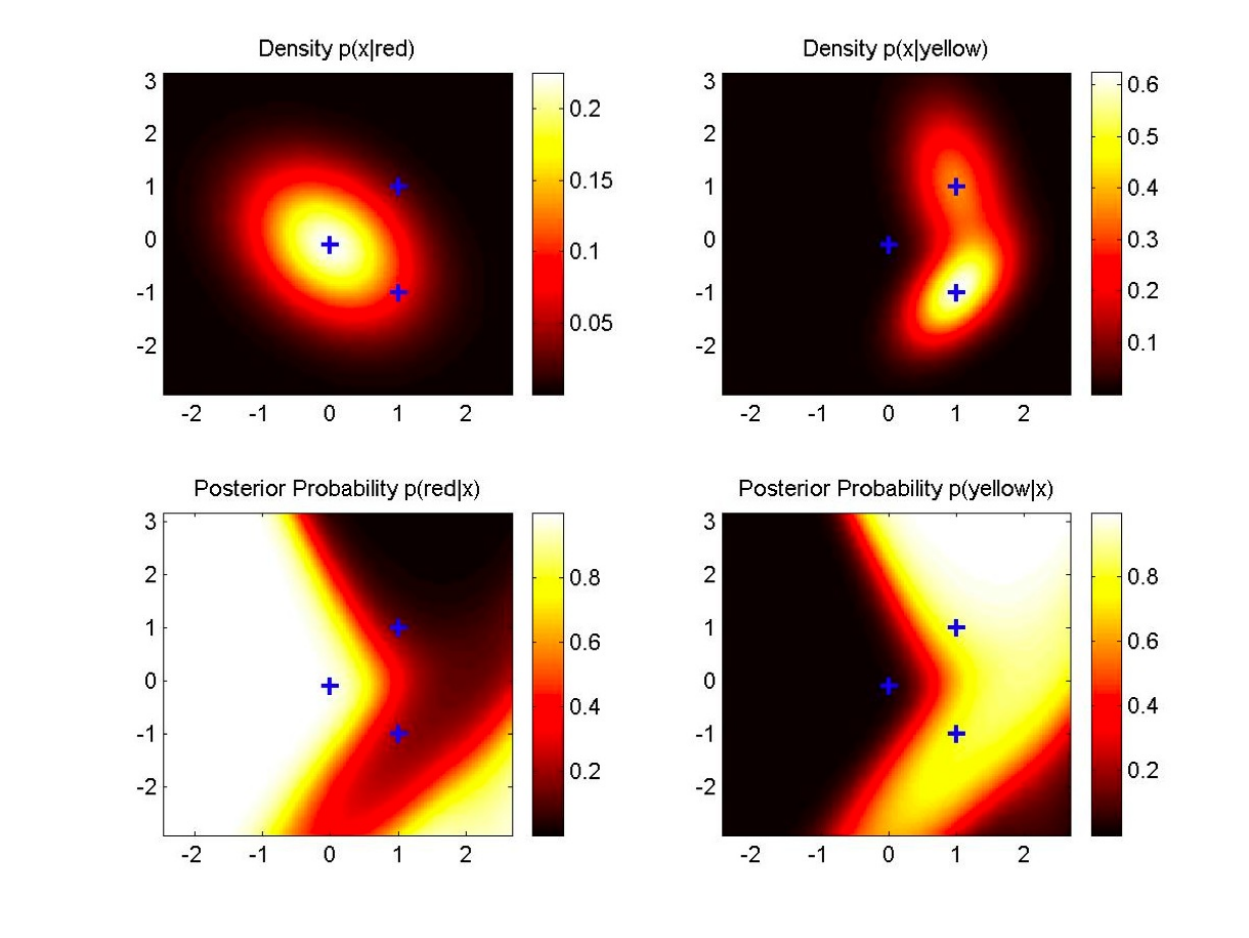
$$g^* \in \arg\min_{g:\mathcal{X}\to\mathcal{Y}} \mathbb{E}_{(X,Y)\sim\rho}[\ell(g(X), Y)]$$

Given a hypothesis space $\mathcal{G}$, **supervised learning** consists in providing an estimate $\hat{h} \in \mathcal{G}$ of g^* leveraging a **finite training sample**:
$S_n := \{(x_i, y_i)_{i=1}^n\}$ containing n identical independent realizations of (X, Y).

