Basics of Machine Learning

SD 210 - P3

Lecture 1 - Introduction to Machine Learning

Florence d'Alché-Buc

Contact: florence.dalche@telecom-paristech.fr,
2A Filière SD, Télécom ParisTech,Université of Paris-Saclay, France

Table of contents

Outline

Overview

- Motivation
- Towards a definition of ML
- Statistical learning

Statistical supervised learning

Perceptron

References

Outline

Overview

- Motivation
- Towards a definition of ML
- Statistical learning

Statistical supervised learning

- Relevance of empirical risk minimization

Perceptron

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 Monte-Carlo Tree search + 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

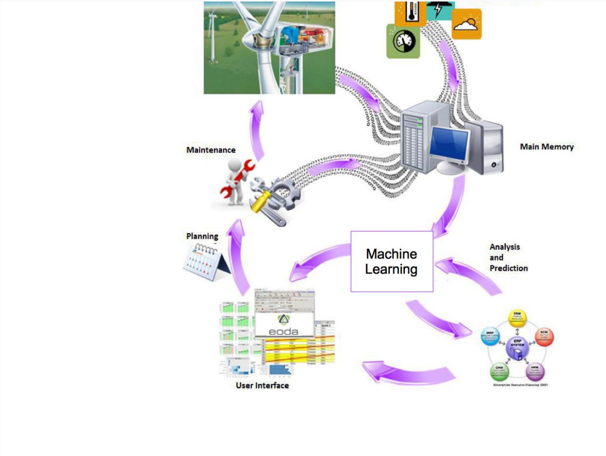
Deep-learning software excels at complex ancient board game.

▸ Readmore

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

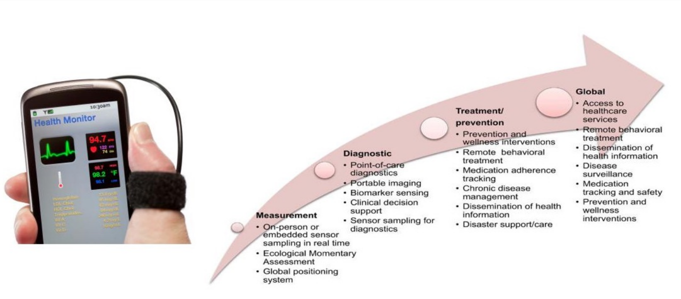
Predictive Maintenance

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.



Read more: `http://www.itproportal.com/2015/08/01/predictive-analytics-will-shape-future-every-sector/#ixzz3zloPifwW`

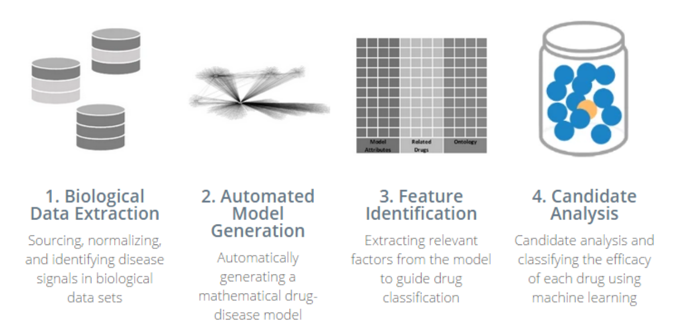
Mobile health monitoring


10:30am
Health Monitor
94.7
98.2 °F
Measurement
• On-person or embedded sensor sampling in real time
• Ecological Momentary Assessment
• Global positioning system
Diagnostic
• Point-of-care diagnostics
• Portable imaging
• Biomarker sensing
• Clinical decision support
• Sensor sampling for diagnostics
Treatment/ prevention
• Prevention and wellness interventions
• Remote behavioral treatment
• Medication adherence tracking
• Chronic disease management
• Dissemination of health information
• Disaster support/care
Global
• Access to healthcare services
• Remote behavioral treatment
• Dissemination of health information
• Disease surveillance
• Medication tracking and safety
• Prevention and wellness interventions

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.



1. Biological Data Extraction

Sourcing, normalizing, and identifying disease signals in biological data sets

2. Automated Model Generation

Automatically generating a mathematical drug-disease model

3. Feature Identification

Extracting relevant factors from the model to guide drug classification

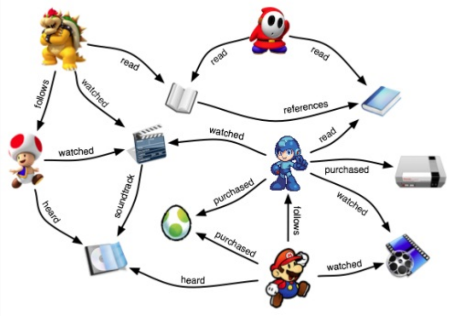
4. Candidate Analysis

Candidate analysis and classifying the efficacy of each drug using machine learning

Read more: ▸ Link

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

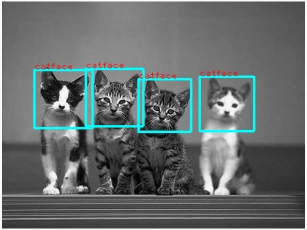
Recommendation system



- "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.

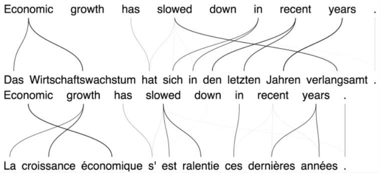
Object recognition



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

and ▸ Link 2 for instance: website of Ivan Laptev

Machine Translation


Economic growth has slowed down in recent years .
Das Wirtschaftswachstum hat sich in den letzten Jahren verlangsamt .
Economic growth has slowed down in recent years .
La croissance économique s' est ralentie ces dernières années .

Read more: ▸ Link

Introduction to Neural Machine Translation with GPUs. Kyunghyun Cho.

Machine Learning everywhere !

