

S203 : Apprentissage automatique Introduction

30/01/2018

S. Herbin, B. Le Saux, A. Chan Hon Tong, (A. Boulch)

Organisation du cours

30/01/2018	Introduction	SH
06/02/2018	Apprentissage supervisé et SVM	SH
13/02/2018	Apprentissage supervisé – Autres classifieurs	BLS
20/02/2018	Apprentissage non supervisé	BLS
06/03/2018	Réseaux de neurones	ACHT
13/03/2018	Réseaux Profonds	ACHT
27/03/2018	Exam et projet	ACHT + SH

Equipe pédagogique



Stéphane Herbin
stephane.herbin@onera.fr



Bertrand Le Saux
bertrand.le_saux@onera.fr



Adrien Chan Hon Tong
adrien.chan_hon_tong@onera.fr



Alexandre Boulch
alexandre.boulch@onera.fr

ONERA

- DTIS: Département Traitement de l'Information et Systèmes à Palaiseau
- Equipe IVA (Image, Vision, Apprentissage)
- Activités de Machine Learning: télédétection, vidéo, drones
- Stages et thèses: (<http://w3.onera.fr/stages/stages-dtis>)

Aujourd'hui

- Organisation du cours
- Introduction générale:
 - Exemples, définitions, problématiques, approches, pratiques
- Deux approches élémentaires à connaître:
 - Modélisation bayésienne
 - k plus proches voisins (kNN)

Organisation

- Cours magistral: 1h
- TP applicatifs: 2h
- Examen: 1h (le 27/03/2018)
- Projet: deux dernières séances de TP + travail libre.
- Note : 3 TP (30%), Exam (30%), Projet (40%)

Apprentissage Automatique (« Machine Learning »)

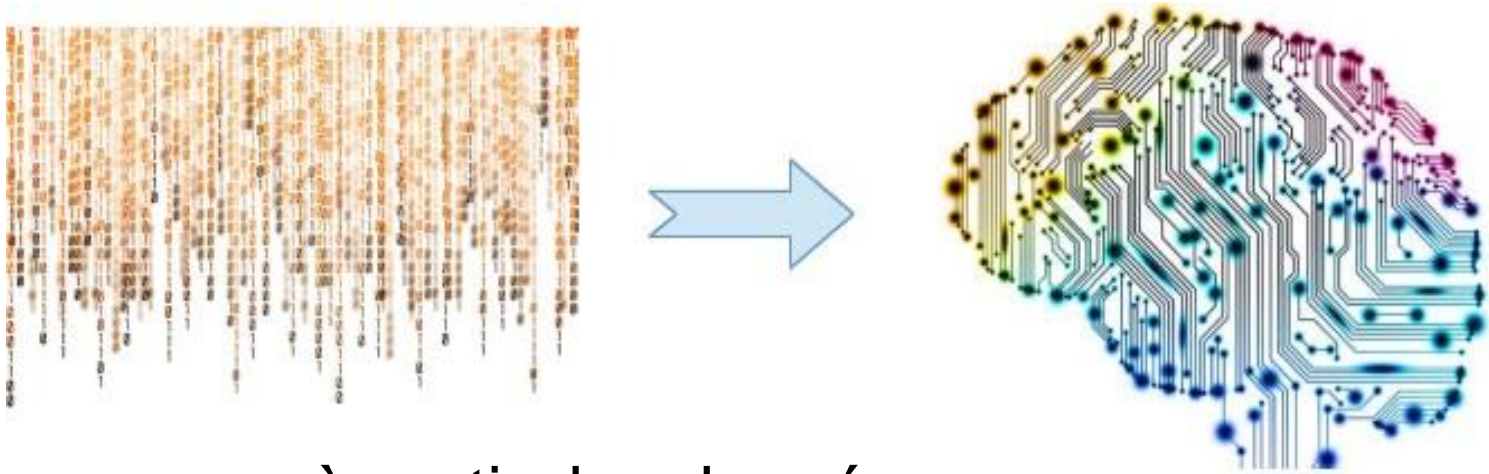


Donner aux ordinateurs la capacité d'apprendre une tâche sans être explicitement programmés à l'exécuter
[Samuel, 59]

Conception, analyse, développement et implémentation de méthodes permettant à une machine (au sens large) d'évoluer par un processus systématique, et ainsi de remplir des tâches difficiles ou problématiques par des moyens algorithmiques plus classiques.
[Wikipedia]

Des structures de données auto-configurables qui permettent à un ordinateur d'exécuter des tâches qui seraient dites "intelligentes" si un humain s'en chargeait.
[Chris Bishop]

Apprentissage Automatique

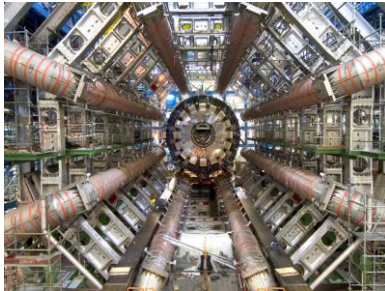


Programmer à partir des données :

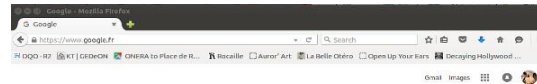
- *Comment extraire l'information des données pour nous aider à prendre de meilleures décisions ?*
- *Comment prendre automatiquement des décisions en fonction des données ?*
- *Comment adapter un système à un environnement changeant ?*

Données = carburant du ML

CERN /
Large Hadron Collider
~70 Po/an



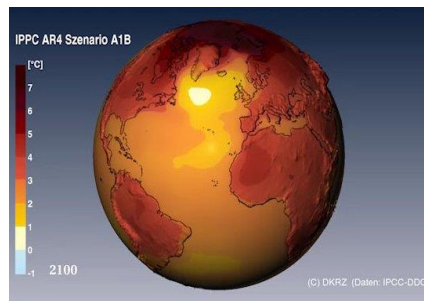
Google :
24 PetaOctets/jour



Copernicus :
> 1Po/an



DKRZ (Climat)
500 Po



Google



Publicité Croquer À propos Confidentialité Conditions Produits

Square Kilometer Array
1376 Po/an (en 2024)



BIG DATA

Apprentissage automatique : applications

Anti-Spam (*Classifieur Bayésien*)



1997 : DeepBlue bat Kasparov

2017: Alpha GO bat Ke Jie



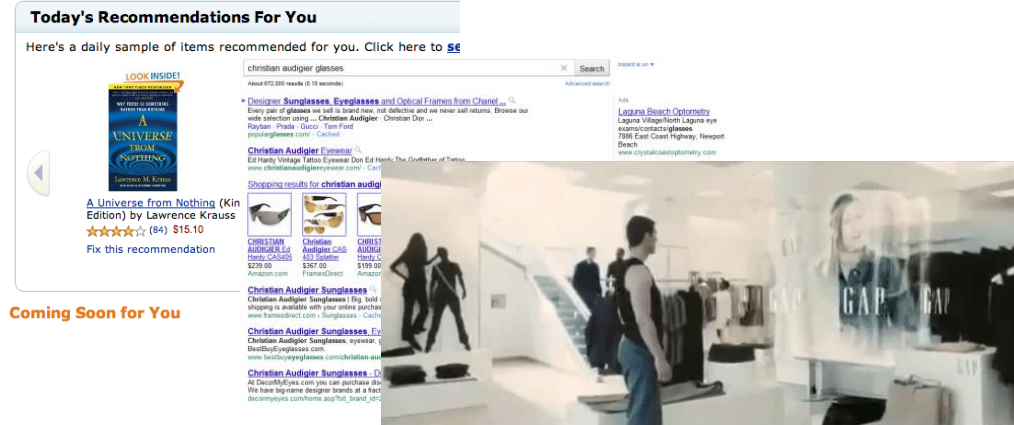
Tri postal automatique (*détection de chiffres manuscrits par réseaux de neurones*)



Apprentissage automatique : applications

Recommandation ciblée
(régression logistique)

Michel, Welcome to Your Amazon.com (if you're not Michel Trottier)

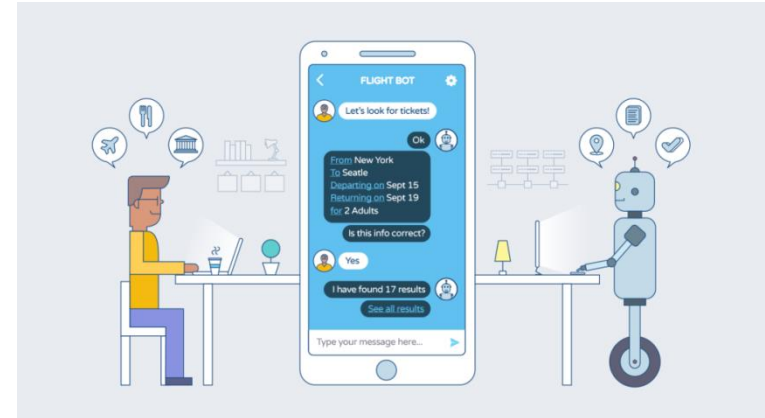


Appareil photo avec détection
de visages (boosting)

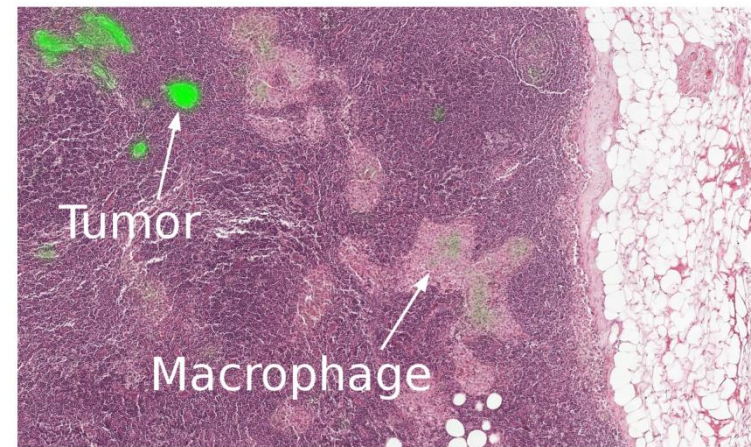


Apprentissage automatique : applications

Chat Bots
(*Réseaux de neurones*)



Détection de tumeurs
(*Réseaux de neurones*)



Applications industrielles du « Machine Learning »

- Predictive maintenance or condition monitoring
- Warranty reserve estimation
- Propensity to buy
- Demand forecasting
- Process optimization
- Telematics

Manufacturing



- Predictive inventory planning
- Recommendation engines
- Upsell and cross-channel marketing
- Market segmentation and targeting
- Customer ROI and lifetime value

Retail



- Alerts and diagnostics from real-time patient data
- Disease identification and risk stratification
- Patient triage optimization
- Proactive health management
- Healthcare provider sentiment analysis

Healthcare and Life Sciences



- Aircraft scheduling
- Dynamic pricing
- Social media – consumer feedback and interaction analysis
- Customer complaint resolution
- Traffic patterns and congestion management

Travel and Hospitality



- Risk analytics and regulation
- Customer Segmentation
- Cross-selling and up-selling
- Sales and marketing campaign management
- Credit worthiness evaluation

Financial Services



- Power usage analytics
- Seismic data processing
- Carbon emissions and trading
- Customer-specific pricing
- Smart grid management
- Energy demand and supply optimization

Energy, Feedstock, and Utilities



Activité économique

Machine Learning-Gen
(96 Companies)

Machine Learning-App
(201 Companies)

Computer Vision-Gen
(97 Companies)

Computer Vision-App
(73 Companies)

Smart Robots
(52 Companies)

Virtual Personal Assistants
(71 Companies)

NLP-Speech Recog.
(65 Companies)

NLP-Gen
(127 Companies)

Speech to Speech Trans.
(15 Companies)

Context Aware Comp.
(28 Companies)

Gesture Control
(30 Companies)

Recommendation Eng.
(54 Companies)

Video ACR
(14 Companies)

Artificial Intelligence

Contact
info@venturescanner.com
 to see all 855 companies

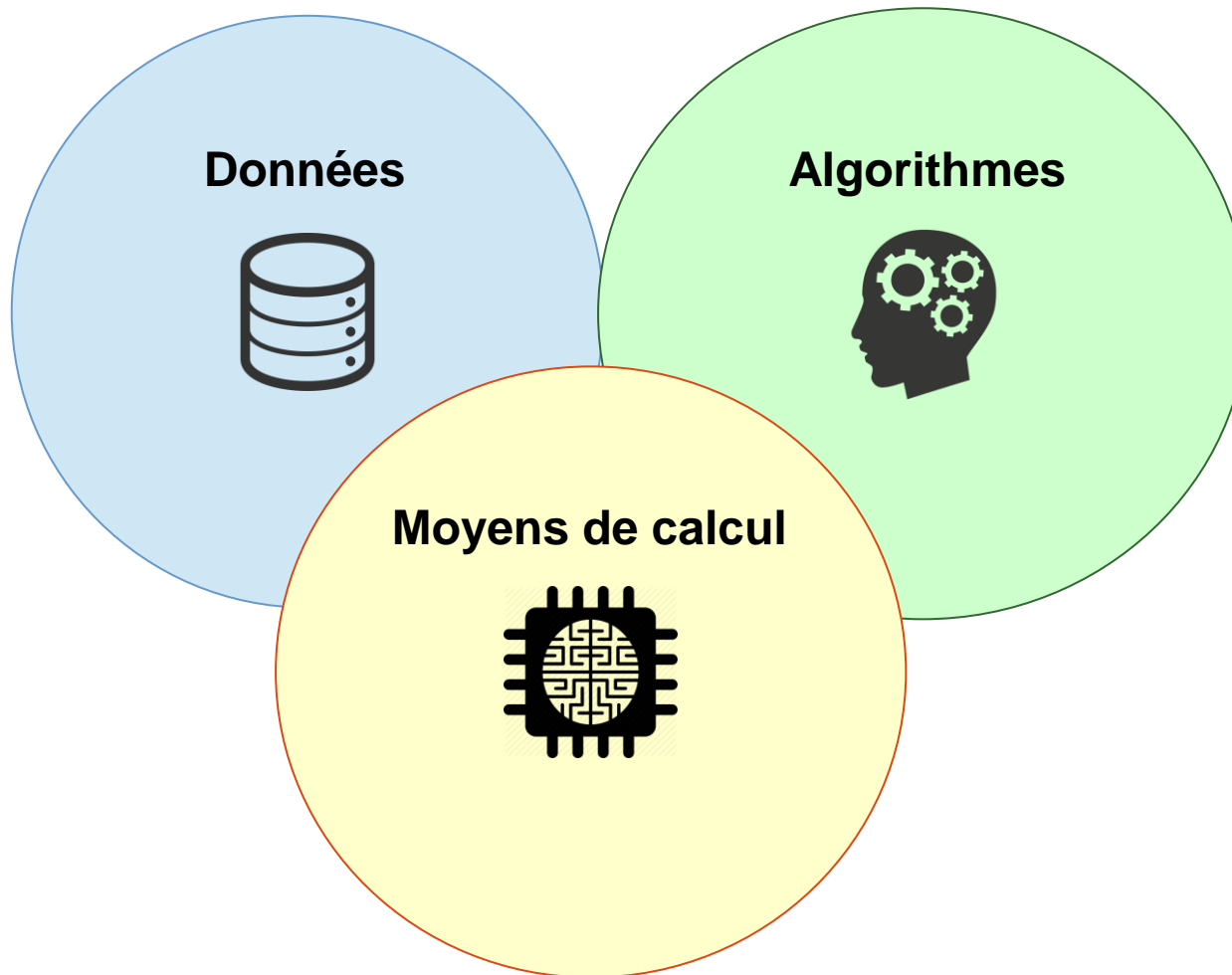
Venture Scanner

Pourquoi l'apprentissage automatique ?