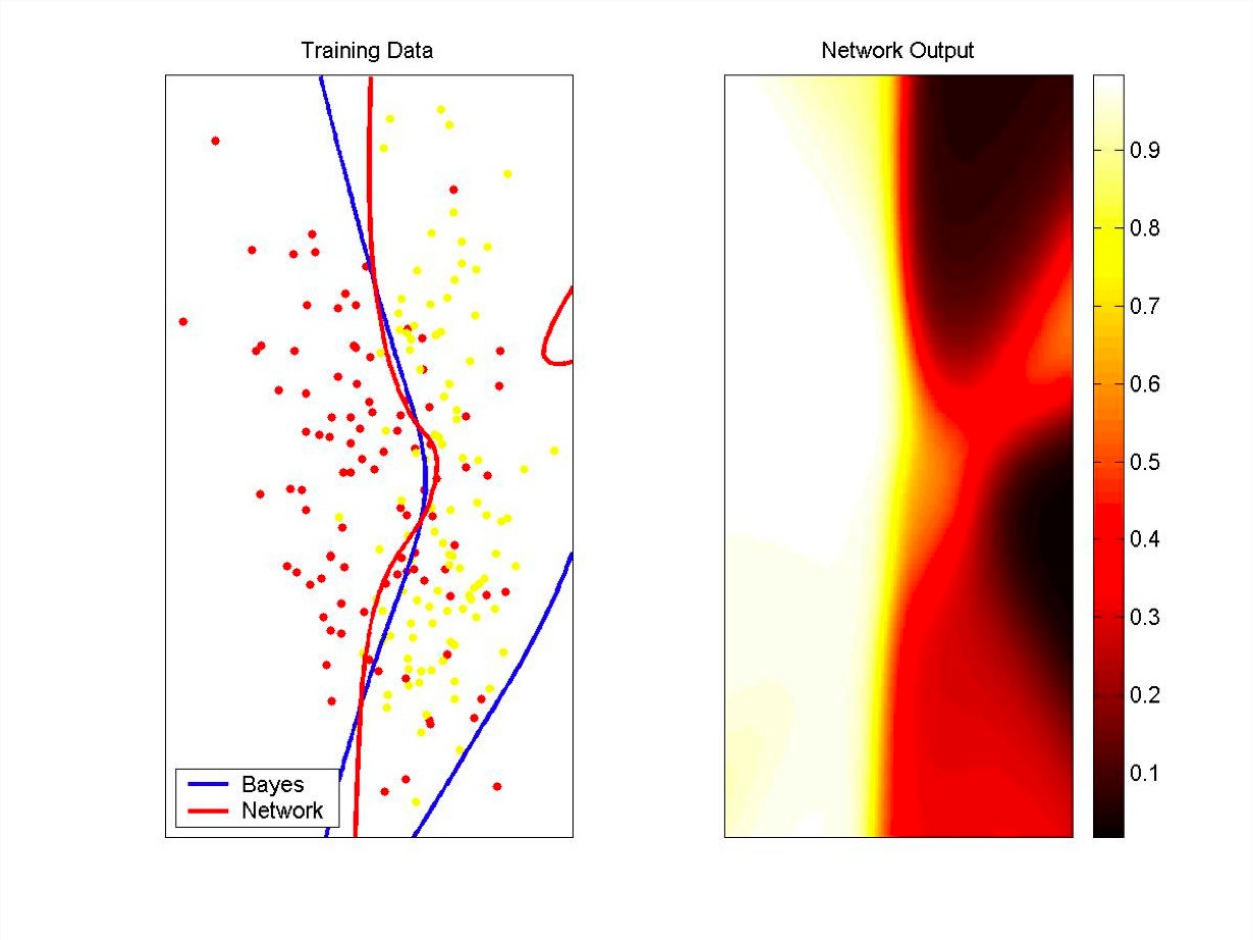
Example in 2D



Now comes the data: example in 2D



Using training set



Minimization of the empirical risk

- Regard a **random pair** (X, Y).
- Consider a **class of** classification **rules** $\mathcal{G}$:
  - one attempts to find **the best rule** in $\mathcal{G}$.
- **Try:** Choose the rule which minimizes a loss function, for example:

$$R(g) = \mathbb{P}\big(g(X) \neq Y\big)\,.$$

- **Problem:** As we cannot calculate $\mathbb{P}$, we cannot calculate $R(g)$ as well.
- **Idea:** Choose a rule that minimizes the empirical version (*i.e.* on the training sample) of the loss function — the **empirical risk**:

$$R_n(g) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{(g(\boldsymbol{x}_i)\neq y_i)}\,.$$

Statistical Machine Learning by Empirical Risk Minimization

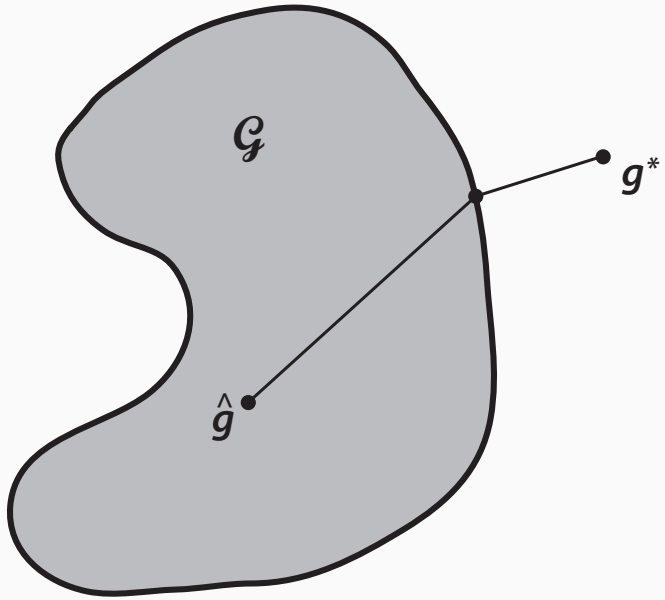
When g is fixed, Law of large numbers : $R_n(g)$ tends towards $R(g)$ almost surely. $(P(lim_n R_n(g) = R(g)) = 1)$;

Statistical Learning by Empirical Risk Minimization

$$\hat{g} \in \arg\min_{g \in \mathcal{G}} R_n(g).$$

Minimization of the empirical risk

Let us denote $R^* = R(g^*)$ (if many minima, the minimum minimorum)



s

Excess risk:

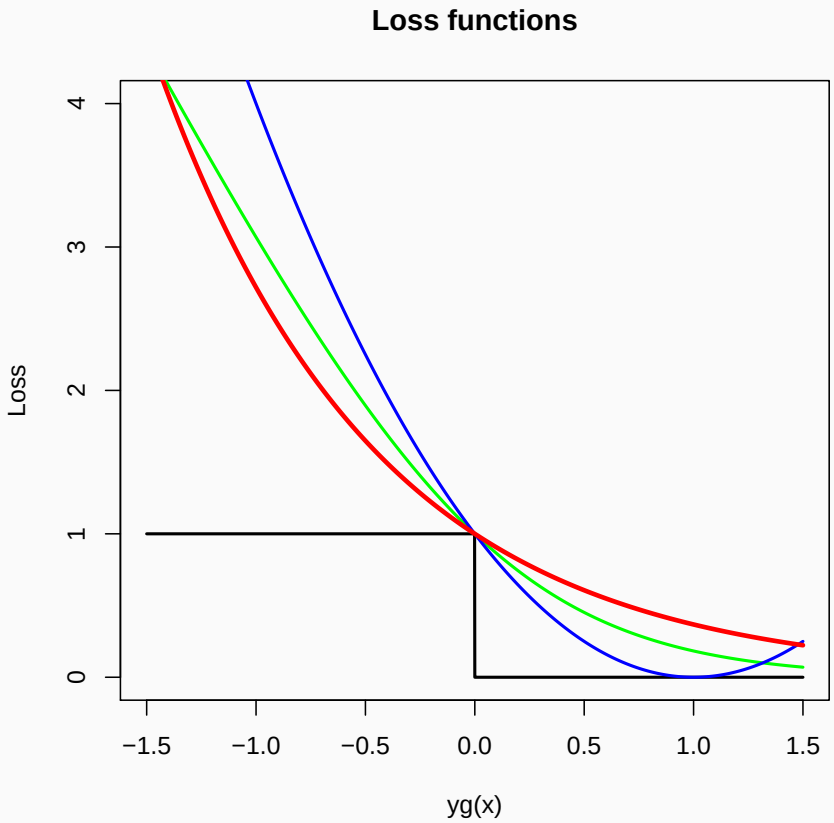
$$R(\hat{g}) - R^* = \underbrace{R(\hat{g}) - \inf_{g \in \mathcal{G}} R(g)}_{\textit{estimation error}} + \underbrace{\inf_{g \in \mathcal{G}} R(g) - R^*}_{\textit{approximation error}} .$$

Risk convexification

(The problem can be raised at the level of true risk minimization as well)

- **Problem:** The function
$$\mathcal{G} \to \mathbb{R}\,, \quad g \mapsto \frac{1}{n}\sum_{i=1}^{n} 1\big(g(\boldsymbol{x}_i) \neq y_i\big)$$
is (usually) **difficult to minimize**.
- **Idea:** Find another loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ such that
$$\mathcal{G} \to \mathbb{R}\,, \quad g \mapsto \frac{1}{n}\sum_{i=1}^{n} \ell\big(g(\boldsymbol{x}_i), y_i\big)$$
is **"easy" to minimize**, *e.g.* a **differentiable** one.
- This is even more the case if the function $u \mapsto \ell(u, v)$ is **convex**.
- The loss function $\ell\big(g(\boldsymbol{x}), y\big)$ should **measure the difference** between the value to be predicted y and $g(\boldsymbol{x})$.
- Thus, *e.g.* for $y \in \{-1, 1\}$, $\ell\big(g(\boldsymbol{x}), y\big)$ should take:
  - **large** values if $yg(\boldsymbol{x}) < 0$,
  - **small** values if $yg(\boldsymbol{x}) > 0$.

Loss functions



— Misclassification:

$$\ell\big(g(\boldsymbol{x}), y\big) = 1\big(yg(\boldsymbol{x}) < 0\big)\,.$$

— Exponential:

$$\ell\big(g(\boldsymbol{x}), y\big) = e^{-yg(\boldsymbol{x})}\,.$$

— Binomial log-likelihood:

$$\ell\big(g(\boldsymbol{x}), y\big) = -\log\big(1+e^{-2yg(\boldsymbol{x})}\big)\,.$$

— Squared error:

$$\ell\big(g(\boldsymbol{x}), y\big) = \big(1 - yg(\boldsymbol{x})\big)^2\,.$$

Summary

- For:
  - a random pair (X, Y),
  - a loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$

  one seeks a classifier close to:

$$g^* = \arg\min_g \mathbb{E}[\ell(g(X), Y)] \, .$$

- **Strategy:** Given a *training sample* $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_n, y_n)$ of (X, Y), one minimizes the empirical version of $\mathbb{E}[\ell(g(X), Y)]$:

$$R_n(g) = \frac{1}{n} \sum_{i=1}^{n} \ell\big(g(\boldsymbol{x}_i), y_i\big) \, .$$

- **Method:** Numerical optimization, for instance, **gradient descent**.
- **Stochastic gradient descent:** Use a single (randomly drawn) observation to iteratively approximate g^*.

Closing the loop - featuring the learning algorithm

Definition

- $\mathcal{S}_n$ is an i.i.d sample of size n, drawn from the joint probability distribution ρ fixed but unknown.
- $\mathcal{S}_n = \{(x_1, y_1), ..., (x_n, y_n)\}$.
- Statistical supervised learning can be defined by:
  - Define a learning algorithm $\mathcal{A} : S_n \rightarrow \mathcal{A}(S_n) \in \mathcal{G}$ such that $\forall \rho$, S_n drawn from ρ, $R(\mathcal{A}(S_n))$ converges towards $R(g*)$ in probability

Outline

Introduction

Introduction to Supervised Learning with hands

Probabilistic and statistical setting of Supervised Learning

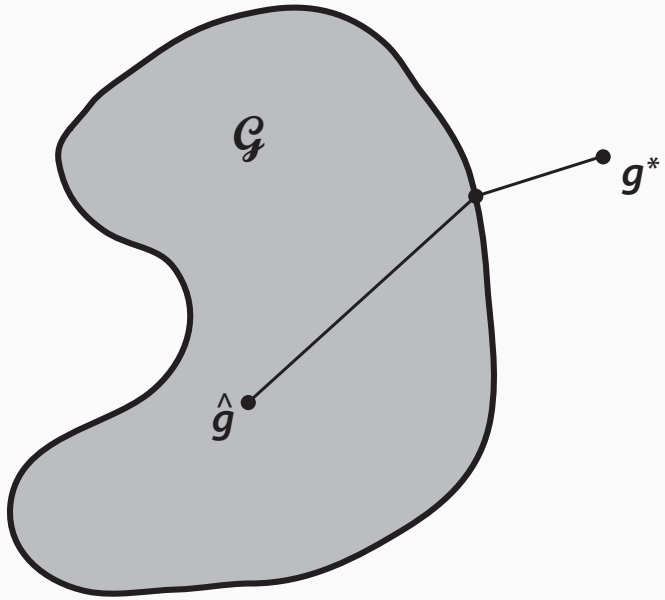
Minimization of the empirical risk

Relevance of Empirical Risk Minimization

References

Minimization of the empirical risk

Let us denote $R^* = R(g^*)$ (if many minima, the minimum minimorum)