Use data to extract a prediction function

- Search engine, text-mining
- Diagnosis, Fault detection
- Business analytics
- Prediction in Heath care, Personalized medecine
- Social networks, link prediction, recommendation

Outline

Overview

Motivation

Towards a definition of ML

Statistical learning

Statistical supervised learning

Relevance of empirical risk minimization

Perceptron

References

A definition of Machine Learning

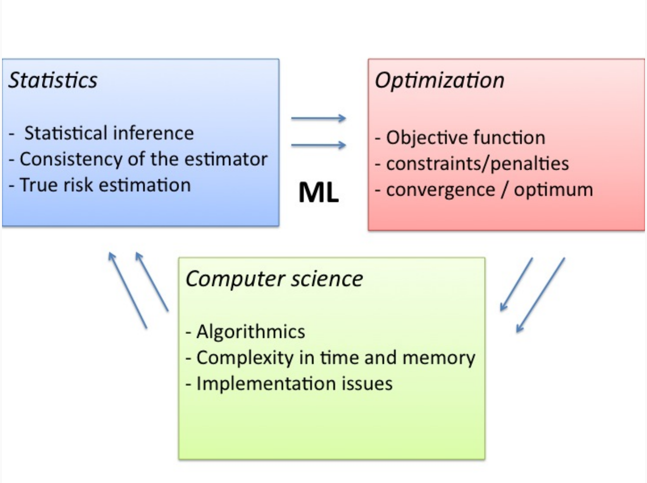
A type of artificial intelligence (AI) that provides computers with the ability to do certain tasks, such as recognition, diagnosis, planning, robot control, prediction, etc., without being explicitly programmed. It focuses on the development of algorithms that can teach themselves to grow and change when exposed to new data.

Experience, tasks and performance measure

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 modelling, game player, robot learning,...
- **Performance measure** : accuracy on new data, ability to generalize

Machine Learning



Statistics

- Statistical inference
- Consistency of the estimator
- True risk estimation

ML

Optimization

- Objective function
- constraints/penalties
- convergence / optimum

Computer science

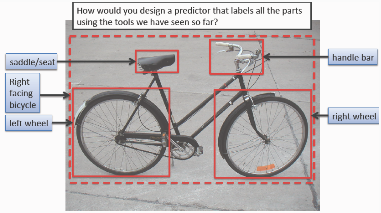
- Algorithmics
- Complexity in time and memory
- Implementation issues

Example 1: a robot that learns

Robot endowed with a set of sensors and a online learning algorithm:

- Sense the environment, act and measure the effect of action
- Goal: play football

Example 2 : object recognition in an image


How would you design a predictor that labels all the parts using the tools we have seen so far?

- Read a data file
- Recognize if parts of the target object are present
- **Goal**: say if an object is present or not in the image.

Two kinds of learning 1/2

Online learning: *the learning algorithm keeps on interacting with the environment*

- robotics
- predictive maintenance
- security in cloud servers
- personalized advertising
- autonomous cars
- personalized healthcare
- security systems

Two kinds of learning 2/2

Offline or batch learning: *the learning algorithm gets a datafile and outputs some function that can be used in turn to new data*

- pattern recognition (a wide panel of applications)
- diagnosis (health, plants)
- link prediction in networks
- data-mining
- social networks analytics

This course: **mainly batch learning.**

Outline

Overview

- Motivation
- Towards a definition of ML
- Statistical learning

Statistical supervised learning

- Relevance of empirical risk minimization

Perceptron

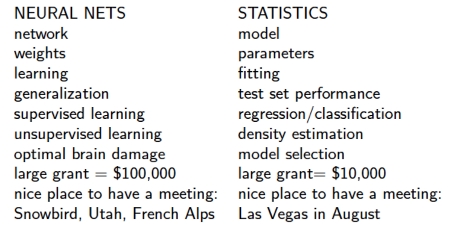
References

Machine learning : statistical or symbolic learning

In the 80's, there was a debate about how to address Machine Learning.

- Symbolic learning
  - Associated to Artificial Intelligence : about logical inference
  - **Goal of learning**: learn logical rules that are consistent (in a logical sense) with observed facts and given rules
  - Interest: interpretability $\neq$ a black box
- Numerical Learning /Connexionism (at this stage, statistical learning was not really born)
  - **Goal of learning**: learn weights (parameters) of neural networks to fit observed data
  - Interest: robustness to noise, efficiency of learning (stochastic gradient descent)
  - At the end of the 90's, connexionism has been replaced by **statistical learning**, giving a more general picture, conciliating machine learning with statistical inference

Slide form Rob Tibshirani (early 90's)

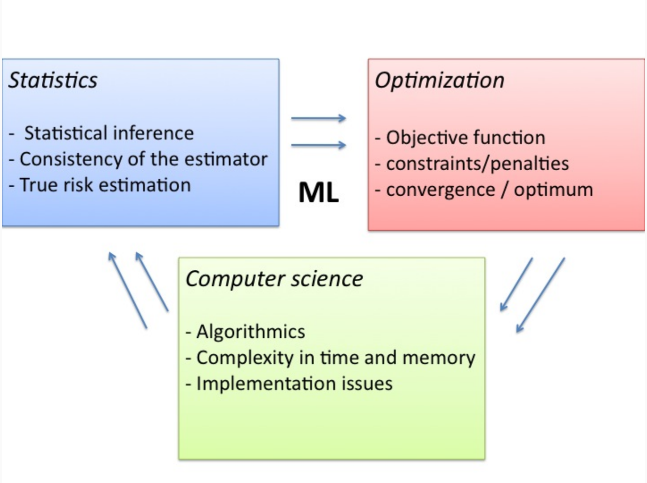


NEURAL NETS	STATISTICS
network	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation
optimal brain damage	model selection
large grant = $100,000	large grant= $10,000
nice place to have a meeting:	nice place to have a meeting:
Snowbird, Utah, French Alps	Las Vegas in August

Statistical Machine Learning

- we build learning algorithms: our algorithms provide estimators
- we are interested on some statistical properties like consistency of the estimators
- but also on the efficiency of the algorithms as optimization procedures.

Statistical Machine Learning



Statistics

- Statistical inference
- Consistency of the estimator
- True risk estimation

ML

Optimization

- Objective function
- constraints/penalties
- convergence / optimum

Computer science

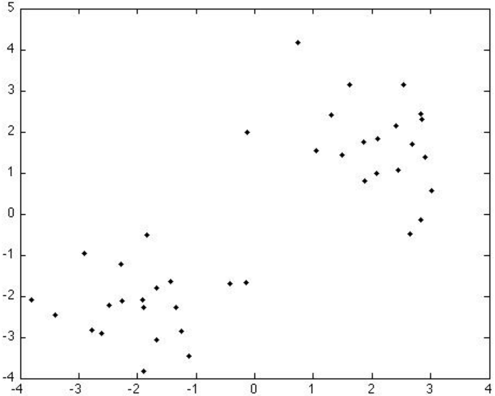
- Algorithmics
- Complexity in time and memory
- Implementation issues

Supervised versus unsupervised learning

- **Supervised Learning (classification, regression)**:
  - Goal: Learn a function f to predict a variable y from an individual x.
  - Data: Learning set (x_i, y_i)
- **Unsupervised Learning (clustering, graphical model)**:
  - Goal: Discover a structure within a set of of individuals $\{x_i\}$.
  - Data: Training set $\{x_i\}$
- First case is better posed.
- Note: most of these algorithms can be implemented offline or online.