- Raison épistémologique
 - On ne sait pas modéliser les problèmes complexes ... mais on dispose d'exemples en grand nombre décrivant la variété des situations
- Raison scientifique
 - L'apprentissage est une faculté essentielle du vivant
- Raison économique
 - La récolte de données est plus facile que le développement d'expertise

« Deep Learning » : le mot clé inévitable

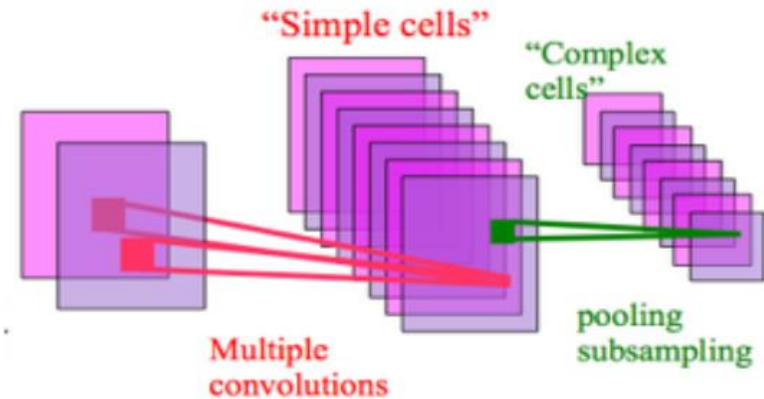
Une rupture scientifique et technologique en apprentissage



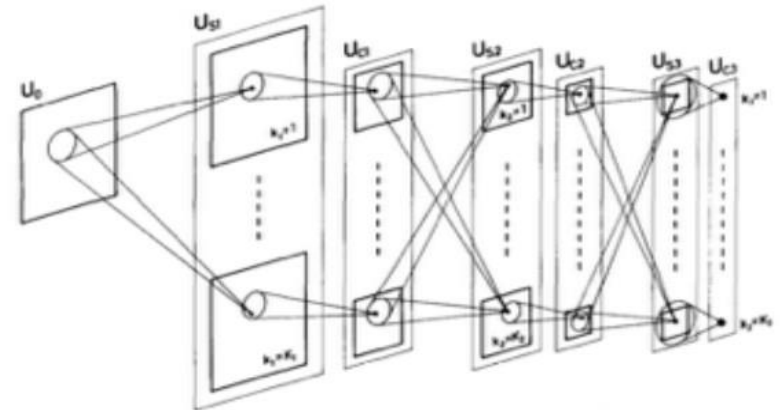
Avant le « Deep Learning »

Modèles de réseaux de neurones **hiérarchiques** :

Hubel & Wiesel [60s]



Fukushima's Neocognitron [70s]



Figures extraites de LeCun @ CVPR'2015

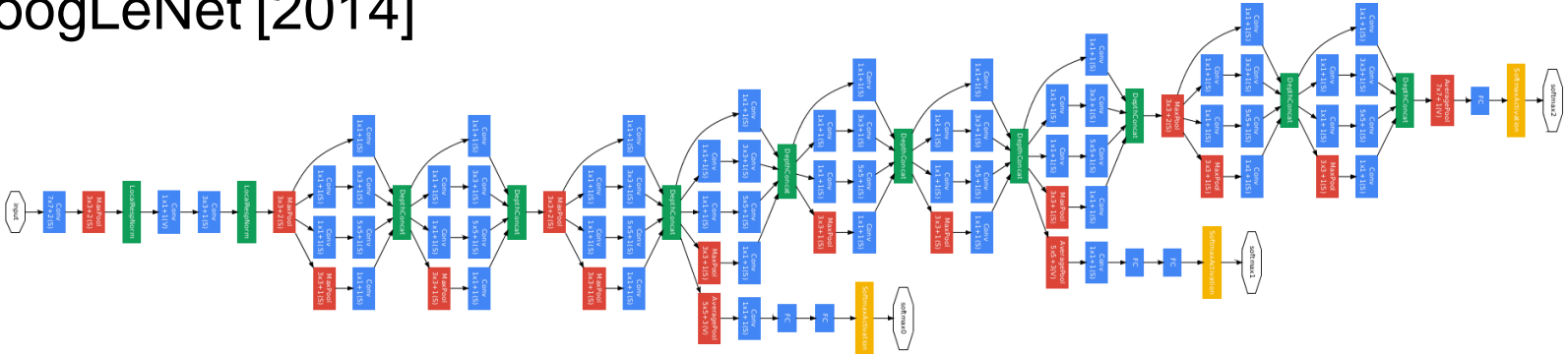
ONERA

THE FRENCH AEROSPACE LAB

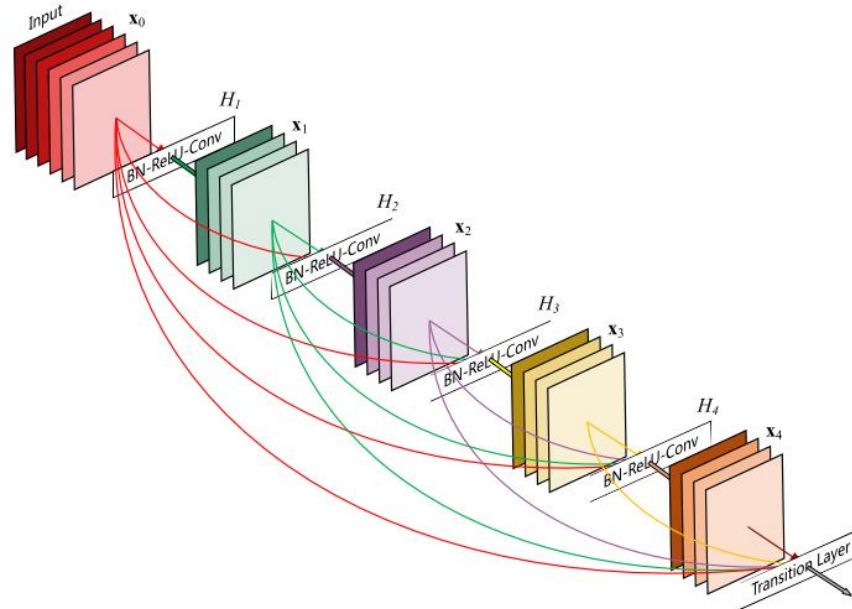
« Deep Learning »

Modèles de réseaux de neurones hiérarchiques :

GoogLeNet [2014]



DenseNet [2016]



« Deep Learning »

Modèles de réseaux de neurones hiérarchiques :

L'invasion [2010s] : classification Imagenet

2012 Teams	%error	2013 Teams	%error	2014 Teams	%error
Supervision (Toronto)	15.3	Clarifai (NYU spinoff)	11.7	GoogLeNet	6.6
ISI (Tokyo)	26.1	NUS (singapore)	12.9	VGG (Oxford)	7.3
VGG (Oxford)	26.9	Zeiler-Fergus (NYU)	13.5	MSRA	8.0
XRCE/INRIA	27.0	A. Howard	13.5	A. Howard	8.1
UvA (Amsterdam)	29.6	OverFeat (NYU)	14.1	DeeperVision	9.5
INRIA/LEAR	33.4	UvA (Amsterdam)	14.2	NUS-BST	9.7
		Adobe	15.2	TTIC-ECP	10.2
		VGG (Oxford)	15.2	XYZ	11.2
ConvNet / non-ConvNet		VGG (Oxford)	23.0	UvA	12.1

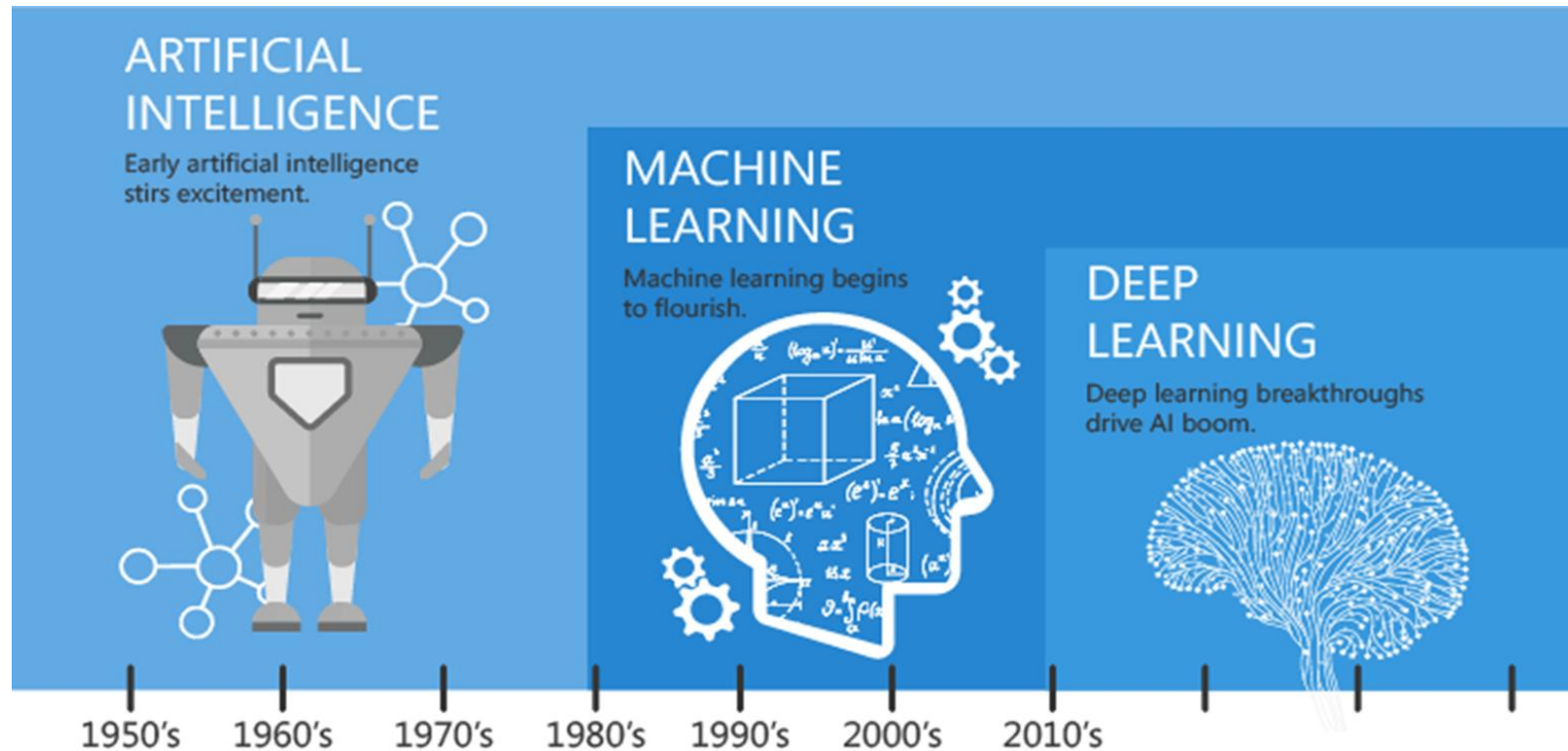
Figure extraite de LeCun @ CVPR'2015

→ *Performants pour une multitude de tâches (autres que vision), avec des jeux de données variés.*

Machine Learning

- Un domaine scientifique hybride:
 - Statistique
 - Intelligence artificielle
 - « Computer science »
- Utilisant des techniques généralistes:
 - Optimisation numérique
 - Hardware
 - Gestion de base de données

Machine Learning & Artificial Intelligence



Since an early flush of optimism in the 1950's, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created ever larger disruptions.

<https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>

Statistique

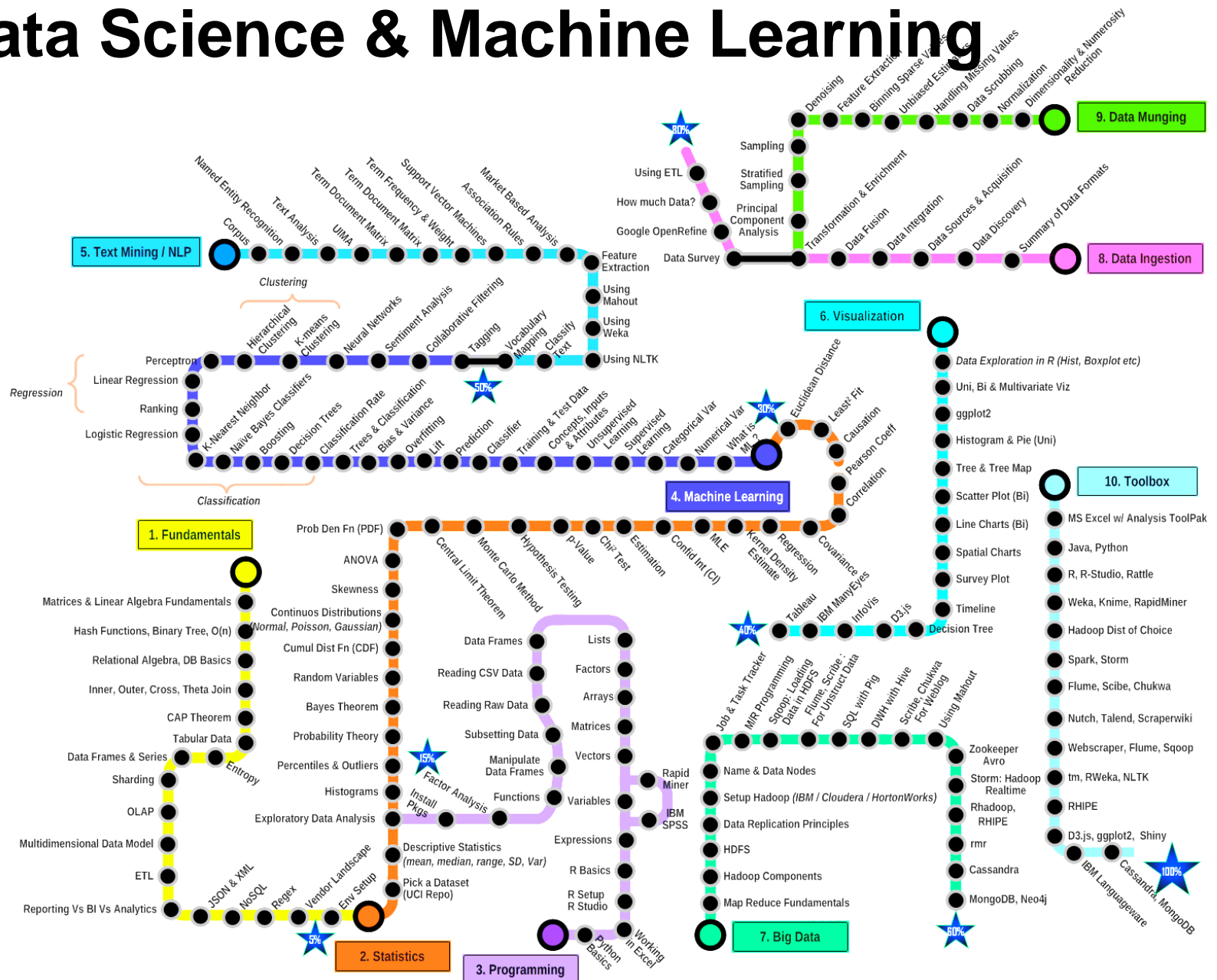
&

Machine Learning

- Données de faible dimension (moins de 100)
- Données très bruitées
- Peu de structure dans la répartition des données (i.i.d.), modélisation par modèles paramétriques simples.
- Le principal problème est de séparer le bruit du signal.
- On sait démontrer que ça marche (preuves de convergence)