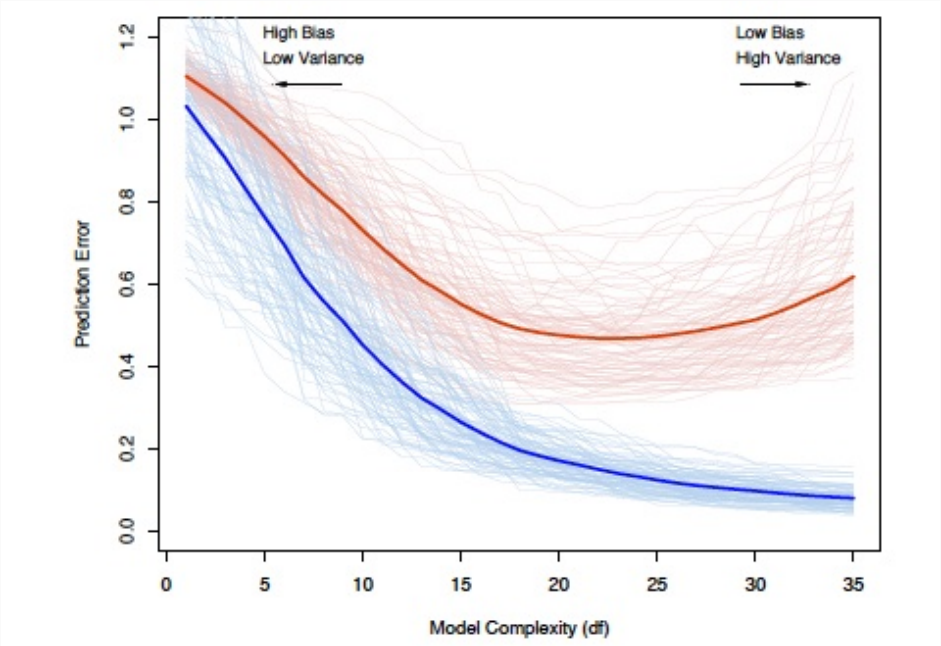
s

Excess risk:

$$R(\hat{g}) - R^* = \underbrace{R(\hat{g}) - \inf_{g \in \mathcal{G}} R(g)}_{\textit{estimation error}} + \underbrace{\inf_{g \in \mathcal{G}} R(g) - R^*}_{\textit{approximation error}} .$$

Bias-variance dilemna

Experimental study



How to choose $\mathcal{H}$?

A compromise bias/variance
Given a fixed training sample:

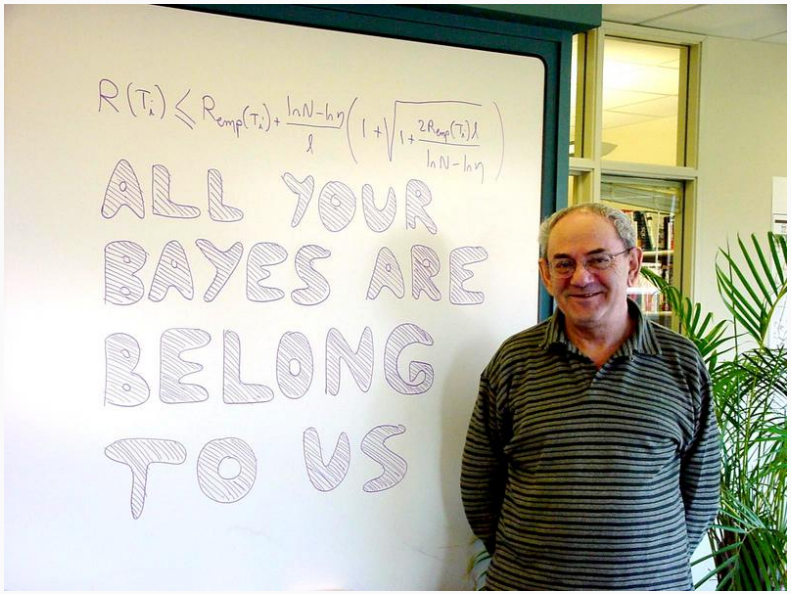
- If model complexity is too low, you cannot reach the target (large bias, no universality) : risk of UNDERFITTING
- If model complexity is too big, you cannot reduce variance (large variance, no consistency) : risk of OVERFITTING

Is empirical risk minimization meaningful ?

Vapnik and Chervonenkis's results

- $\forall \mathbb{P}, \mathcal{S}_n$ drawn from P, $\forall h \in \mathcal{H}, R(h) \leq R_n(h) + \mathcal{B}(d, n)$
- where d is a measure of complexity of $\mathcal{H}$

Generalization bounds



Vladimir Vapnik in front of a white board, claiming for statistical learning against Bayesian inference

Question: learning guarantee
If we measure the empirical risk $R_S(h)$ associated to a classifier h, what can we say about its true risk $R(h)$?