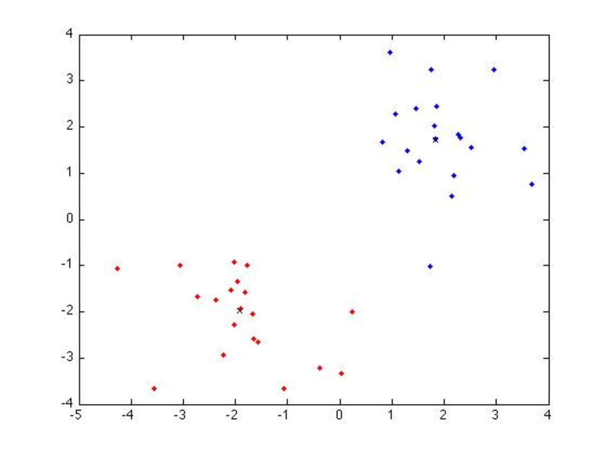
Example of clustering in 2D

Here are the data:

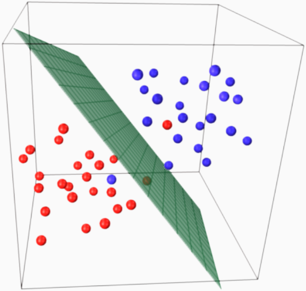


Example of clustering in 2D

Here are the data:



Example of supervised classification in 2D



Outline of this course

1. Feb 10: Introduction to Statistical Machine Learning (3h lecture)
2. Feb 24: Introduction to machine methodology/ first classifiers (featuring perceptron) - (1h30 lecture)
3. Feb 24: Practical session 1 (1h30)
4. March 3 : Support Vector Machine and kernels (3h lecture)
5. March 10 : Practical session 2 (3h)
6. March 24 : Decision and regression trees - (1h30) Lecturer: Pietro Gori
7. March 24 : Practical session 3 - (1h30)
8. March 31 : Bagging and Random Forests - (1h30)
9. March 31 : Practical session 34 - (1h30)
10. April 6 : different date, a thursday : introduction to clustering, mixture models (3h) Lecturer: Umut Simsekli
11. April 7: Practical session (1h30)
12. April 24: exam (1h30)

Teaching team

Lecturers

- Pietro Gori, PhD, assistant prof. (Image processing/ Machine Learning)
- Umut Simsekli, PhD, assistant prof. (Bayesian learning) also in charge of the challenge
- Florence d'Alché, prof. (Machine learning, kernels, structured outputs)

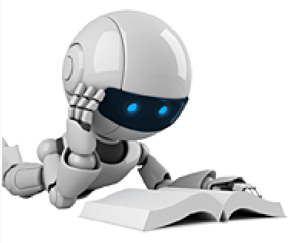
Teaching Assistants

- Moussab Djerrab, PhD student (Machine Learning)
- Pietro Gori, assistant prof. (Image processing/ Machine Learning)
- Alexandre Garcia, PhD student (Machine Learning)
- Balamurugan Palaniappan, phD, postdoc (Machine Learning)

How to work for this course ?

- The program is long compared to the scheduled courses
- You will need to **read and learn by yourselves** to complete the scheduled hours
- It is mandatory to code and run machine learning programs: be careful about black boxes (always ask what is in the box)

How to work for this course ?



Several books/sources can help you

1. The elements of statistical learning, Hastie, Tibshirani, Friedman, Springer (free pdf).
2. Pattern recognition and machine learning, Chris Bishop
3. Several video-lectures and online courses
4. Machine Learning Meetup in Paris every month ▸ ML Meetup

Evaluation

- 2 practical sessions will be evaluated: 5 pts
- A challenge to address: 7 pts
- An exam (a list of questions): 8 pts

Outline

Overview

- Motivation
- Towards a definition of ML
- Statistical learning

Statistical supervised learning

- Relevance of empirical risk minimization

Perceptron

References

Outline

Overview

Statistical supervised learning

- Relevance of empirical risk minimization

Perceptron

References

Outline

Overview

- Motivation
- Towards a definition of ML
- Statistical learning

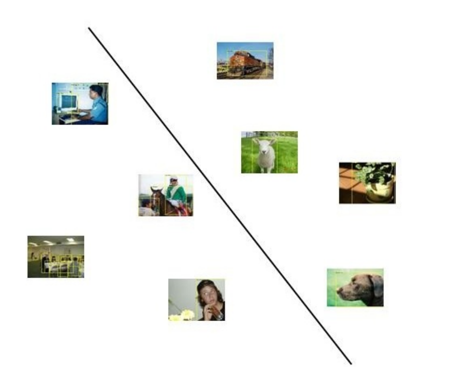
Statistical supervised learning

- Relevance of empirical risk minimization

Perceptron

References

Goal of Supervised classification



- Build a software that automatically classify data into two classes
- Images with human versus images with no human

Use training dataset to define the classifier

Computer science/algorithmics

- Training dataset: $\mathcal{S}_n = \{(image, label)\} = \{(x_i, y_i), i = 1, \ldots, n\}$
- Define an algorithm $\mathcal{A}$ that takes the training dataset and provide a function that classifies the data
- At the end, two pieces of code:
  - a program that implements $\mathcal{A}$
  - a program that makes a prediction given some input (here an image)

Use training dataset to define the classifier

Statistical Inference

- Training dataset: $\mathcal{S}_n = \{(image, label)\} = \{(x_i, y_i), i = 1, \ldots, n\}$
- Define an estimation procedure that takes the training dataset and provide a function that classifies the data
- At the end,
  - What are the properties of the estimator ? Consistency ....
  - What error does make the estimated function ?

Use training dataset to define the classifier

Mathematical programming / optimization

- Training dataset: $\mathcal{S}_n = \{(image, label)\} = \{(x_i, y_i), i = 1, \ldots, n\}$
- Define an optimization problem and then an optimization algorithm that takes the training dataset and provide a function that classifies the data
- At the end,
  - What are the properties of the optimization algorithm ? convergence towards a global/local minimum...
  - Complexity in time ... Complexity in memory...