- Grandes dimensions (plusieurs dizaines de milliers)
- (Faible bruit)
- Données sont organisées selon une certaine structure de dépendances mais complexe.
- Le principal problème est la maîtrise de la structure cachée des données.
- On sait démontrer que ça marche (démonstration sur exemples ou cas tests)

Data Science & Machine Learning



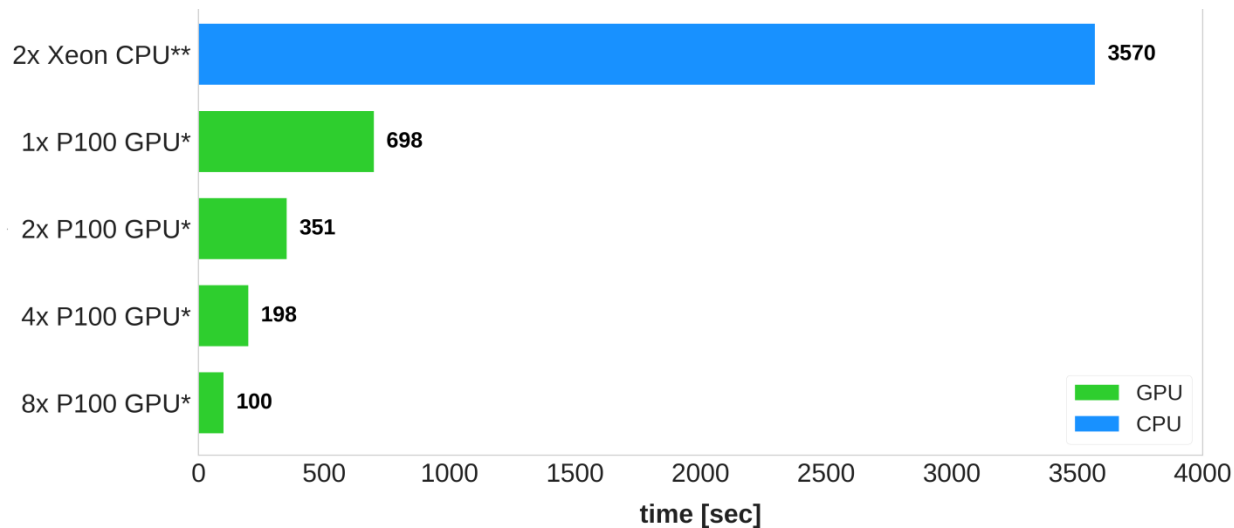
Hardware & Machine Learning

GPU!

Calcul parallèle

Rapide

Economique



<http://github.com/h2oai/perf/>

*NVIDIA DGX-1, **Dual Intel Xeon E5-2630 v4
U.S. Census dataset (predict Income): 45k rows, 10k cols
Elastic Net Model Parameters: 5-fold cross-validation, $\alpha = \{\frac{1}{7}, i = 0 \dots 7\}$, full λ -search

Domaines techniques utilisant du ML

- ML comme outil de conception
 - Vision & Reconnaissance des formes
 - Traitement du langage
 - Traitement de la parole
 - Big Data
 - Robotique
- ML comme outil explicatif
 - Neuroscience
 - Psychologie
 - Sciences cognitives

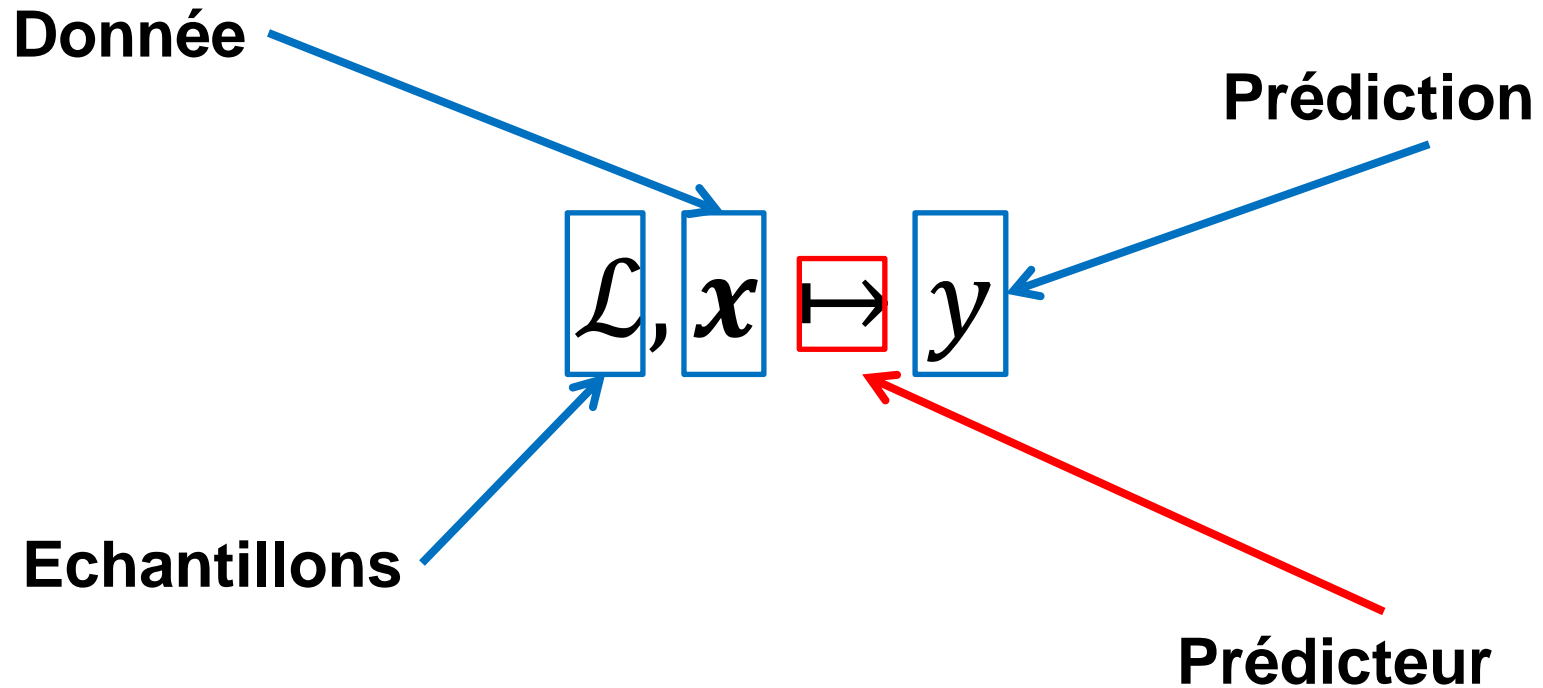
COMMENT ÇA SE PRATIQUE?

Démarche

Machine Learning

- Donnée (x)
 - Mesures, texte, image, enregistrement, vidéo
- Prédiction (y)
 - Décision, choix, action, réponse, préférence, groupe, valeur...
- Echantillons (\mathcal{L})
 - Exemples de données et de prédictions (bonnes ou mauvaises)

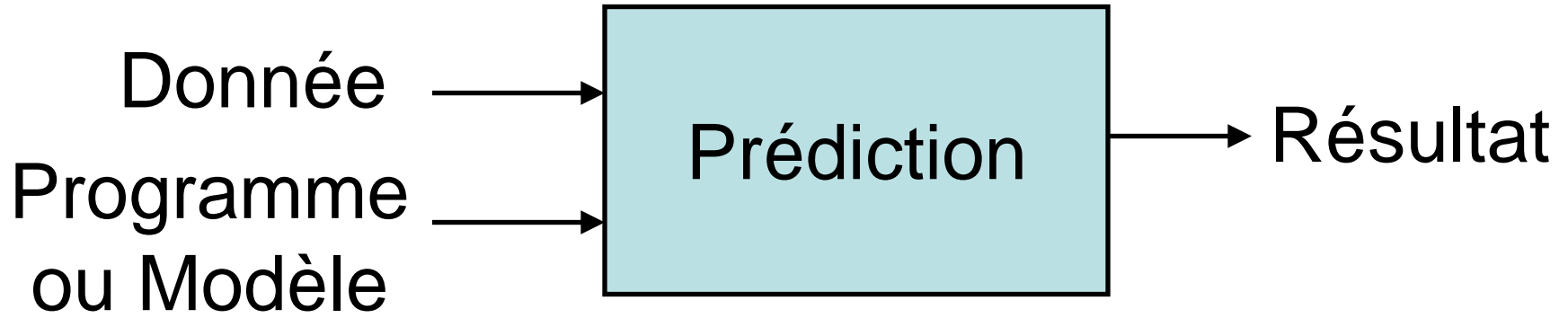
Machine Learning



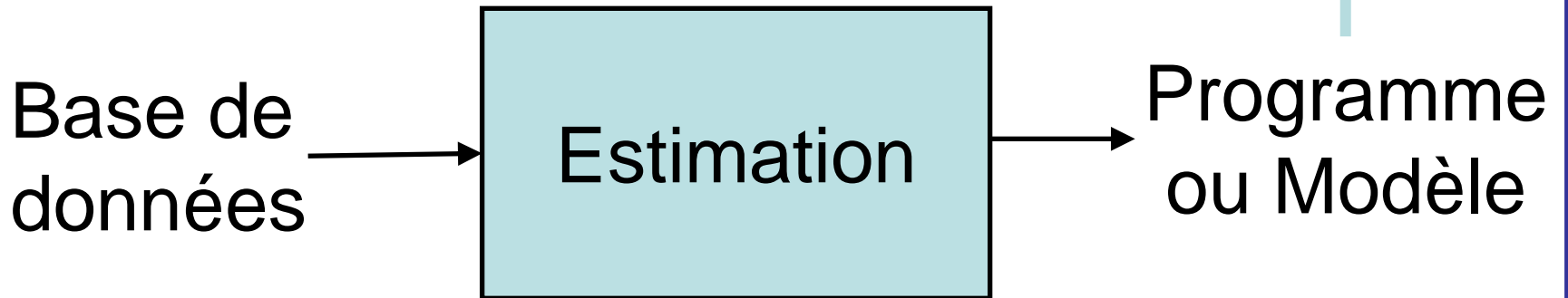
- Hypothèse forte: les échantillons contiennent toute l'information exploitable
- Prédicteur = « interpolateur » à partir des données \mathcal{L}

Deux phases

Test



Apprentissage



Deux phases

Test

$$\theta, x \mapsto y$$

Apprentissage

$$\mathcal{L} \mapsto \theta$$



Exemple: Reconnaissance de chiffres manuscrits



- Comment définir les éléments ?

$$\mathcal{L}, \theta, x, y$$

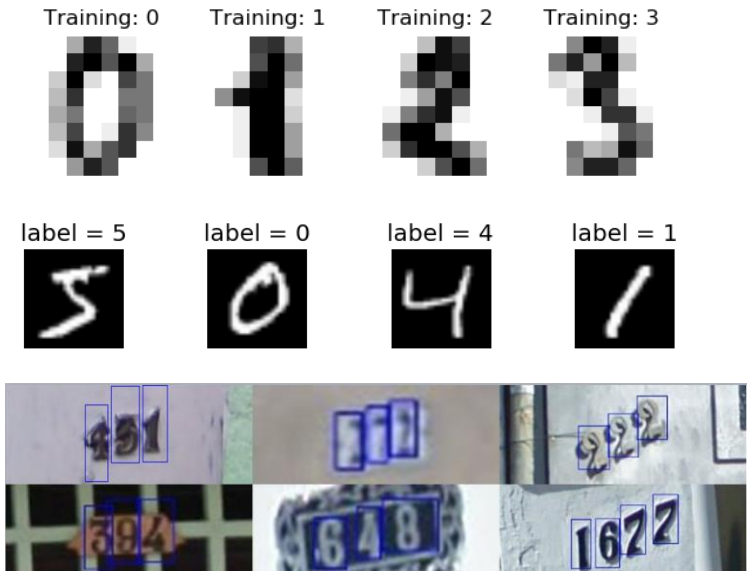
- Les fonctions d'apprentissage et de prédiction?

$$\mathcal{L} \mapsto \theta$$

$$\theta, x \mapsto y$$

Etape 1: choix de la base d'apprentissage

- Elle existe:
 - Scikit-learn:
 - MNIST:
 - SVHN:



- Il faut la construire:
 - Recueil de données existantes
 - Expérimentations (photos, mesures...)

Etape 2: analyse & représentation des données

0	0	0	0
0	1	0.5	0
0	0.5	1	0.5
0	0	0.5	1

Image

**Extraction de
caractéristiques**

*Un exemple de
mise en forme*

0
0
0
0
0
1
0.5
0
0
0.5
1
0.5
0
0
0.5
1

Vecteur

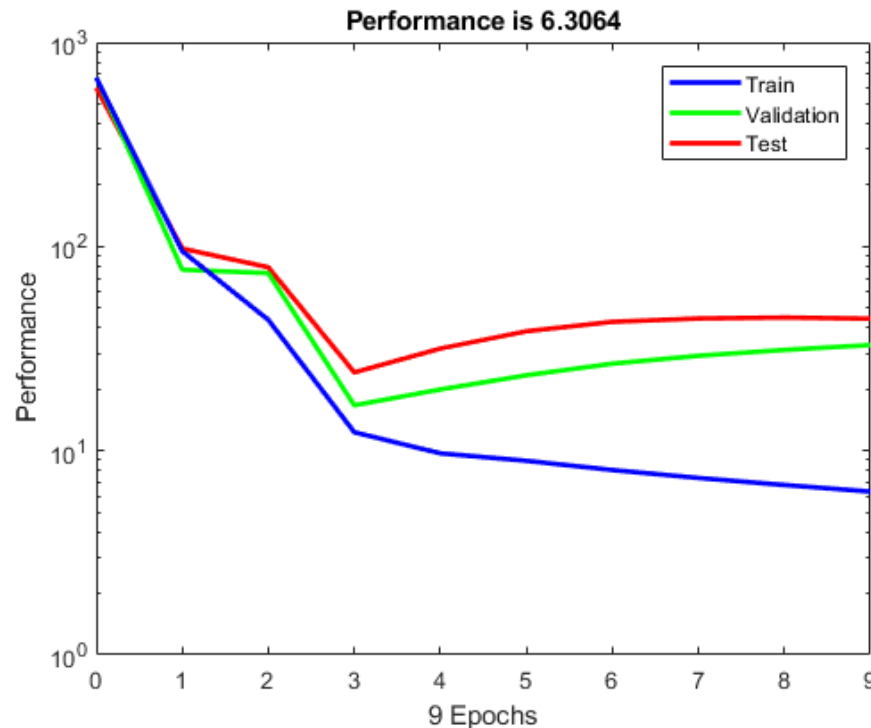
Etape 3: choix de l'approche

- Quel type?
 - Classification
 - On connaît les classes cibles → Apprentissage supervisé
- Nature des données?
 - Vecteurs de taille fixe mais grands → algorithmes avec bon contrôle de la régularisation
- Taille de la base de données?
 - Grande (> 10000 exemples) → optimisation efficace
- Plusieurs choix possibles:
 - Arbres de décision, SVM, Réseaux de neurones...