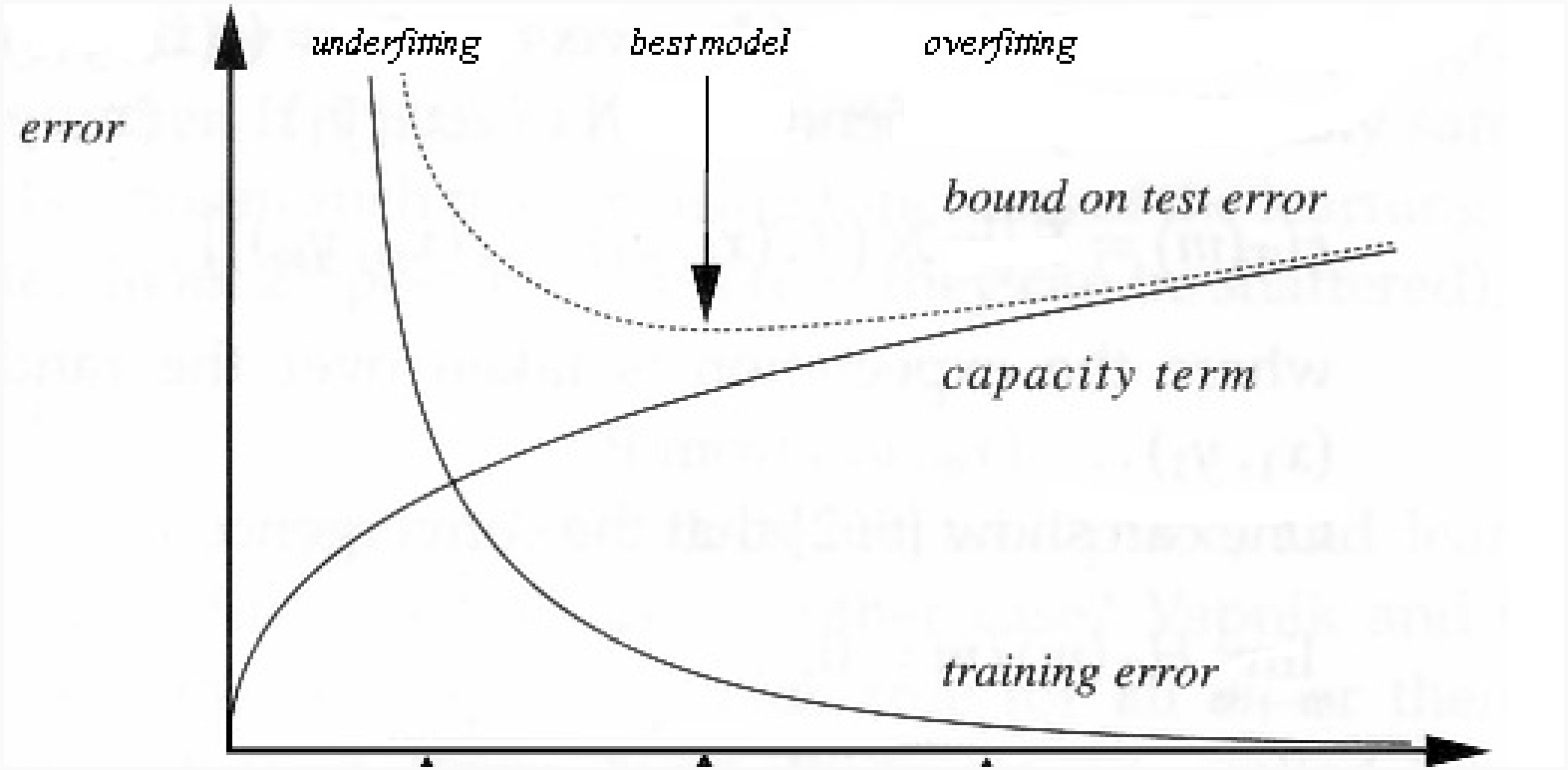
Read more: ▸ Link towards a small tutorial with proof

VC-dimension generalization bounds

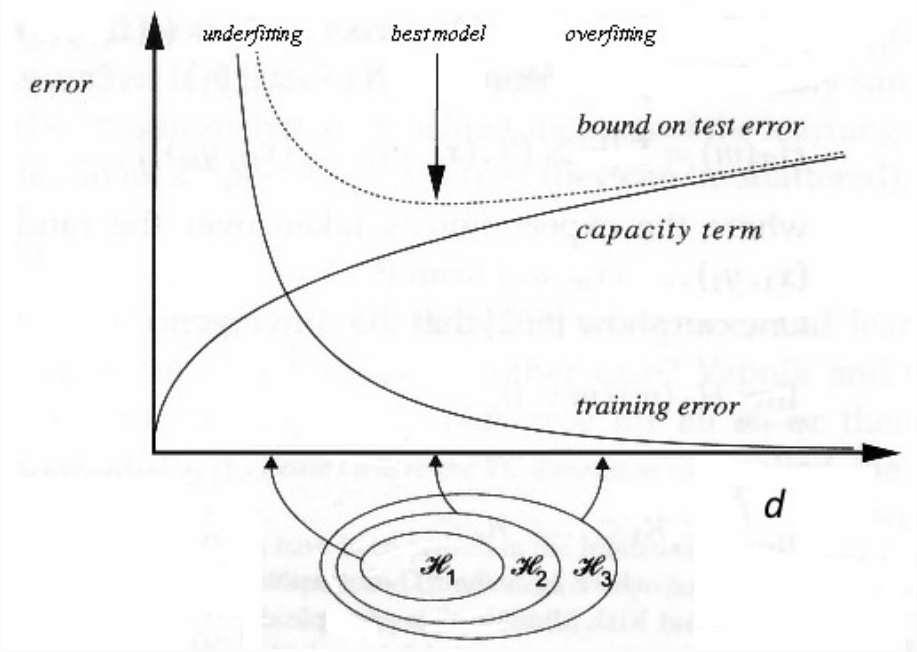
Theorem:
Let $\mathcal{H}$ be a family of functions taking values in $\{-1, +1\}$ with VC-dimension d_{VC}. Then, for any $\delta > 0$, the following holds for all $h \in \mathcal{H}$ with probability greater than 1 - δ

$$R(h) \leq R_n(h) + \sqrt{\frac{8 d_{VC}(\ln \frac{2n}{d_{VC}} + 1) + 8 \log(\frac{4}{\delta})}{n}}$$

Error (risk) versus h



Principle of Structural Risk Minimization



Vapnik proposed to replace empirical minimization principle by structural risk minimization, the underlying idea is to control the complexity of family $\mathcal{H}$ while reducing the empirical error.