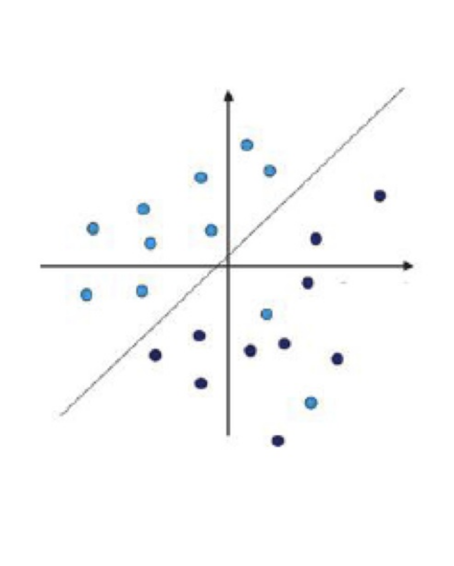
What do we need to determine an image classifier?

- Choose a way to represent image
- Choose a family of classification functions
- Formulate the learning problem as an optimization one
- Define an optimization algorithm
- Evaluate the quality of the classifier learnt from data

What do we need to determine an image classifier?

- Choose a way to represent image (the input) : x: a grey-level matrix as a huge vector
- output : y : 0 or 1
- A classifier: linear or nonlinear ?
- Learning algorithm : minimizes some cost function
- Empirical measures: accuracy/ classification error, test error, Cross-validation

Linear classifier



Building an image classifier?

- n images
- Image i → a vector $\mathbf{x}_i \in \mathbb{R}^p$,$i = 1, \ldots, n$
- Label: $y_i \in \{0, 1\}$
- A linear classifier:$h(\mathbf{x}) = s(w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_p x_p)$
- with $s(z) = \frac{1}{1+exp(-\frac{1}{2}z)}$, $z \in \mathbb{R}$
- Simple example: minimization of $\mathcal{L}(w; \mathbf{x_1}, \ldots, \mathbf{x_n}) = \frac{1}{n} \sum_{i=1}^{n} (h(\mathbf{x}_i) - y_i)^2$
- Find w such that $\mathcal{L}(w; \mathbf{x_1}, \ldots, \mathbf{x_n})$ be minimal

A probabilistic view of the learning problem (1): no data !

- Let's 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
- Y a random variable that takes its value in $\mathcal{Y}$: Y encodes some output property
- $\mathcal{Y} = \mathbb{R}$ in case of regression
- $\mathcal{Y} = \{1, -1\}$ in case of binary supervised classification

A probabilistic view of the learning problem (2)

- Let's note $\mathcal{D}$, the class of measurable functions from $\mathcal{X}$ to $\mathcal{Y} \cup \mathbb{R}$.
- Given $\mathcal{H} \subset \mathcal{D}$ and a local loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, the problem of supervised learning consists in solving the following optimization problem:
    - $\hat{h} = \arg min_{h \in \mathcal{H}} \mathbb{E}[\ell(h(x), y)]$
- Zero-One Loss: $\ell(h(x), y) = 1$ if $y \neq \text{sign}(h(x)))$, 0 otherwise.
- Margin (a criterion to be maximized) : $m(h(x), y) = y\text{sign}(h(x))$
- Equivalently : a loss to be minimized: $\ell(h(x), y) = \max(0, 1 - yh(x))$
- Prediction for x: take $\text{sign}(h(x))$

Binary Supervised Classification

- True risk (generalization error):$R(h) = \mathbb{E}_P[\ell(h(x), y)]$
- Find h that minimizes $R(h) = \sum_{y=-1,1} P(Y = y) \int_{\mathbb{R}^p} \ell(h(x), y) p(x|Y = y) dx$

Bayes Rule

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

Bayes classifier: a proposal for classification in an ideal world

Definition

$$h_{bay}(x) = argmax_{k=1,-1} P(Y = k|x)$$

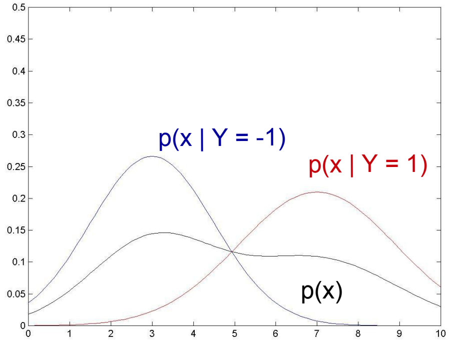
.

Bayes risk

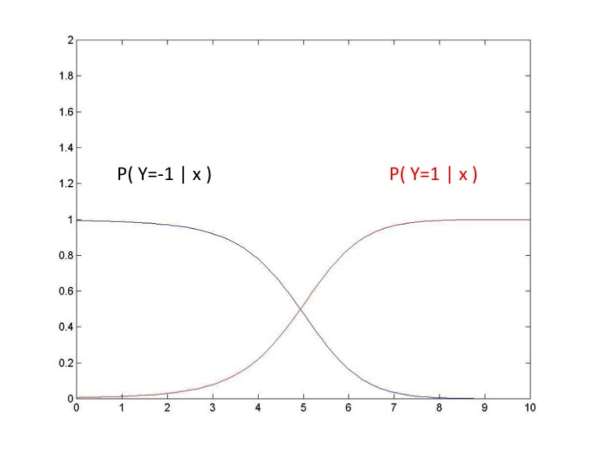
$$\begin{aligned} R(h_{bay}) &= \int_{R_1} P(h_{bay}(x) \neq 1) p(x) dx + \int_{R_{-1}} P(h_{bay}(x) \neq -1) p(x) dx & (1) \\ &= \int_{R_1} P(y = -1|x) p(x) dx + \int_{R_{-1}} P(y = 1|x) p(x) dx & (2) \end{aligned}$$

It can be shown that $R_{Bayes} = R(h_{Bayes})$ is minimal.