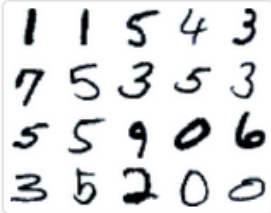
Etape 4: optimisation

Apprentissage =

- appliquer un optimiseur
- régler ses paramètres
- vérifier que l'apprentissage se passe bien










Etape 5: évaluation



MNIST 50 results collected

Units: error %

Classify handwritten digits. Some additional results are available on the [original dataset page](#).

Result	Method	Venue	Details
0.21%	Regularization of Neural Networks using DropConnect 	ICML 2013	
0.23%	Multi-column Deep Neural Networks for Image Classification 	CVPR 2012	
0.23%	APAC: Augmented PAttern Classification with Neural Networks 	arXiv 2015	
0.24%	Batch-normalized Maxout Network in Network 	arXiv 2015	<button>Details</button>
0.29%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree 	AISTATS 2016	<button>Details</button>
0.31%	Recurrent Convolutional Neural Network for Object Recognition 	CVPR 2015	
0.31%	On the Importance of Normalisation Layers in Deep Learning with Piecewise Linear Activation Units 	arXiv 2015	

Résumé des étapes

1. Constituer une base d'apprentissage
2. Analyser et visualiser les données (format, distribution, représentation, dimensions, quantité...)
3. Concevoir le modèle (type de prédicteur, principe d'apprentissage)
4. Optimiser (l'apprentissage proprement dit)
5. Evaluer

LES VARIÉTÉS D'APPRENTISSAGE

Types de prédictions

- **Classification**

- Binaire: spam / non spam
- Identification: « tata Monique »

- **Régression**

- Prédiction de température, de cours de bourse
- Localisation d'objet dans image

- **Regroupement**

- Photos dans base de données personnelle

- **Texte**

- « C'est un chat qui saute sur une table. »

Types d'apprentissage

- **Apprentissage supervisé**
 - Les données d'apprentissage contiennent les objectifs de prédiction (annotations)
- **Apprentissage non supervisé**
 - Les données d'apprentissage sont brutes
- **Apprentissage semi-supervisé**
 - Les données d'apprentissage sont partiellement annotées
- **Apprentissage par transfert**
 - Les données d'apprentissage sont proches du problème visé
- **Apprentissage par renforcement**
 - Les prédictions sont issues d'une séquence d'actions et sont caractérisées par une mesure de qualité (« reward »)

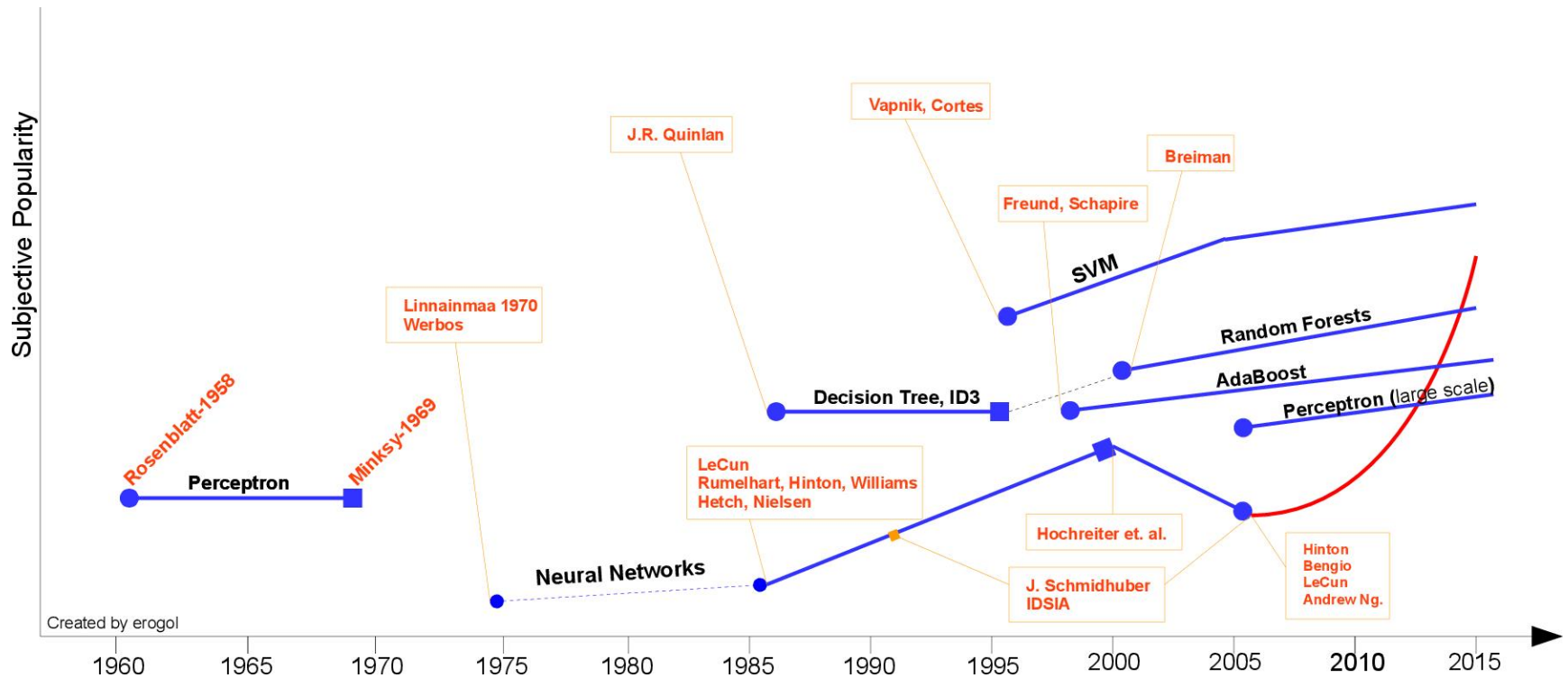
Optimisation

- **Optimisation convexe**
 - Ex. Minimisation séquentielle de problème quadratique
- **Optimisation stochastique**
 - Ex. Descente de gradient stochastique, Algorithmes génétiques
- **Optimisation sous contraintes**
 - Ex. Programmation linéaire
- **Optimisation combinatoire**
 - Ex. Algorithmes gloutons

Format du prédicteur

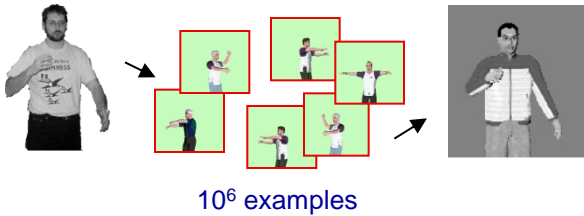
- Dépend de la forme des données (vecteurs, listes, réels/discret) et du type de prédiction
- Exemples
 - Plus proches voisins
 - Machines à vecteurs de supports (SVM)
 - Arbre de décision
 - Ensembles de classifieurs (forêts aléatoires, « boosting »...)
 - Réseaux de neurones
 - Règles/Programmation logique
 - Modèles probabilistes (Réseaux bayésiens, Chaînes ou champs de Markov...)
 - Etc.

Frise historique



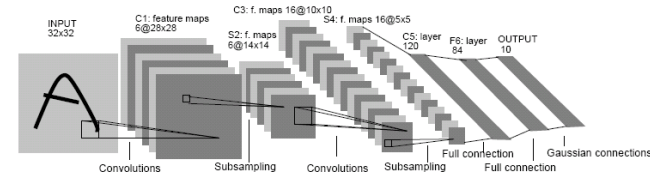
Quelques approches d'apprentissage supervisé en vision

Nearest neighbor



Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

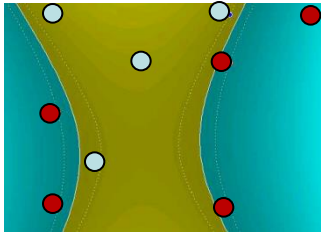
Neural networks (Deep Learning)



LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998

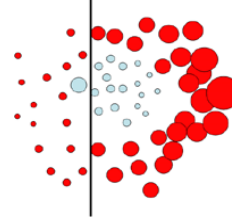
...

Support Vector Machines



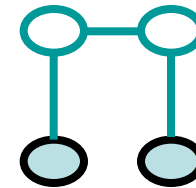
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,...

Boosting



Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,...

Conditional Random Fields



McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
...

Evaluation

- Dépend du type de prédiction
- Classification
 - Taux d'erreur moyen
 - Matrice de confusion
 - Précision/rappel
 - Courbe ROC
- Régression
 - Erreur quadratique
- Détection
 - Taux de recouvrement moyen

Construire son chantier d'apprentissage

- Préparer les données
 - Simplifier/débruiter/formater/homogénéiser
- Diviser en trois ensembles:
 - **Apprentissage** (“Train”) pour optimiser les paramètres du modèle.
 - **Validation** pour évaluer la qualité de l'apprentissage et optimiser les hyper-paramètres
 - **Test** pour estimer la qualité de l'apprentissage dans son contexte d'utilisation, i.e. **l'erreur de généralisation**.
- L'ensemble de test n'est jamais utilisé pour l'apprentissage (optimisation), seulement pour son évaluation.

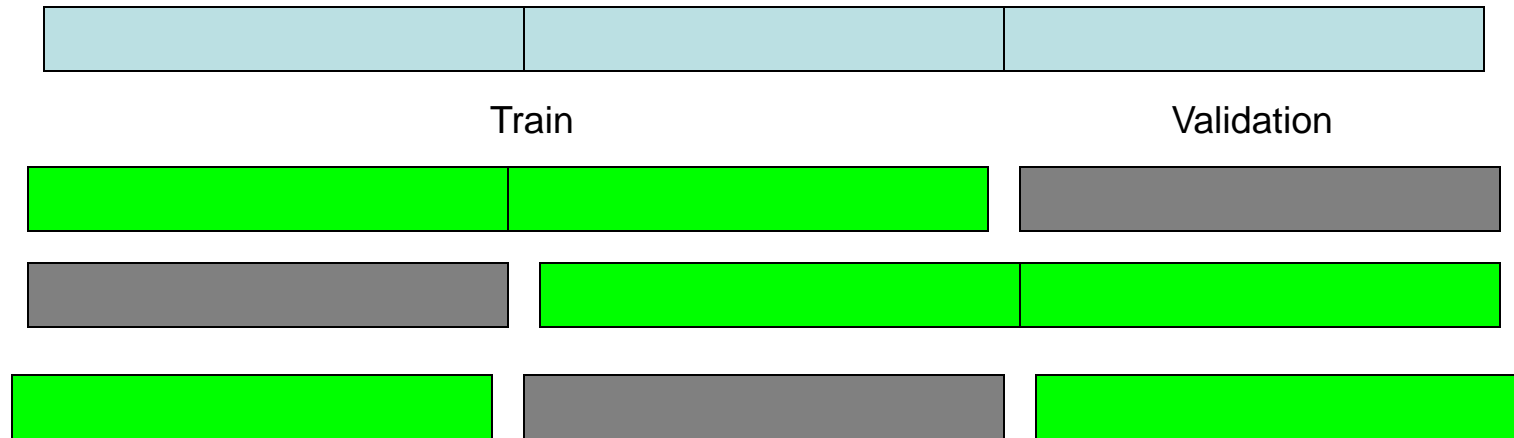
Validation croisée

- Permet d'estimer l'erreur de généralisation à partir des données d'apprentissage (« astuce »)
- Principe:
 - Division des données en k sous ensembles (« fold »)
 - Choix d'une partie comme ensemble de *validation* fictif, les autres comme *train*
 - Apprentissage sur l'ensemble *train*
 - Estimation des erreurs sur *validation*
- On fait tourner l'ensemble de *validation* sur chacune des parties
- L'erreur de généralisation estimée est la moyenne des erreurs sur chaque ensemble de *validation*

Stratégies de partitionnement

- k-fold

Données

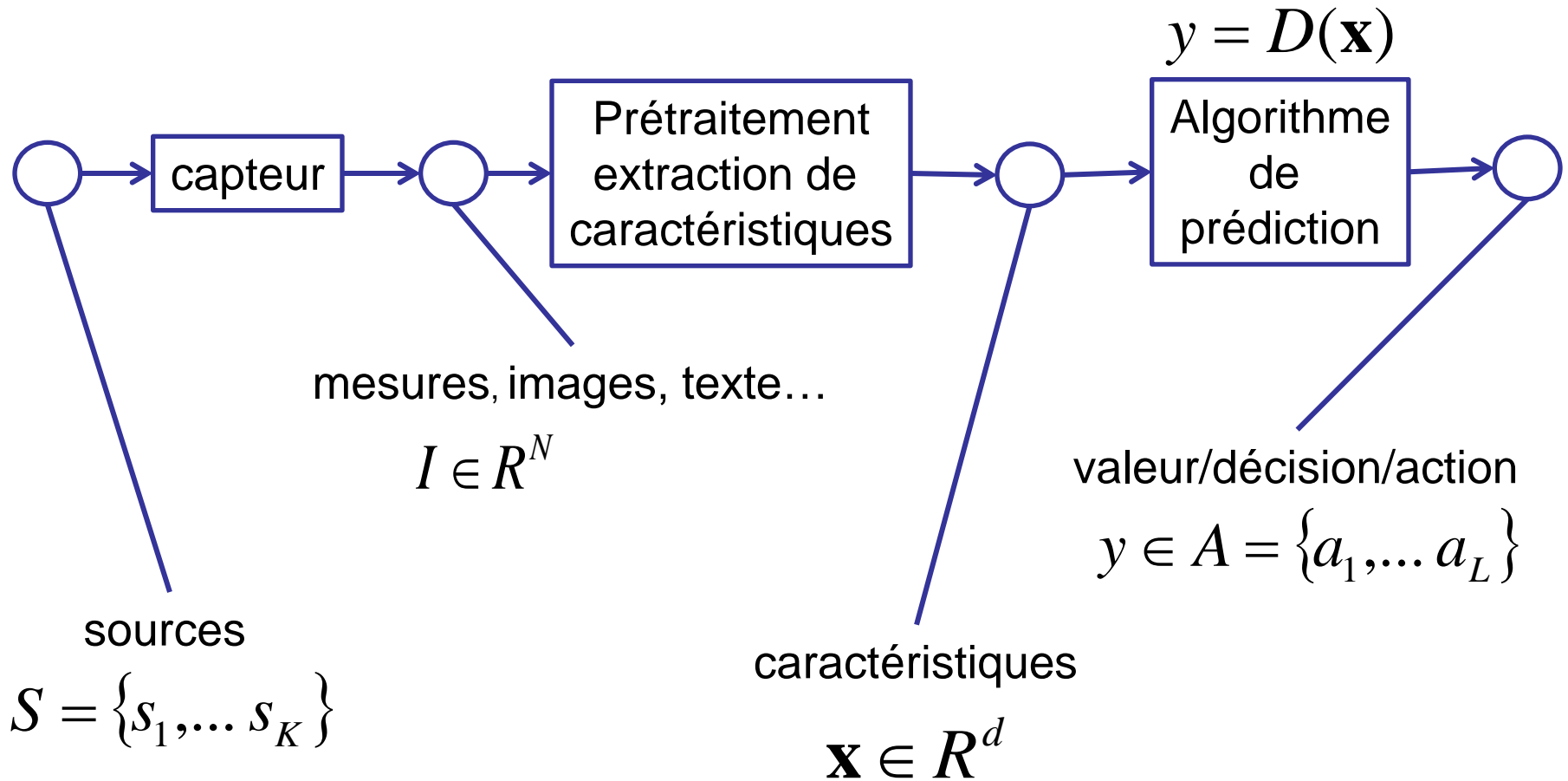


- Leave-one-out



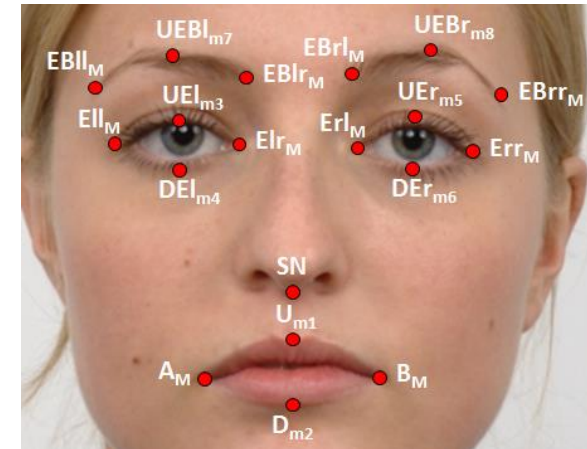
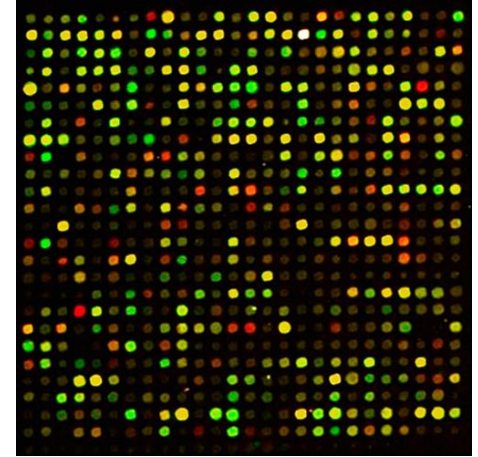
Extraction de caractéristiques