Shattering

Definition: **Shattering**

$\mathcal{H}$ is said to shatter a set of data points $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ if, for all the 2^n possible assignments of binary labels to those points, there exists a function $h \in \mathcal{H}$ such that the model h makes no errors when predicting that set of data points.

Vapnik-Chervonenkis dimension

Definition: **VC-dimension**

The VC-dimension of a hypothesis set $\mathcal{H}$ is the size of the largest set that can be fully shattered by $\mathcal{H}$:

$$d_{VC}(\mathcal{H}) = max\{m : \exists(\mathbf{x}_1, \ldots, \mathbf{x}_m) \in \mathcal{X}^m \text{that are shattered by } \mathcal{H}\}$$

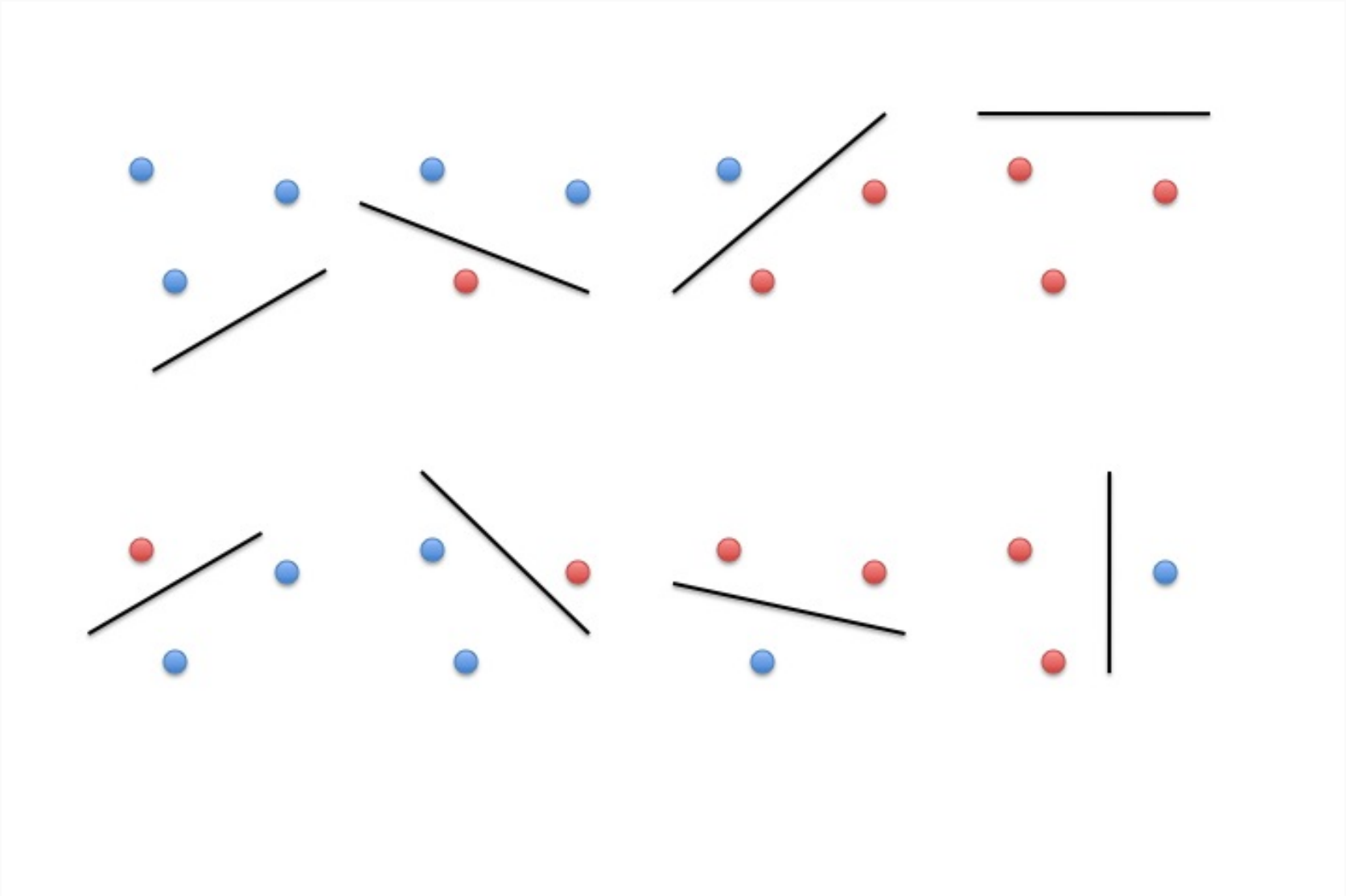
N.B.: if $d_{VC}(\mathcal{H}) = d$, then there exists a set of d points that is fully shattered by $\mathcal{H}$, but this DOES NOT imply that all sets of dimension d or less are fully shattered !

VC-dimension of Hyperplanes

What is the VC-dimension of hyperplanes in $\mathbb{R}^2$ (denoted $\mathcal{H}_2$) ?
Obviously $d_{VC}(\mathcal{H}_2) \geq 2$
Let us try with 3 points :