A 1D example with gaussian laws



Bayesian classifier: Gaussian laws

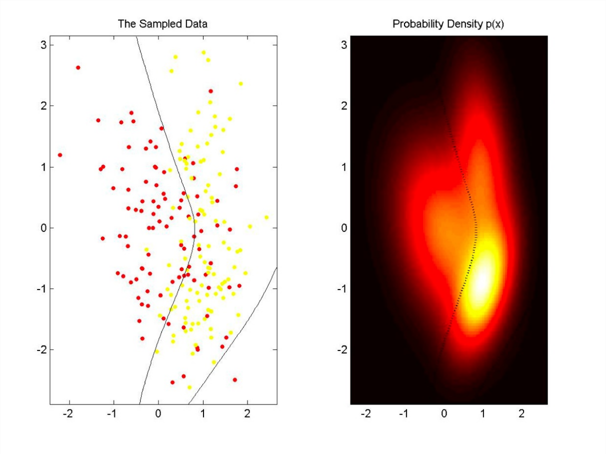


First take-home message

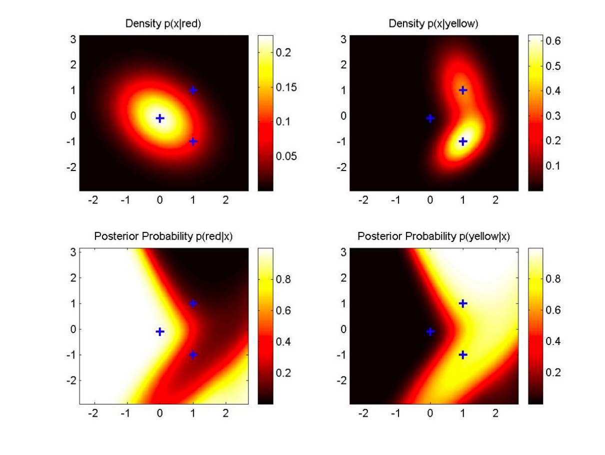
- The target function in supervised classification is the Bayes classifier for the $0-1$loss
- The target function in regression is $h(x) = \mathbb{E}[Y|x]$ for the square loss
- Now we call h_{target} the true target function

Exercise: *prove that $\mathbb{E}[Y|x]$ is the target function for the square loss.*

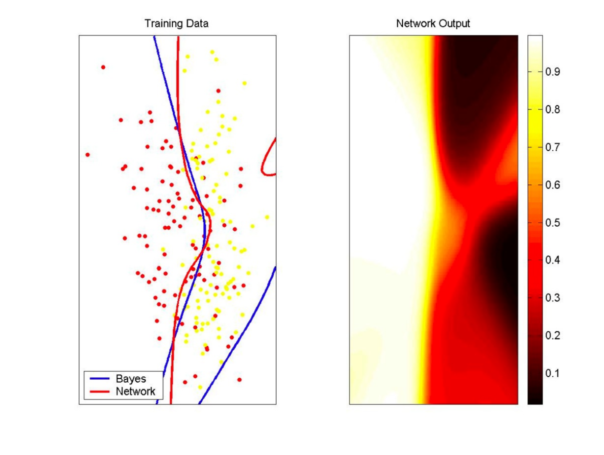
Example in 2D



Example in 2D



Using training set



Here comes the data: a statistical definition of the learning problem

Definition

- $\mathcal{S}_n$ is an i.i.d sample of size n, drawn from the joint probability law P(X,Y) fixed but unknown.
- $\mathcal{S}_n = \{(x_1, y_1), ..., (x_n, y_n)\}$.
- Statistical learning is defined by:
    - Define a learning algorithm $\mathcal{A} : S_n \rightarrow \mathcal{A}(S_n) \in \mathcal{H}$ such that $\forall P$, S_n drawn from P, $R(\mathcal{A}(S_n))$ converges towards $R(h_{target})$ in probability

Outline

Overview

- Motivation
- Towards a definition of ML
- Statistical learning

Statistical supervised learning

- Relevance of empirical risk minimization

Perceptron

References

Erreur d'excès, erreur d'approximation et erreur d'estimation

Considérons ici la perte $0-1$: Soit $R^* = \inf_h R(h)$, le risque de Bayes.
Soit $R_{\mathcal{H}} = \inf_{h \in \mathcal{H}} R(h)$.
Supposons $h_n \in \mathcal{H}$ est le classifieur estimé à partir des données S_n par minimisation du risque empirique ou par tout autre principe employant les données.

Erreur d'excès, erreur d'approximation et erreur d'estimation

$$R(h_n) - R^* = R(h_n) - R_{\mathcal{H}} + R_{\mathcal{H}} - R^*$$

L'excès d'erreur que fait h_n pa rapport au risque de Bayes est égal à la somme de deux termes:

- $R(h_n) - R_{\mathcal{H}}$: l'erreur d'estimation, mesurant à quel point on s'approche de l'optimum dans $\mathcal{H}$
- $R_{\mathcal{H}} - R^*$: l'erreur d'approximation, inhérente à la classe de fonctions choisie. par exemple, si la frontiere de séparation est non linéaire et que je me restreins à un classifieur linéaire.

Empirical risk minimization

Definition

- Empirical risk: $R_n(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h(x_i), y_i)$
- $\mathcal{A}(S_n) = \arg min_{h \in \mathcal{H}} R_n(h)$
- Where $\mathcal{H}$ is a tractable hypothesis set

How to choose $\mathcal{H}$?

A compromise bias/variance

- If $\mathcal{H}$ is too small, you cannot reach the target (large bias, no universality)
- If $\mathcal{H}$ is too big, you cannot reduce variance (large variance, no consistency)

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

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 $R(h)$?

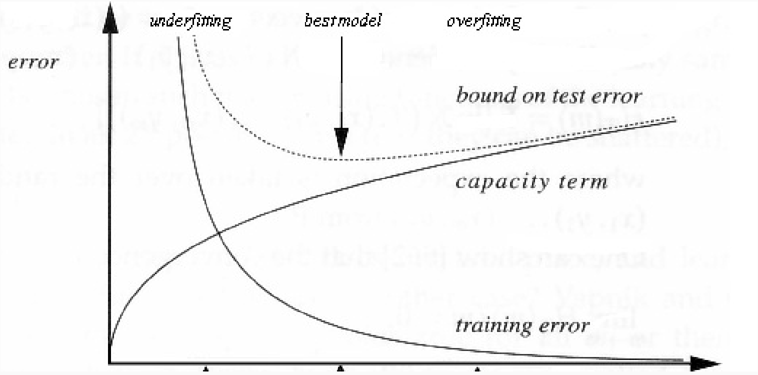
VC-dimension generalization bounds

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

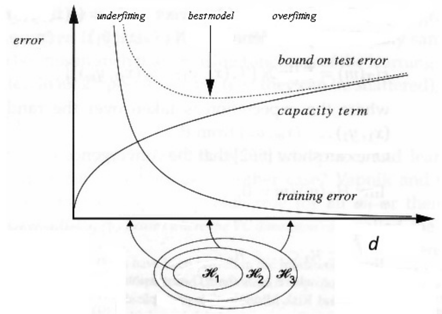
$$P_{\mathcal{S}} \left[R(h) - R_n(h) \leq \sqrt{\frac{2d \log(\frac{en}{d})}{n}} + \sqrt{\frac{\log(\frac{1}{\delta})}{2n}} \right] \geq 1 - \delta$$

where $P_{\mathcal{S}}$ denotes the probability over a random sampling $\mathcal{S} \sim P^n$, with P as the joint probability distribution on $X \times Y$.