Chaîne de prédiction

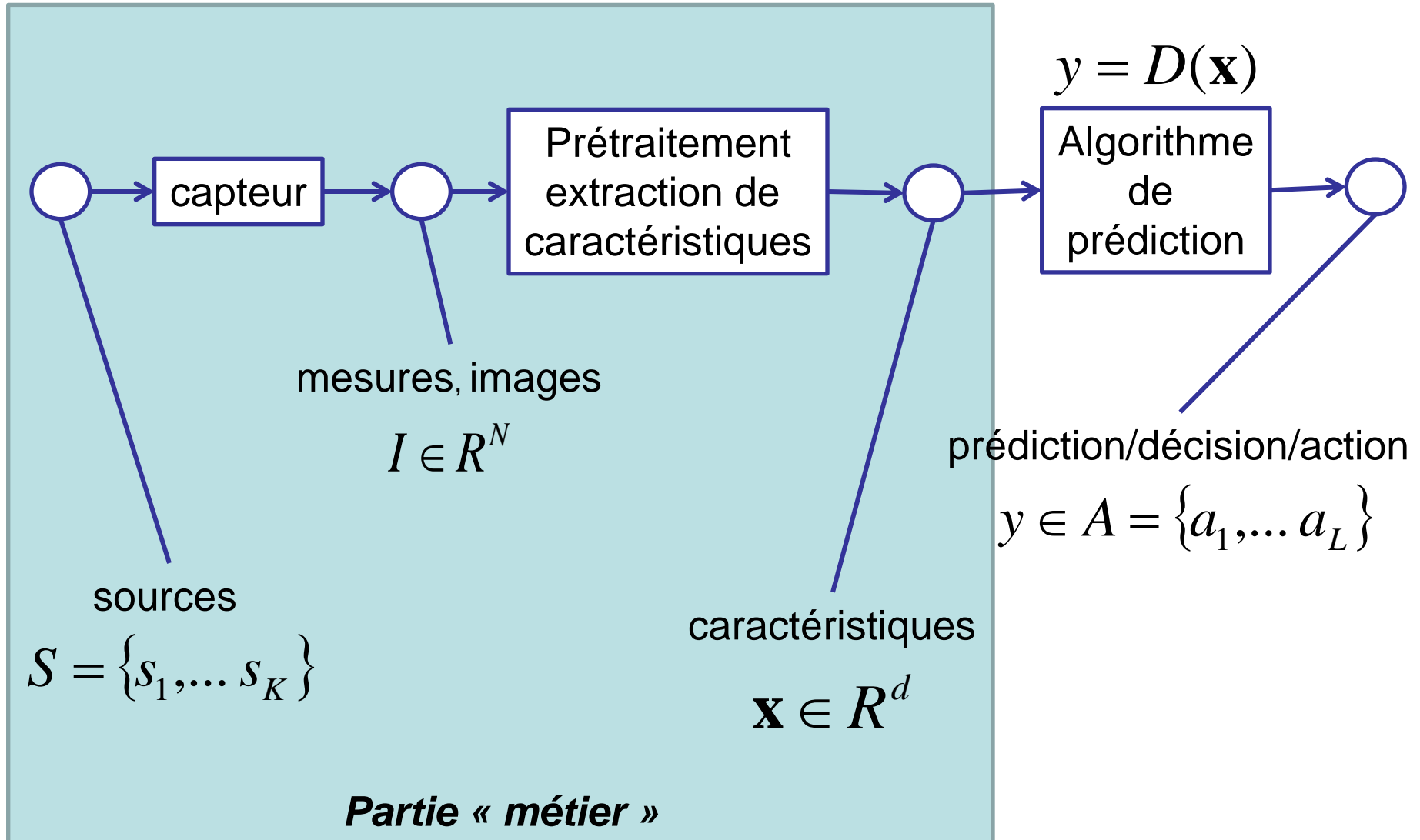


Extraction de caractéristiques

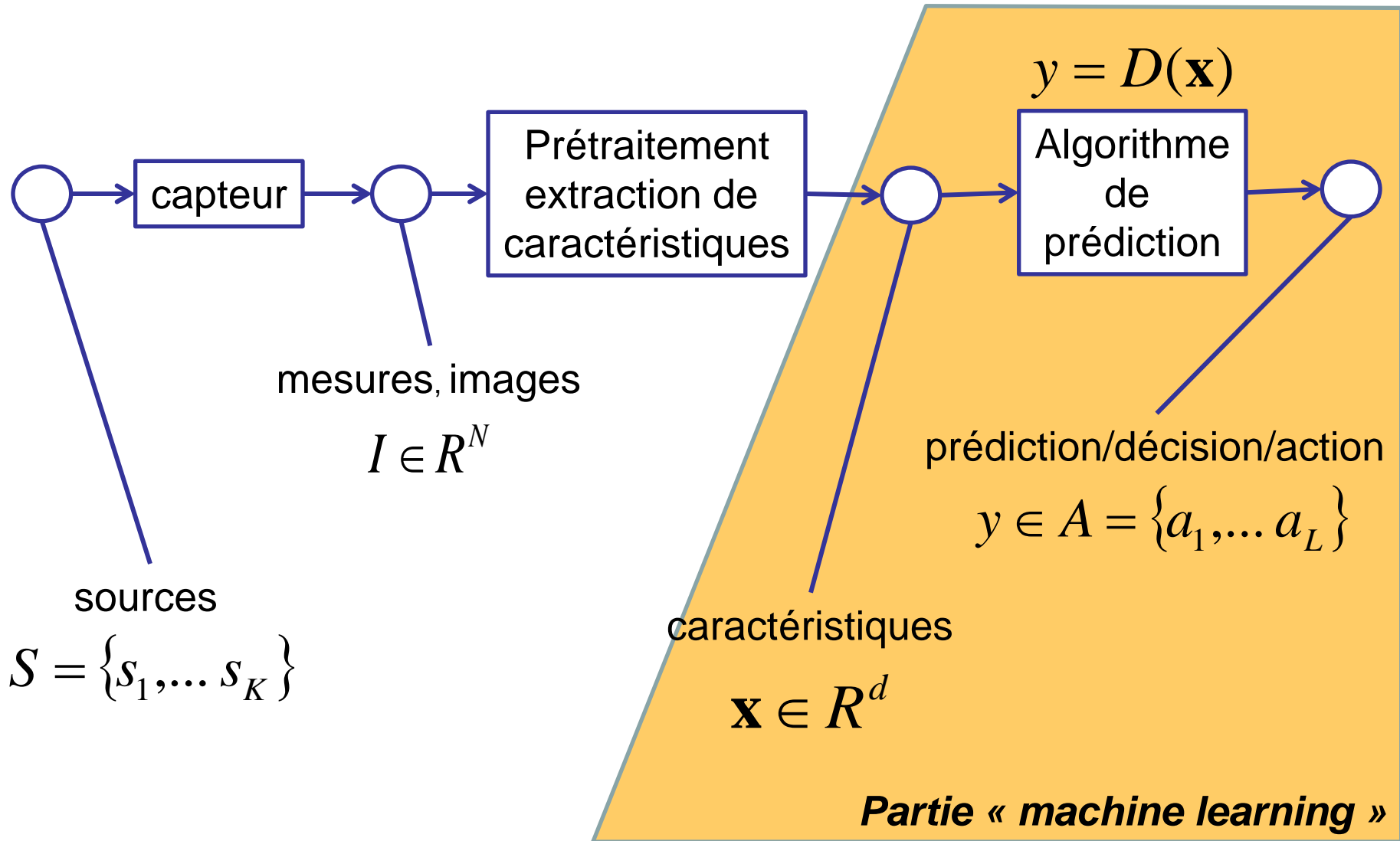
- « Feature extraction » en anglais
- Données brutes pas exploitables directement:
 - Bruitées
 - Trop complexes
 - Information utile noyée
- Deux cas de figure:
 - On sait ce qui est important et pourquoi
→ modélisation
 - On ne sait pas décrire ce qui est important
→ on l'apprend!
- Deux exemples:
 - Microarrays: on ne sait pas quoi extraire
 - Reconnaissance d'émotions: on ne sait pas comment combiner les caractéristiques de visage



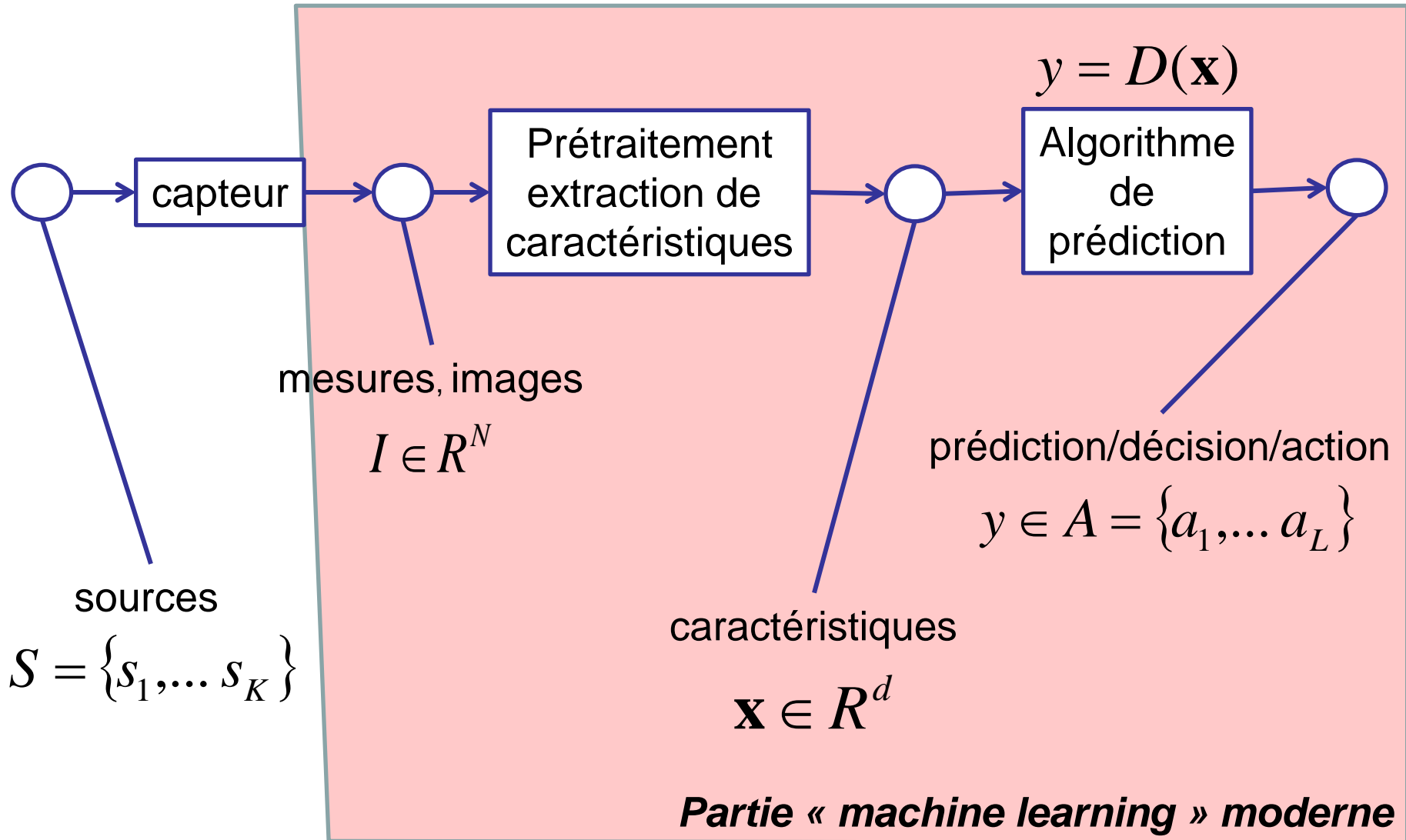
Prédiction: qu'apprend-on?



Prédiction: qu'apprend-on?



Prédiction: qu'apprend-on?