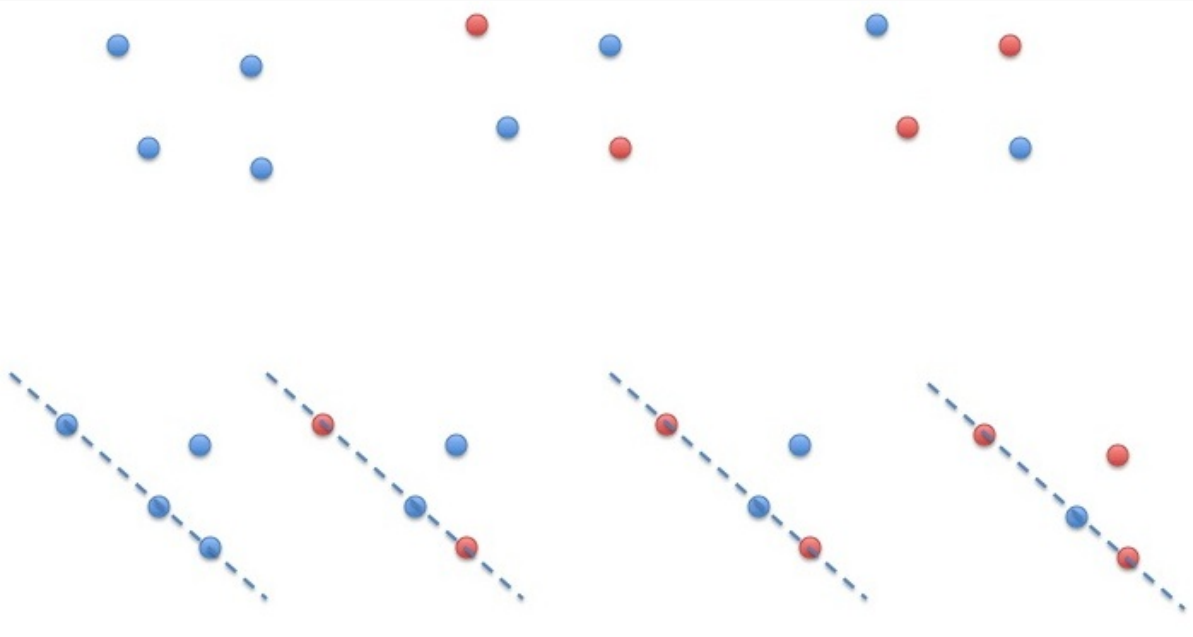
VC-dimension of Hyperplanes

What is the VC-dimension of hyperplanes in $\mathbb{R}^2$ (denoted $\mathcal{H}_2$) ?
Let us consider the following triplet of points



VC-dimension of Hyperplanes

What is the VC-dimension of hyperplanes in $\mathbb{R}^2$ (denoted $\mathcal{H}_2$) ?
For any set of 4 points, either 3 of them (at least) are aligned or no triplet of points is aligned.



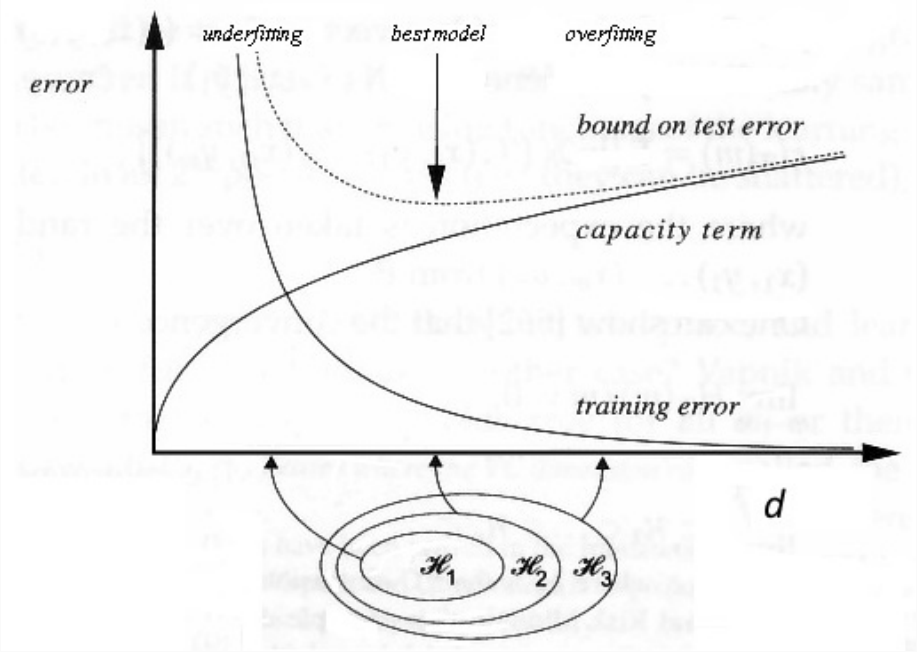
We can show that it is not possible for $\mathcal{H}_2$ to shatter 4 points.
Then $d_{VC}(\mathcal{H}_2) = 3$.

VC-dimension of Hyperplanes

More generally, one can prove :