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 $\mathcal{H}$:

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

N.B.: if VCdim($\mathcal{H}$) = 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 !

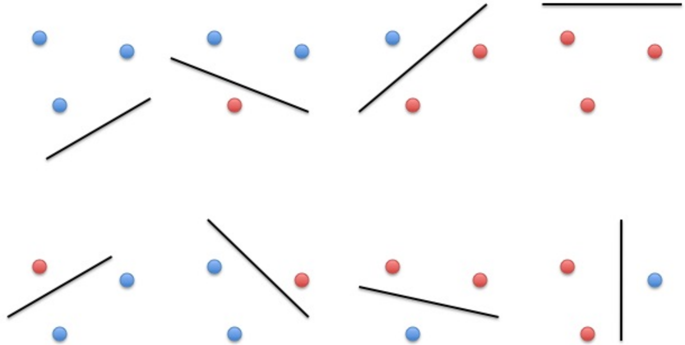
VC-dimension of Hyperplanes

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

VC-dimension of Hyperplanes

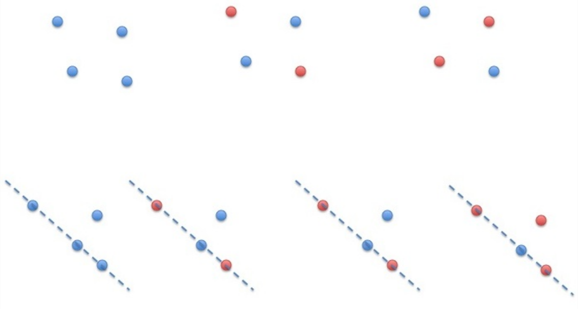
What is the VC-dimension of hyperplanes in $\mathbb{R}^2$ (denoted $\mathcal{H}_2$) ?

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.



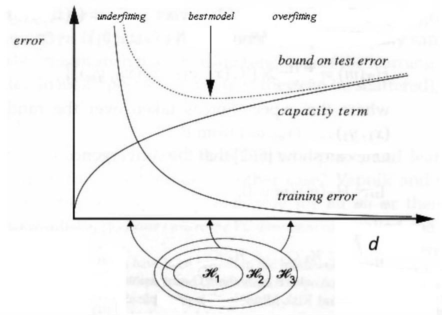
We can show that it is not possible for $\mathcal{H}_2$ to shatter 4 points.
Then $\text{VCdim}(\mathcal{H}_2) = 3$.

VC-dimension of Hyperplanes

More generally, one can prove :

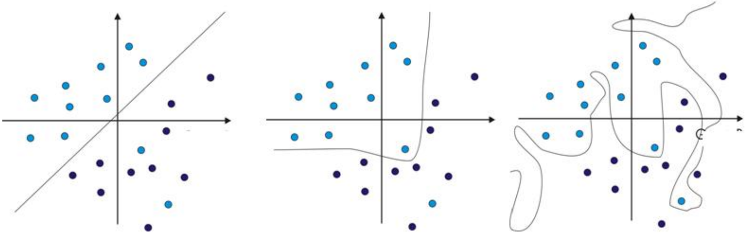
$$VCdim(\mathcal{H}_d) = d + 1$$

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*

Statistical learning: a methodology

- Three main problems to be solved :
  - **Representation problem**: determine in which representation space the data will be encoded and determine which family of mathematical functions will be used
  - **Optimization problem (focus of the course)**: formulate the learning problem as an optimization problem, develop an optimization algorithm
  - **Evaluation problem**: provide a performance estimate

Statistical learning: a methodology

- Three main problems to be solved :
    - **Representation problem**: determine in which representation space the data will be encoded and determine which family of mathematical functions will be used for your task
    - **Optimization problem** : formulate the learning problem as an optimization problem, develop an optimization algorithm
    - **Evaluation problem**: provide an estimation of performance

Statistical learning for supervised classification

Two main family of approaches:

1. Discriminant approaches : just find a classifier which does not estimate the Bayes classifier
2. Generative probabilistic approaches that are built to model $h(x) = \hat{P}(Y = 1|x)$ using $p(x|Y = 1)$, $p(x|Y = -1)$ and prior probabilities.

Outline

Overview

Statistical supervised learning

Perceptron

References

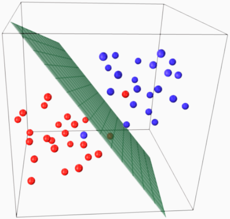
Classifieur linéaire

Définition

Supposons $\mathbf{x} \in \mathbb{R}^p$

$f(\mathbf{x}) = \text{signe}(h(x)) = \text{signe}\mathbf{w}^T\mathbf{x} + w_0)$

L'équation : $\mathbf{w}^T x + w_0 = 0$ définit un hyperplan dans l'espace euclidien $\mathbb{R}^p$



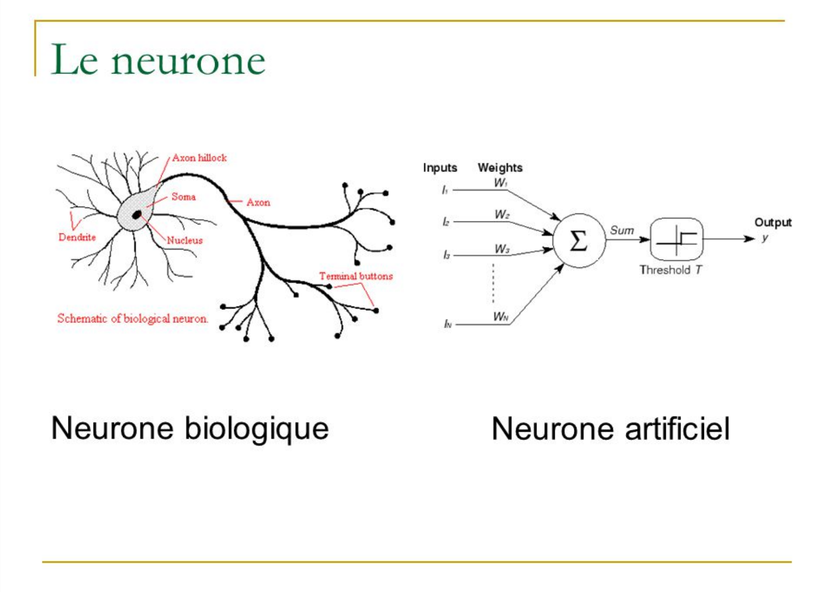
How to learn a linear classifier ?