DEUX APPROCHES ÉLÉMENTAIRES

Modélisation bayésienne
Plus proches voisins

Théorie Bayésienne de la décision

- On considère les données x, y comme des variables aléatoires.
- On les modélise par des lois de probabilités:
 - $P(x), P(y)$: lois a priori (ou marginales)
 - $P(x, y)$: loi jointe
 - $P(x | y)$: vraisemblance conditionnelle
 - $P(y | x)$: loi a posteriori
- Classification: $y \in \{1, 2 \dots N\}$ est une étiquette
- On cherche à prédire une unique étiquette y^* à partir de x
$$x \mapsto y^*$$
- Théorie de la décision démontre que le meilleur choix est:

$$y^* = \arg \max_y P(y | x)$$

Théorie Bayésienne de la décision

- Deux questions:
 - Comment calculer $P(y | x)$ = apprentissage
 - Comment trouver le max = prédiction
- « Astuce »: utiliser la loi de Bayes

$$P(y | x) = \frac{P(x | y) P(y)}{P(x)}$$

- On connaît en général la fréquence d'occurrence des classes y
- On sait plus facilement calculer la vraisemblance: $P(x | y)$
 - « Si je sais dans quelle classe je suis, je sais décrire le comportement/distribution de mes données »
- Le max sur y ne dépend que de $P(x | y)$ et $P(y)$

$$y^* = \arg \max_y P(x | y) P(y)$$

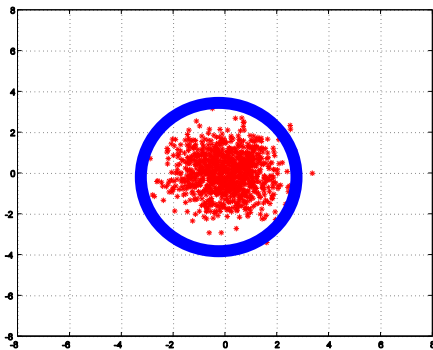
Approche Bayésienne multivariée

- Calcul de la loi conditionnelle: Modèle multivarié
- Par ex. modèle gaussien décrivant $\mathbf{x} = [x_1, x_2 \dots x_d] \in \mathbb{R}^d$:

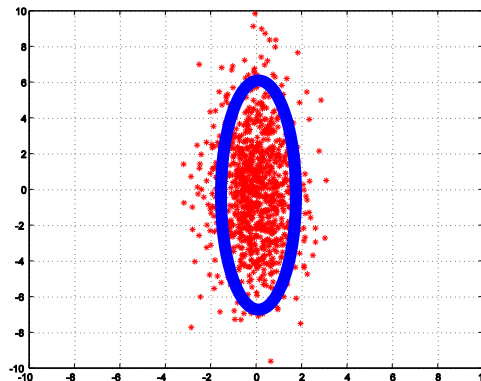
$$P(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} \sqrt{|\Sigma|}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

- Permet de décrire les corrélations entre dimensions.
- Mais demande de connaître la forme des distributions + limitation à petites dimensions.
- Si modélisation gaussienne et deux classes, prédiction se réduit à calculer une fonction de degré 2

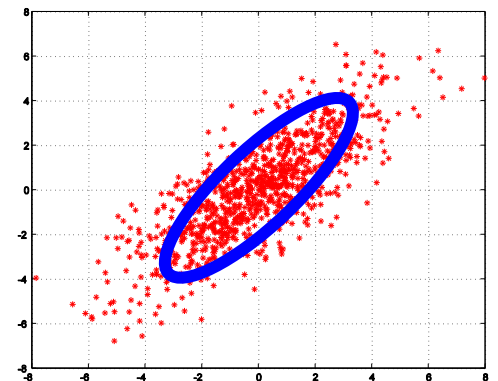
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



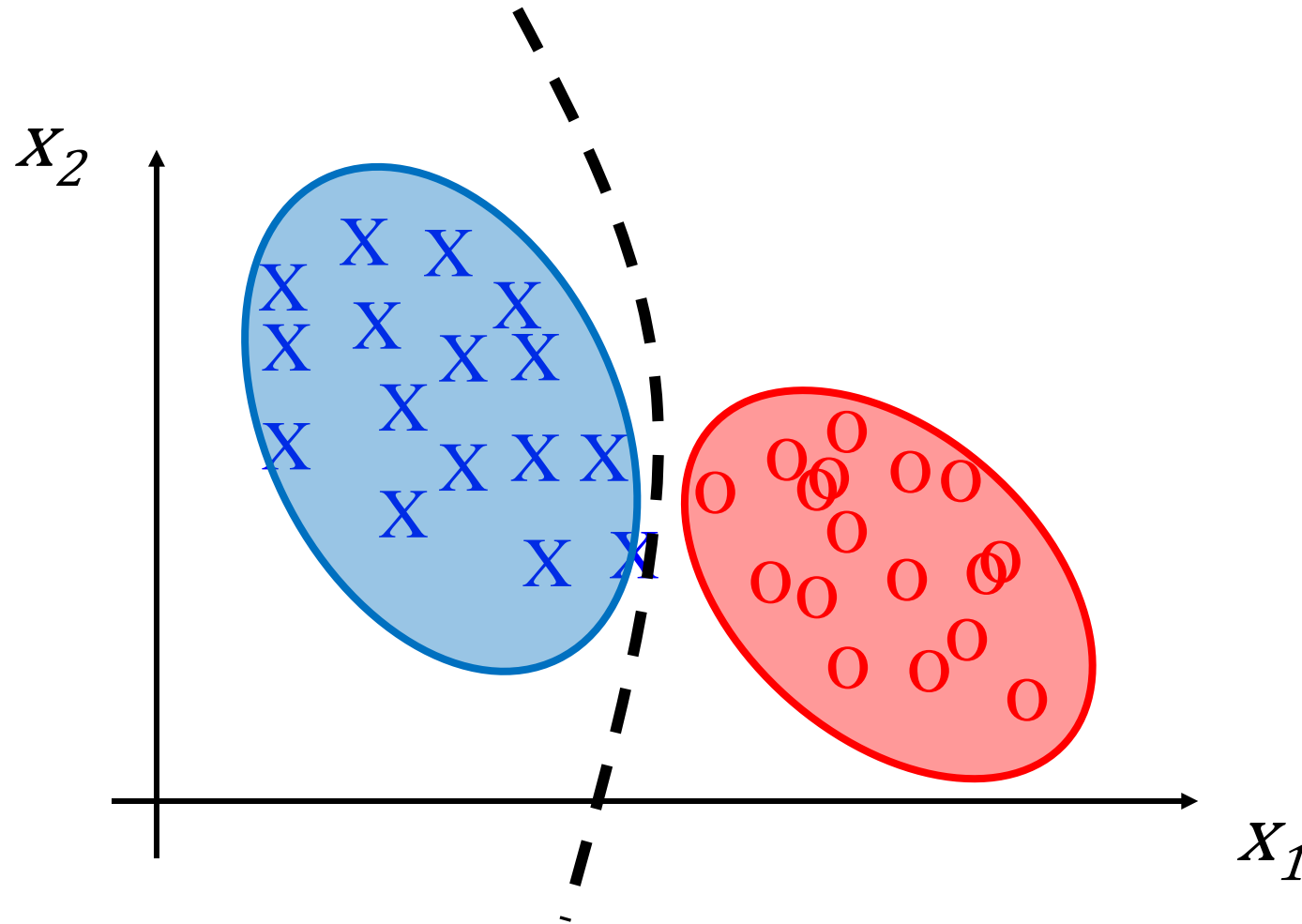
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 9 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix} = R \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix} R^{-1}$$



Approche gaussienne multivariée



Approche Bayésienne Naïve

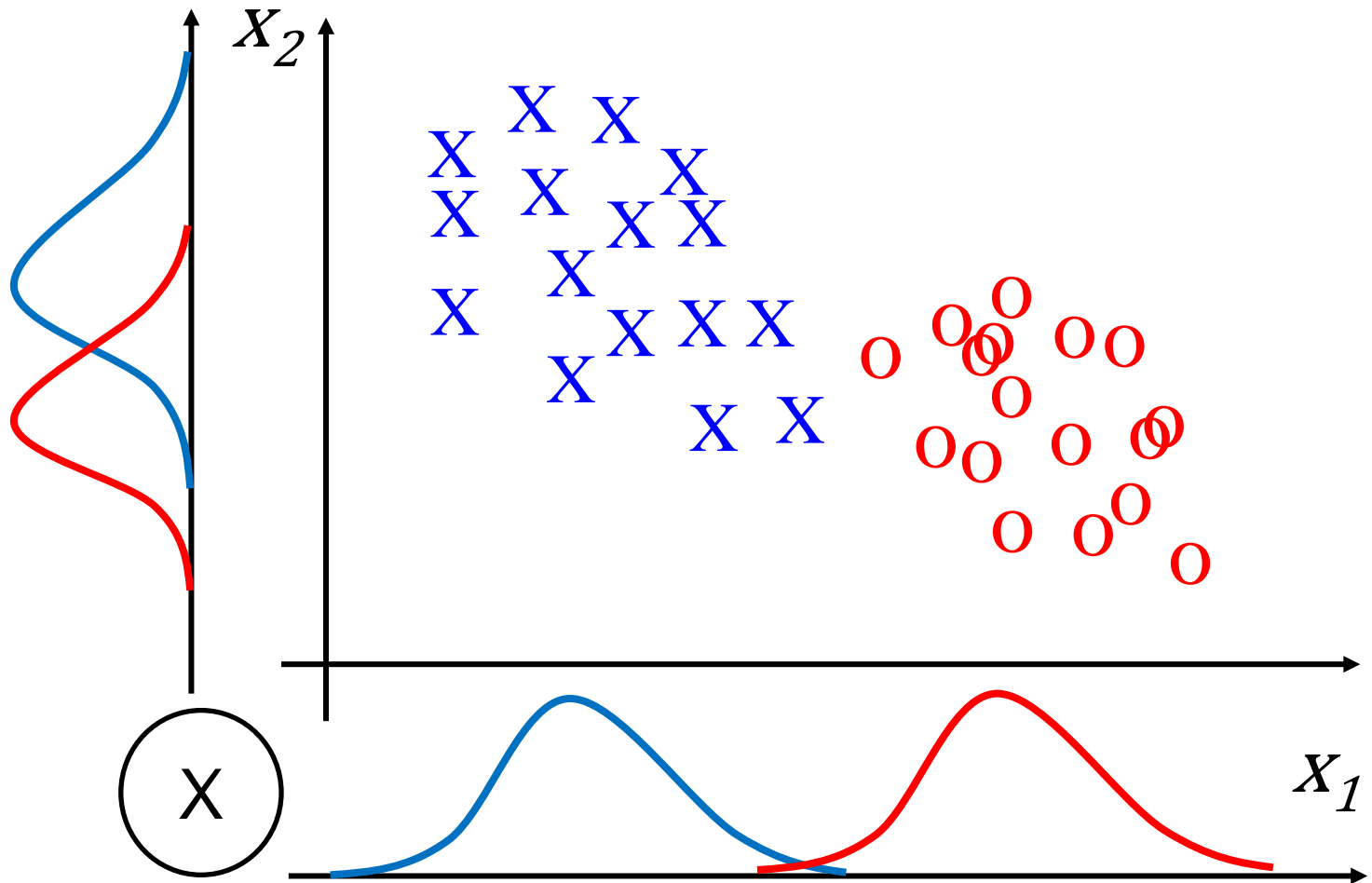
- Calcul de la loi conditionnelle: hypothèse d'indépendance.

$$\begin{aligned}P(x_1, x_2 \dots x_d | y) &= P(x_1 | x_2 \dots x_d, y) P(x_2 \dots x_d | y) \\&= P(x_1 | y) P(x_2 \dots x_d | y) \\&= P(x_1 | y) P(x_2 | y) \dots P(x_d | y)\end{aligned}$$

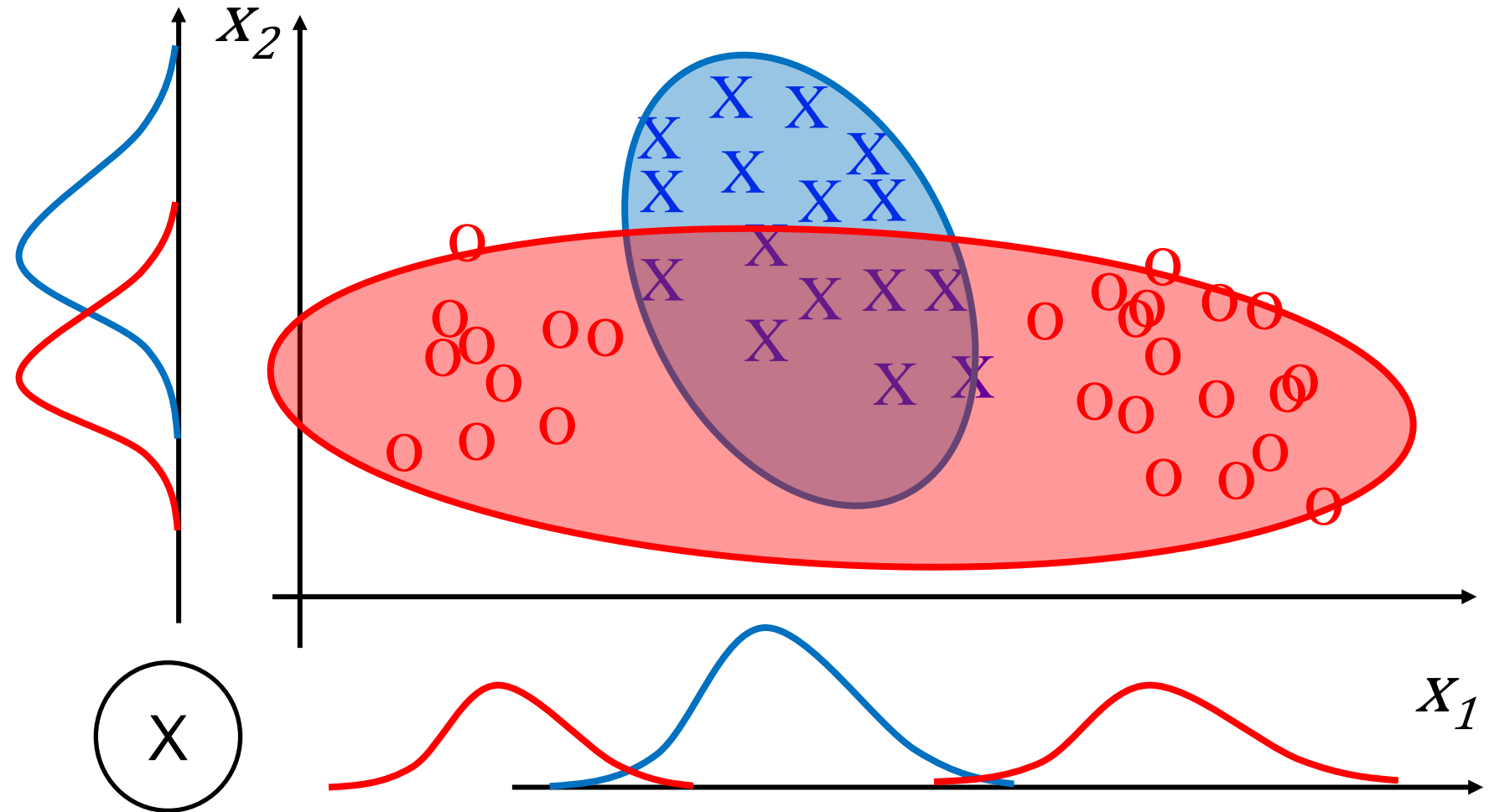
- On calcule la vraisemblance globale dimension par dimension
➔ Problème 1D, modèles plus faciles à estimer (gaussien, binomial, histogrammes, mélange de gaussiennes...)
- En pratique, on calcule plutôt la log-vraisemblance pour des questions de stabilité numérique

$$\begin{aligned}\log P(\mathbf{x} | y) &= \sum_i \log P(x_i | y) \\y^* &= \arg \max_y \log P(\mathbf{x} | y) + \log P(y)\end{aligned}$$