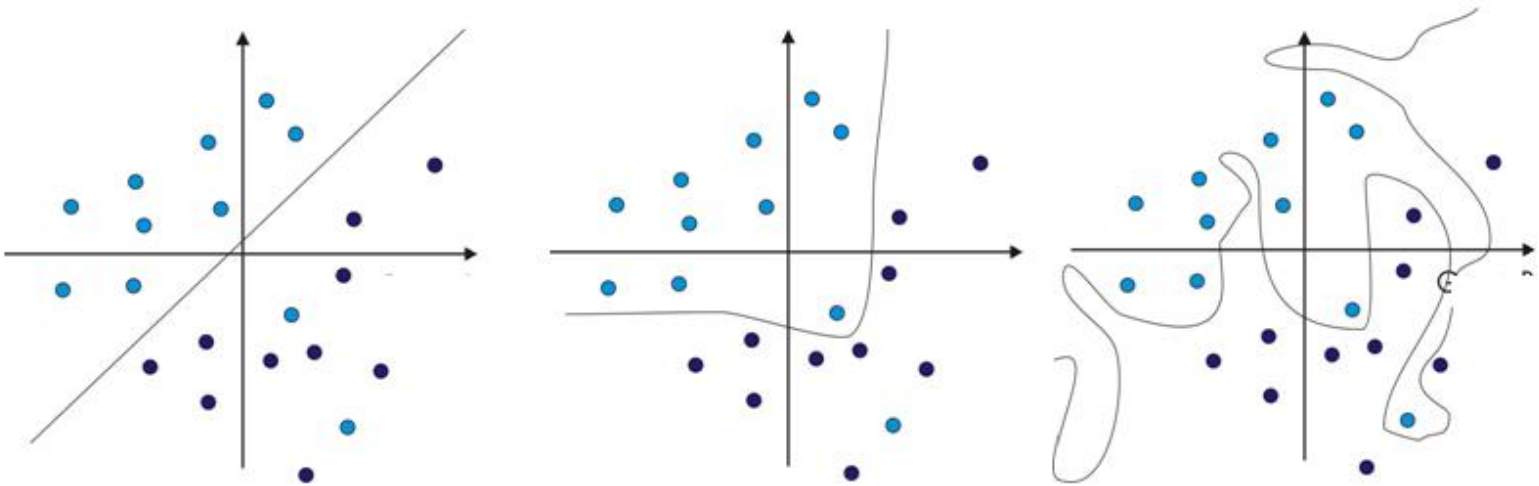
$$d_{VC}(\mathcal{H}_d) = d + 1$$

Principle of Structural Risk Minimization



Vapnik proposed to replace empirical minimization principle by structural risk minimization, the underlying idea is to control the complexity of family $\mathcal{H}$ while reducing the empirical error.

In practice, how to avoid overfitting



Optimization problem in practice: regularization

Pb1
$Min_h R_n(h)$ s.c $\Omega(h) \leq C$

Pb2
$Min_h \Omega(h)$ s.c $R_n(h) \leq C$

Pb3
$Min_h R_n(h) + \lambda \Omega(h)$

- Ω(h): measures the complexity of a single function h

Machine Learning with regularization

Supervised Learning

Let $\mathcal{S}_n = \{(x_i, y_i)\}_{i=1}^n$, a i.i.d. sample drawn from p 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 $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$, $\Omega : \mathcal{H} \to \mathbb{R}^+$, the goal is now to find a solution of:

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

Role of $\Omega(h)$: control of the model complexity, more generally imposition of some prior knowledge

Complexity regularization

- When performing machine learning, we are in effect building **a non-linear model** of the physical phenomenon responsible for the generation of the input-output examples used to train the model.
- As the model design is statistical in nature, we need an appropriate **tradeoff** between **reliability** of the training data and **goodness** of the model.
- We may realize this tradeoff by minimizing the **total risk**, expressed as a function of the parameter vector h, as follows:

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

- $\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i))$ is the standard **performance metric**, which depends on both the model and the input data, and during learning can be defined as, *e.g.*, a mean-square error.
- $\Omega(h)$ is the **complexity penalty**, where the notion of complexity is measured in terms of the model's parameters (weights) alone.
- λ is a **regularization parameter**.

Back to Support vector Machines

We solve:

$$\hat{h} := \min_{h \in \mathcal{H}_k} \frac{1}{n} \sum_{i=1}^{n} \underbrace{\max(0, 1 - y_i(h(x_i) + b))}_{hinge\ \ loss} + \lambda \|h\|^2_{\mathcal{H}_k}$$

and we get:

$$\hat{g}(x) := \text{sign}(\hat{h}(x) + b) = \text{sign}(\sum_{i=1}^{n} y_i \alpha_i k(x, x_i) + b)$$

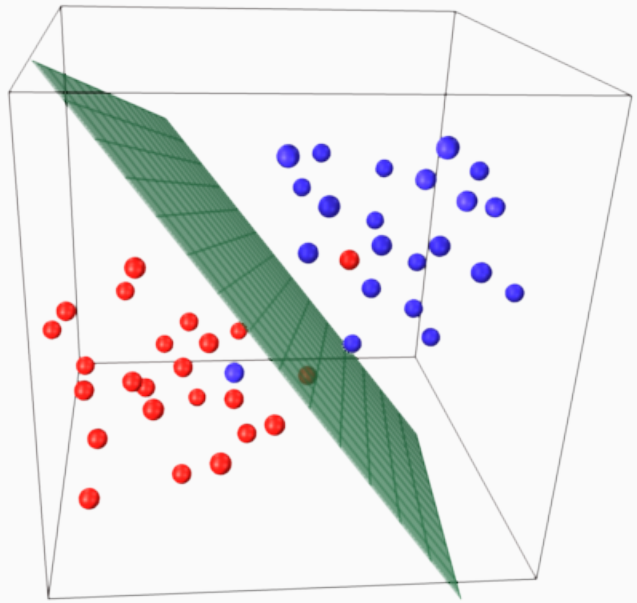
Machine Learning: two tasks

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

- **Learning**: get $g_n = \mathcal{A}(\mathcal{S}_n, \mathcal{H}, \ell, \lambda, \Omega)$ 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
- **Prediction**: given x, and compute $g_n(x)$

Machine Learning: key components

- Data representation
- Hypothesis space
- Loss function
- Learning algorithm
- Evaluation metrics
- Model selection



Example of supervised learning

Outline

Introduction

Introduction to Supervised Learning with hands

Probabilistic and statistical setting of Supervised Learning

- Minimization of the empirical risk

Relevance of Empirical Risk Minimization

References

Bibliography

- Vapnik (1998): *Statistical Learning Theory*. John Wiley & Sons.
- Bishop (1999): *Pattern Recognition and Neural Networks*, Springer.
- Hastie, Tibshirani, Friedman (2001): *The Elements of Statistical Learning*, Springer.
- Haykin (2009): *Neural Networks and Learning Machines*, Pearson.
- Abu-Mostafa, Magdon-Ismail, Lin (2012): *Learning from Data: A Short Course*.
- James, Witten, Hastie, Tibshirani (2013): *An Introduction to Statistical Learning*. Springer.
- Bertsekas, (2016): *Nonlinear Programming*. Athena Scientific.
- Goodfellow, Bengio, Courville (2016): *Deep Learning*, MIT Press.
- Mohri, Rostamizadeh, Talwalkar (2018): *Foundations of Machine Learning*. MIT Press.
- Bach (2024): *Learning Theory from First Principles*, MIT Press.