- Model: perceptron or formal neuron (Rosenblatt 1957, 1959)
- Learning algorithm: formerly, perceptron rule, then (stochastic) gradient descent algorithm for perceptron

A linear classifier: the formal neuron and perceptron

- First model proposed by McCullogh and Pitts (physiologists) in 1943 to model the activity of a neuron
- Input signals represented by a vector $\mathbf{x}$ is processed by a neuron whose weighted synapses are linked to the input
- The neuron computes a weighted sum of the components of the signal
- Rosenblatt proposed a learning rule in 1959

Neurone

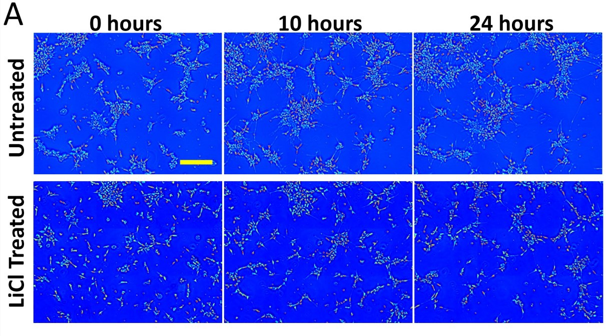


Le neurone

Neurone biologique

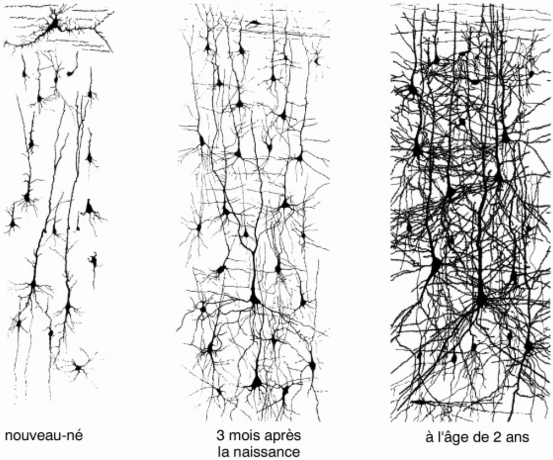
Neurone artificiel

Neuron network growth over 24 hours



In 2014, the group of Gabriel Popescu at Illinois U. visualized a growing net of stem cell neurons using spatial light interference microscopy (SLIM). Ref: `http://light.ece.illinois.edu/wp-content/uploads/2014/03/Mir_SRep_2014.pdf`
Video: `https://youtu.be/KjKsU_4s0nE`

Développement des réseaux de neurones chez l’enfant



nouveau-né

3 mois après la naissance

à l'âge de 2 ans

Développement des réseaux de connections entre les neurones chez l'enfant.

Re: Museum de Toulouse `http://www.museum.toulouse.fr/-/connecte-a-vie-notre-cerveau-le-meilleur-des-reseaux-2-3-`

Formal neuron and perceptron

- $h_{perc}(x) = \text{sign}(\mathbf{w}^T x)$
- $sign(a) = 1$ if $a \geq 0$ and -1 otherwise

Données d'apprentissage:

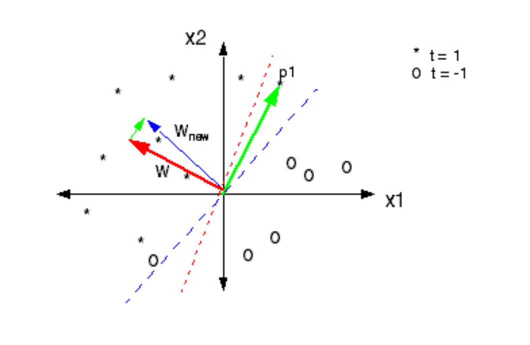
- $\mathcal{S} = \{(\mathbf{x_1}, \mathbf{y_1}), ..., (\mathbf{x_n}, \mathbf{y_n})\}$
- $x_i \in \mathbb{R}^{p+1}$: the 0^{th} component is fixed to 1.
- $y_i \in \{-1, +1\}$

Algorithme classique du perceptron

Algorithme (pseudo code)

```
• Continue = 1
• Fixer T nombre maximal d'itérations
• $\mathbf{w}_0 = 0$
• $k = 0$
• $\epsilon << 1$
• TANT QUE (continue $> \epsilon$) ou $(nk < T)$ FAIRE
    • Pour i=1 à n
        • $k = k + 1$
        • Si $y_i \neq sign(\mathbf{w}^T x_i)$, alors je corrige: w(k+1) = w(k) + $y_i.\mathbf{x}_i$
        • Sinon pas de correction
    • CONT = $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|$
```

Correction effectuée par le perceptron



Convergence de l'algorithme du perceptron

L'algorithme converge si les données sont exactement linéairement séparables. Voici un théorème un tout petit peu plus général.

Théorème de convergence

Supposons qu'il existe un paramètre $\mathbf{w}^*$ tel que $\|\mathbf{w}\| = 1$, et $\gamma > 0$ tels que pour tout $i = 1, \ldots n$:

$$y_i(\mathbf{x}_i^T \mathbf{w}^*) \geq \gamma$$

et qu'il existe $R > 0$: $\|\mathbf{x}_i\| \leq R$,
Alors l'algorithme du perceptron converge en au plus $\frac{R^2}{\gamma^2}$ itérations.