Approche bayésienne naïve



Approche bayésienne naïve vs. multivariée



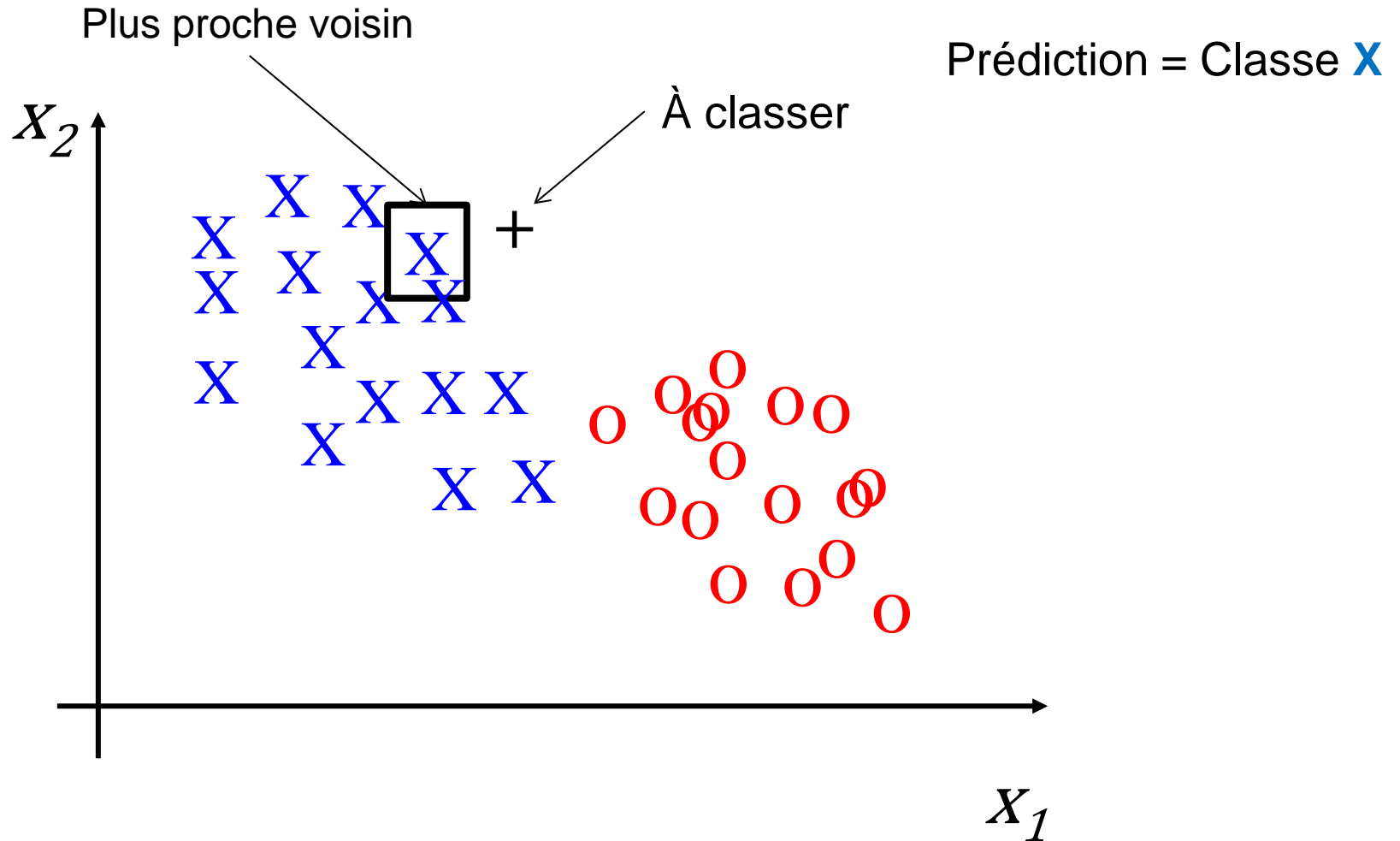
Approche bayésienne: résumé

- Théorie probabiliste de la décision → calcul de la loi a posteriori
- Expression de la loi a posteriori:
 - Hypothèse d'indépendance conditionnelle.
 - Modèle gaussien multivarié
- Apprentissage
 - Estimation de lois paramétriques simples
- Prédiction
 - Calcul de log-vraisemblance et max sur hypothèses
- Quand l'utiliser? (limitations)
 - Petits problèmes bien modélisés (gaussien multivarié)
 - Caractéristiques non corrélées (bayésien naïf, mais ça peut aussi marcher si c'est corrélé)

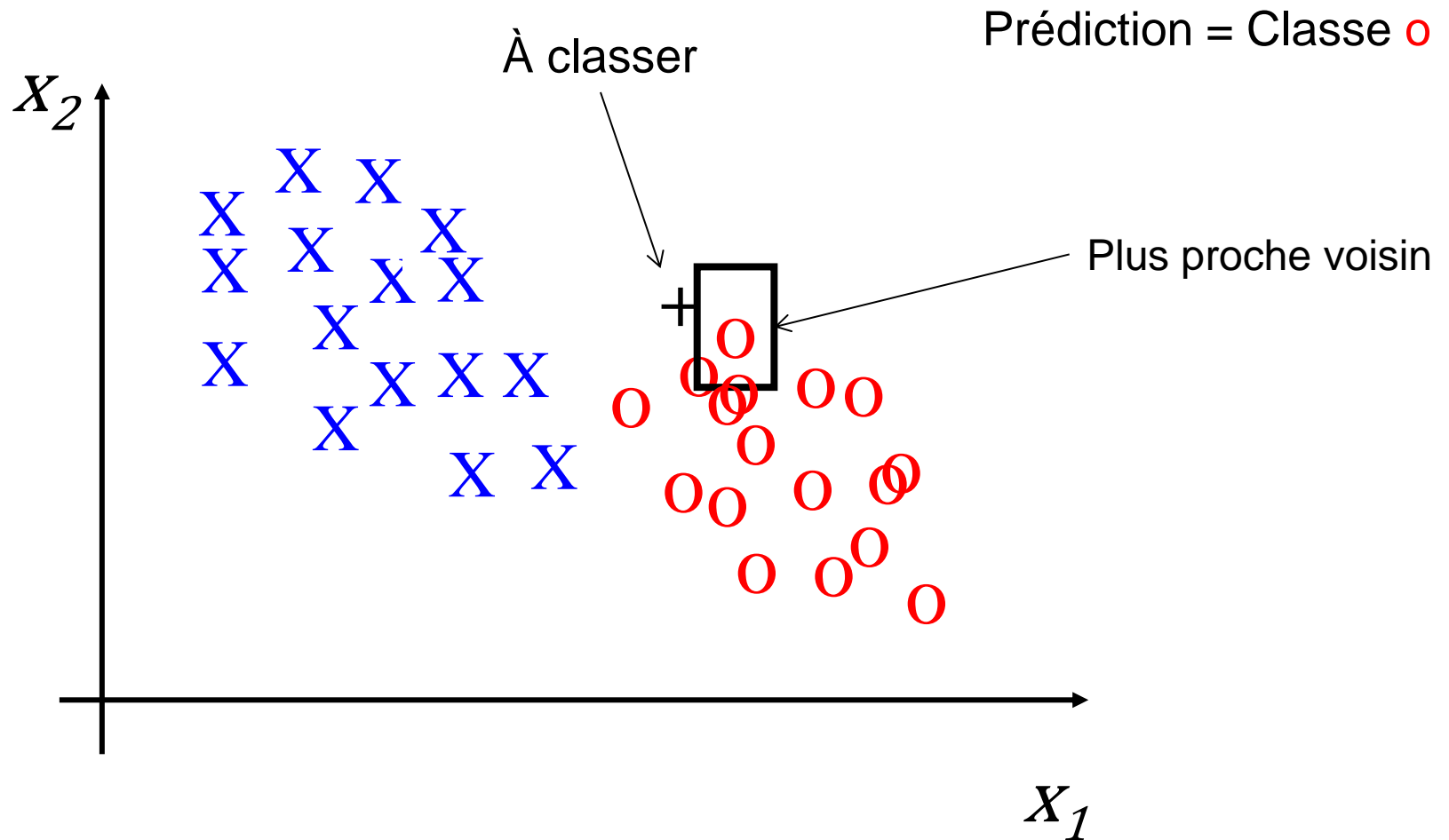
Plus proche(s) voisin(s)

- Principe:
 - Deux échantillons **proches** dans l'espace de représentation ont les mêmes prédictions
 - Pour prédire, il suffit de trouver l'exemple annoté **le plus** proche, et d'associer son annotation (étiquette, valeur...)
- Que veut dire « proche » ?
 - Nécessite la définition d'une métrique ou mesure de similarité $d(\mathbf{x}, \mathbf{x}')$
 - Plusieurs métriques possibles: distance euclidienne (L2), city-block (L1), Minkowski, Mahalanobis...
 - On peut aussi « apprendre » la métrique ou mesure de similarité
- Que veut dire « le plus proche » ?
 - Base d'échantillons annotés $\mathcal{L} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_N, y_N)\}$
 - Recherche de l'échantillon le plus proche: $i^* = \arg \max_i d(\mathbf{x}, \mathbf{x}_i)$
 - Assigne comme prédiction l'annotation du plus proche: $y^* = y_{i^*}$

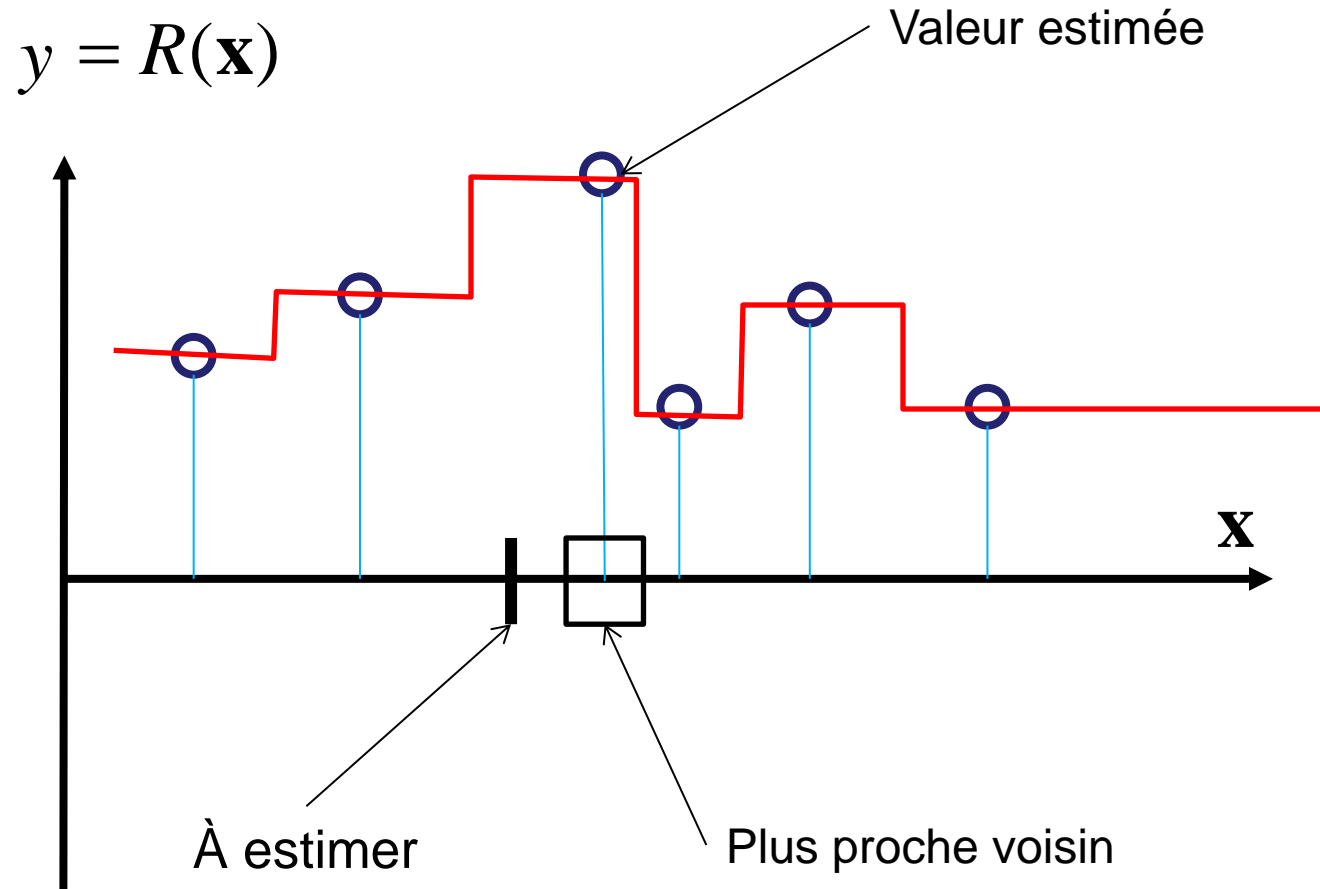
Classification ppv



Classification ppv

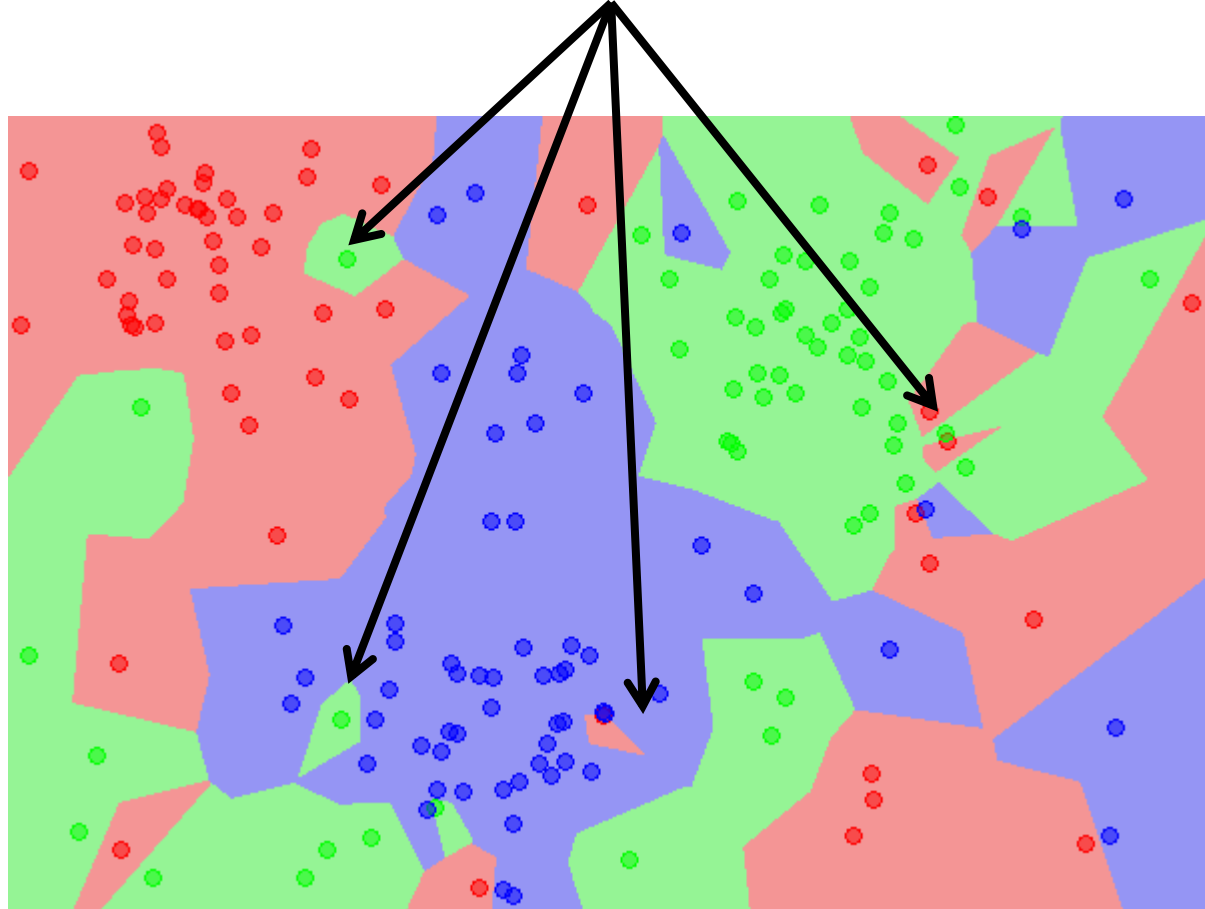


Régression PPV



Fonction de classification

Données bruitées → Régions isolées → mauvaise régularité des prédictions



Chaque échantillon définit une région homogène de l'espace de représentation

k-plus proches voisins (« k-NN »)