Preuve 1/2

- $\mathbf{w}(0) = 0$
- Supposons que la k-ieme erreur est faite sur l'exemple d'indice t, nous avons:

$$\begin{aligned}
\mathbf{w}(k+1)^T\mathbf{w}^* &= (\mathbf{w}(k) + y_t\mathbf{x}_t)^T\mathbf{w}^* \\
&= \mathbf{w}(k)^T\mathbf{w}^* + y_t\mathbf{x}_t^T\mathbf{w}^* \\
&\geq \mathbf{w}(k)^T\mathbf{w}^* + \gamma
\end{aligned}$$

Par récurrence sur k : $\mathbf{w}(k+1)^T\mathbf{w}^* \geq k\gamma$
(Par Cauchy-Schwartz:
$\|\mathbf{w}(k+1)^T\mathbf{w}^*\| \leq \|\mathbf{w}(k+1)\|\|\mathbf{w}^*\| \leq \|\mathbf{w}(k+1)\|$)
donc : $\|\mathbf{w}(k+1)\| \geq \|\mathbf{w}(k+1)^T\mathbf{w}^*\| \geq k\gamma$

Preuve 2/2

On dérive ensuite une majoration pour $\|\mathbf{w}(k+1)\|$:

$$\begin{aligned} \|\mathbf{w}(k+1)\|^2 &= \|\mathbf{w}(k) + y_t \mathbf{x}_t\|^2 \\ &= \|\mathbf{w}(k)\|^2 + y_t^2 \|\mathbf{x}_t\|^2 + 2 y_t \mathbf{x}_t^T \mathbf{w}(k) \end{aligned}$$

le terme de correction $2 y_t \mathbf{x}_t^T \mathbf{w}(k)$ est par définition négatif donc:

$$\|\mathbf{w}(k+1)\|^2 \leq \|\mathbf{w}(k)\|^2 + R^2$$

Par récurrence sur k, on a :

$$\|\mathbf{w}(k+1)\|^2 \leq kR^2$$

Au final, en prenant les deux inégalités:
on a : $k^2\gamma^2 \leq \|w(k+1)\|^2 \leq kR^2$ donc : $k \leq \frac{R^2}{\gamma^2}$
Si k est borné par $\frac{R^2}{\gamma^2}$, cela veut dire qu'en au plus $\frac{R^2}{\gamma^2}$ itérations, on n'a plus de corrections à faire.

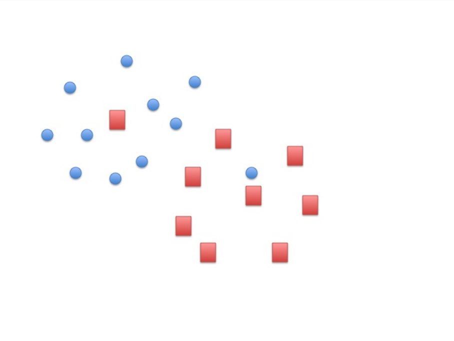
Limitations du perceptron

Deux types de non séparabilité:

1. Données presque " linéairement séparables": quelques "outliers" dans les données, qu'il vaut mieux éviter d'apprendre à classer
2. Données séparables mais avec une frontière non linéaire
3. NB : on cumule en général les deux difficultés

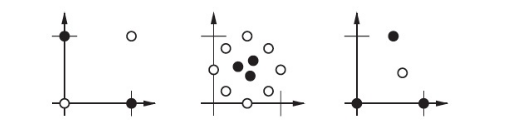
Limites d'un perceptron 1

Exemple 1 de données non séparables:
Outliers dans les données : mieux vaut éviter de les



Limites d’un perceptron 2

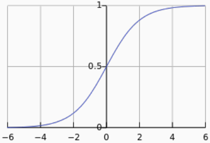
Exemple 2 : des données non séparables linéairement mais il existe une frontière non linéaire:



- **Premier problème :** XOR problem: un perceptron seul ne peut implémenter une fonction XOR
- Solution :
  - Soit on rajoute une couche de neurones avant le neurone de sortie perceptron multi-couches (algorithme d’apprentissage par rétro-propagation du gradient), Werbos 1974, Le Cun 1985, Rumelhart et al. 1986.
  - Soit on transforme les données en les plongeant dans un espace où elles sont linéairement séparables (see Practical session)

Apprendre un perceptron(vue plus générale)

- Remplacer la fonction signe par une sigmoide différentiable
  - $sigm(x) = \frac{1}{1+\exp(-\frac{1}{2}x)}$



- Définir une fonction de perte différentiable
  - $\ell_i(\mathbf{w}) = (y_i - \ sigm(\mathbf{w}^T x))^2$
  - $L(\mathbf{w}) = \sum_i \ell_i(\mathbf{w})$

Apprendre un perceptron (vue plus générale)

Algorithme d'apprentissage du perceptron par gradient local stochastique

```
STOP = faux
$\epsilon$; nbIter; $j = 0$; $t = 0$
Initialiser $\mathbf{w}_0$
Jusqu'à ce que STOP soit vrai:
    1. Pour j =1 jusqu'à n:
        • Tirer uniformément un indice $i$ parmi $\{1, \ldots, n\}$
        • $\mathbf{w}^{\mathbf{t+1}} = \mathbf{w}^{\mathbf{t}} - \eta \nabla_{\mathbf{w}} \ell_{\mathbf{i}}(\mathbf{w})$
        • $t \rightarrow t + 1$
    2. STOP = $(L(||\mathbf{w}(t) - \mathbf{w}(t-1)|| < \epsilon)$ et $(nbIter \leq nbMax)$
```

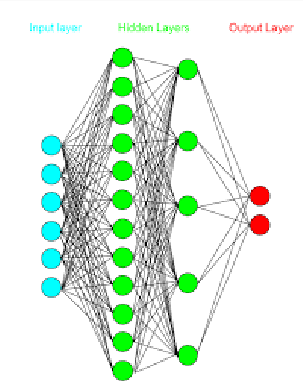
Perceptron

- Early stopping: arrêter avant de sur-apprendre (nbIter petit)
- Eviter le sur-apprentissage : contrôler la norme du vecteur **w** pendant l'apprentissage
  - La fonction de perte devient : $L(\mathbf{w}) = \sum_i \ell_i(\mathbf{w}) + \lambda||\mathbf{w}||^2$
  - La mise à jour devient : $\mathbf{w^{t+1}} = \mathbf{w^t} - \eta\nabla_{\mathbf{w}}\ell_{\mathbf{i}}(\mathbf{w})$
- Variante intéressante (celle utilisée en pratique)
  - Descente de gradient local et stochastique (Bottou 1991: application aux réseaux de neurones)

Passer au non linéaire

- Définir $\phi : \mathcal{X} \rightarrow \mathcal{F}$ une fonction non linéaire de re-description (en anglais *feature map*)
- Empiler les couches de neurones, chaque couche de neurone étant vue comme une redescription des entrées

Réseau de neurones formels (perceptron multi-couches)



Outline

Overview

Statistical supervised learning

Perceptron

References

References

- The elements of statistical learning, Hastie, Tibshirani, Friedman, Springer (free pdf).
- Pattern recognition and machine learning, Chris Bishop