- Principe: décision à partir de plusieurs exemples de la base de données d'apprentissage
- On ordonne les échantillons d'apprentissage en fonction de leur distance à la donnée à classer:

$$d(\mathbf{x}, \mathbf{x}_{(1)}) \leq d(\mathbf{x}, \mathbf{x}_{(2)}) \leq \dots \leq d(\mathbf{x}, \mathbf{x}_{(N)})$$

- On choisit les k plus proches
- On prédit en choisissant la classe recueillant le plus de votes

$$y^* = \arg \max_y \sum_{i=1}^k \delta(y, y_{(i)})$$

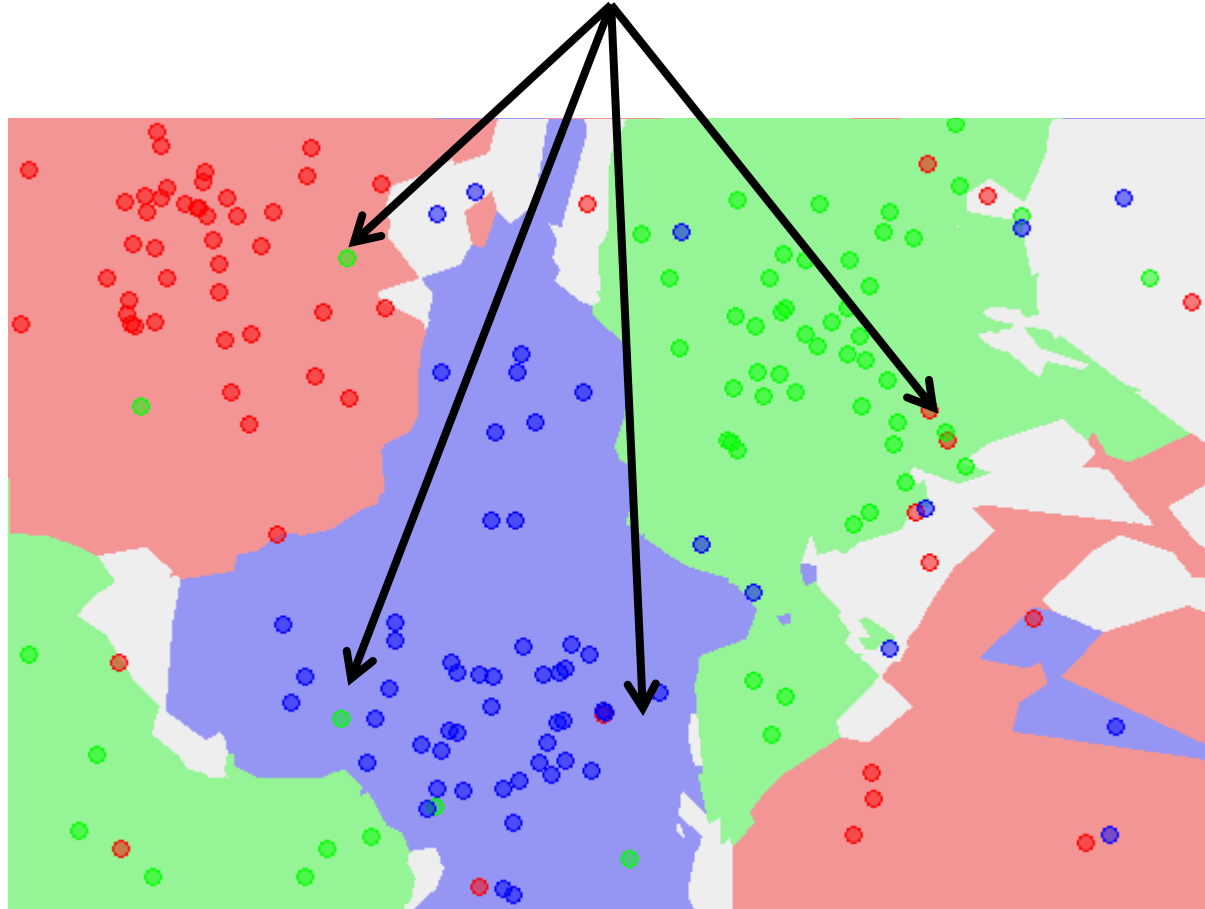
Où δ est la fonction de Kronecker (elle vaut 1 si égal, 0 sinon)

- Si pas de max (ambiguïté sur la prédiction) on ne décide pas!
- On peut aussi pondérer les votes:

$$y^* = \arg \max_y \sum_{i=1}^k K(\mathbf{x}, \mathbf{x}_{(i)}) \delta(y, y_{(i)})$$

Fonction de classification 5 ppv

Données bruitées → Régions isolées → mauvaise régularité des prédictions



Chaque échantillon définit une région homogène de l'espace de représentation

Propriétés statistiques

Bornes statistiques asymptotiques ($N \rightarrow \infty$)

$$E \leq E_{kNN} \leq E \left(2 - \frac{LE}{L-1} \right)$$

Où E est l'erreur théorique optimale (Bayes), L est le nombre de classes et E_{kNN} est l'erreur des k-ppv.

« L'erreur du k-NN est au plus deux fois moins bonne que l'erreur minimale. »

Coût de la prédiction du k-ppv

- Calcul de la prédiction dépend pour chaque exemple x d'un calcul + tri par rapport aux N exemples de la base:

$$d(x, x_{(1)}) \leq d(x, x_{(2)}) \leq \dots \leq d(x, x_{(N)})$$

- Pour N et d grands, coût important de la recherche exhaustive $O(Nd)$. Il existe:
 - Des algorithmes efficaces de recherche pour problèmes de tailles moyennes (KDtree)
J. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transaction on Mathematical Software*, vol. 3, no. 3, pp. 209–226, 1977.
 - Des algorithmes d'approximation pour les grandes bases ($>10^6$).
Jegou, H., Douze, M., & Schmid, C. (2011). Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1), 117-128.
- Autre manière: pré-calculer les surfaces de séparation entre classes. La complexité de prédiction est alors liée à la complexité de la surface et/ou de son approximation. On verra comment d'autres approches permettent de l'estimer directement.

La malédiction des grandes dimensions

- Lorsque la dimension d de l'espace de représentation augmente, les points sont tous aussi proches ou aussi loin.
- On peut montrer, pour une distribution quelconque de N points tirés de manière indépendante dans \mathbb{R}^d , que:

$$\lim_{d \rightarrow \infty} E \left[\frac{d_{max} - d_{min}}{d_{min}} \right] = 0$$

- Ce n'est plus vrai si les points sont corrélés...heureusement!
- On peut interpréter les techniques de Machine Learning comme des moyens de repérer les bonnes corrélations entre données.
- Conséquence pour les approches « plus proches voisins »:
 - Ca ne marche que pour les faibles dimensions
 - Ou il faut **réduire** les dimensions de représentation avant de calculer les distances.

Comportement des PPV

- Avantages
 - Schéma flexible, facile à mettre en œuvre, dépendant de la définition d'une similarité entre données.
 - Bonnes propriétés statistiques ($N \rightarrow \infty$)
- Mais...
 - Temps de calcul prohibitif pour grandes bases
 - Algorithmes efficaces de recherche optimaux ou sous-optimaux
 - Régularité dépend des données, pas de l'apprentissage
 - Le k-PPV (« kNN ») pour lisser et réduire le bruit
 - Malédiction des grandes dimensions (« Curse of dimensionality »)
 - Réduire la dimension de représentation (cf. cours N°4)

« Plus proches voisins »: résumé

- Hypothèse de régularité = Si observations proches, même comportement
- Deux questions:
 - Que veut dire « proche »?
 - Comment trouver les plus proches?
- Apprentissage
 - Aucun
- Prédiction
 - Tri des distances aux échantillons + vote
- Quand l'utiliser? (limitations)
 - Efficace sur petits problèmes (dimensions & nombre d'exemples)
 - Pb du « curse of dimensionality » + temps de calcul
 - Disposer d'une mesure de similarité adaptée aux données

Références

- K. Fukunaga, Introduction to Statistical Pattern Recognition (Second Edition), Academic Press, New York, 1990.
- P.A. Devijver and J. Kittler, Pattern Recognition, a Statistical Approach, Prentice Hall, Englewood Cliffs, 1982)
- R.O. Duda and P.E. Hart, Pattern classification and scene analysis, John Wiley & Sons, New York, 1973.
- L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, Classification and regression trees, Wadsworth, 1984.
- S. Haykin, Neural Networks, a Comprehensive Foundation. (Macmillan, New York, NY., 1994)
- L. Devroye, L. Györfi and G. Lugosi, A Probabilistic Theory of Pattern Recognition, (Springer-Verlag 1996)
- V. N. Vapnik, The nature of statistical learning theory (Springer-Verlag, 1995)
- C. Bishop, Pattern Recognition and Machine Learning, (Springer-Verlag, 2006).
- Jerome H. Friedman, Robert Tibshirani et Trevor Hastie, The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Springer-Verlag 2009).
- Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, An MIT Press book (<http://www.deeplearningbook.org>)
- Kevin Murphy, Machine Learning: a Probabilistic Perspective, (MIT Press, 2013)
- Hal Daumé III, A Course in Machine Learning (<http://ciml.info/>)

Bases de données

- UCI Repository: <http://www.ics.uci.edu/~mlearn/MLRepository.html>
- UCI KDD Archive: <http://kdd.ics.uci.edu/summary.data.application.html>
- Statlib: <http://lib.stat.cmu.edu/>
- Delve: <http://www.cs.utoronto.ca/~delve/>
- Kaggle: <https://www.kaggle.com/>
- Benchmarks (Vision):
 - ImageNet: <http://image-net.org/>
 - MS COCO: <http://cocodataset.org/>
 - MNIST et plus:
http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html
 - CV on line: <https://computervisiononline.com/datasets>

Journaux

- Journal of Machine Learning Research www.jmlr.org
- Machine Learning
- Neural Computation
- Neural Networks
- IEEE Transactions on Neural Networks
- IEEE Transactions on Pattern Analysis and Machine Intelligence
- Annals of Statistics
- Journal of the American Statistical Association
- ...

Conférences

- International Conference on Machine Learning (ICML)
- European Conference on Machine Learning (ECML)
- Neural Information Processing Systems (NIPS)
- Uncertainty in Artificial Intelligence (UAI)
- Computational Learning Theory (COLT)
- International Joint Conference on Artificial Intelligence (IJCAI)
- International Conference on Neural Networks (Europe)
- Conference of the American Association for Artificial Intelligence (AAAI)
- IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
- European Conference on Computer Vision (ECCV)
- International Conference on Computer Vision (ICCV)
- IEEE International Conference on Data Mining (ICDM)
- ...

A retenir pour aujourd'hui

- Apprentissage = programmer à partir des données
- Plusieurs types d'apprentissage
- Démarche générique:
 - Constitution d'une base d'apprentissage
 - Analyse préliminaire des données
 - Conception du modèle
 - Optimisation
 - Evaluation
- Deux approches élémentaires:
 - Modélisation bayésienne
 - Plus proches voisins

Le TP1

- Partie 1: Les deux approches élémentaires sur une première base
 - Programmation Python
 - Application de la démarche
- Partie 2: Utilisation de la bibliothèque scikit-learn
 - Les deux approches sur une autre base
- TP non noté, mais à m'envoyer par mail: stephane.herbin@onera.fr
- Correction par retour de mail si envoi avant semaine prochaine
- Le prochain TP sera noté!

Utilisation des « Notebook Python »

td2_classification_supervisee - Mozilla Firefox

td2_classification_sup

localhost:8888/notebooks/td2_classification_sup

jupyter td2_classification_supervisee (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

plt.show()

après chaque fonction de visualisation.

Activité 1.1 : Mon premier classifieur.

Repérer les paramètres utiles des fonctions et la manière de les utiliser. Lancer une première chaîne de calcul pour apprendre un classifieur linéaire pour le jeu de données 0 et des valeurs par défaut. Visualiser les résultats et calculer l'erreur sur les jeux d'apprentissage et de test avec la fonction `score`. Utiliser la fonction `predict` et comparer les sorties produites avec les vraies valeurs. Vérifier avec la fonction `score` que les valeurs d'erreur sont les mêmes. Recommencer la séquence d'apprentissage avec les autres distributions de données (1 à 3).

```
[1]: # Librairies utiles standard
import numpy as np
import matplotlib.pyplot as plt

# Pour visualiser et récupérer les données
import iogs_td_util as td

# L'algorithme SVM dans la bibliothèque scikit-learn
from sklearn import svm

import random
```

```
In [2]: # Classifieur
svc = svm.SVC(kernel='linear', max_iter=-1)

# Premier jeu de données
trainX, trainY, testX, testY=td.generate_data(0)
```

Texte

Exécution des
cellules

Code

Utilisation des « Notebook IPython »

- Copier et décompresser dans votre environnement l'archive ensta_s203_td1.zip
- Ouvrir un terminal et:
 - Positionner l'environnement anaconda: source activate root
 - Se placer dans le répertoire où l'archive est décompressée
 - Ouvrir un navigateur (firefox)
 - Taper en ligne de commande: jupyter notebook
- Vérifier que le navigateur ouvre une page Jupyter
- Lancer le notebook td1_knn_bayesien.ipynb en cliquant sur le lien
- Modifiez directement le notebook, et sauvegardez-le régulièrement: c'est votre compte-rendu !

Liste (à vérifier)

"caio garcia-cancian" <caio.garcia-cancian@ensta-paristech.fr>;
"Ferreira Marcelo" <marcelo.ferreira@ensta-paristech.fr>;
"chuan qin" <chuan.qin@ensta-paristech.fr>;
"Fray Ribeiro Lais" <lais.fray-ribeiro@ensta-paristech.fr>;
"maxime chor" <maxime.chor@ensta-paristech.fr>;
"Zhi Zhou" <zhi.zhou@ensta-paristech.fr>;
"Chen Hao" <hao.chen@ensta-paristech.fr>;
"pierre-louis rerolle" <pierre-louis.rerolle@ensta-paristech.fr>;
"kaori watanabe" <kaori.watanabe@ensta-paristech.fr>;
"Fabian Antonio Rincon Vija" <fabian-antonio.rincon-vija@ensta-paristech.fr>;
"chaima adjif" <chaima.adjif@ensta-paristech.fr>;
"ghassen ben-hattab" <ghassen.ben-hattab@ensta-paristech.fr>;
"ahmed-aziz yedes" <ahmed-aziz.yedes@ensta-paristech.fr>;
"amal hasni" <amal.hasni@ensta-paristech.fr>;
"Yanis Kemmar" <yanis.kemmar@ensta-paristech.fr>;
"Thales Loiola Raveli" <thales.loiola-raveli@ensta-paristech.fr>;
"Juan Pablo SANCHEZ PEREZ" <juan-pablo.sanchez-perez@ensta-paristech.fr>;

Le projet

- A réaliser en binôme si vous voulez
- Un rapport à fournir
- Deux options: « kaggle » ou « biblio »
- KAGGLE
 - Une compétition à choisir sur le site <https://www.kaggle.com>
 - Produire un fichier de résultat et l'évaluer
 - Justifier la démarche
 - ATTENTION: vérifiez que vous aurez les ressources de calcul
- BIBLIO
 - Un ensemble d'articles de type recherche à analyser sur un thème
 - Possibilité de reproduire les résultats si code disponible (bonus)
 - Pour les futurs chercheurs
- Décision pour la semaine prochaine...