Apprentissage

Automatique Introduction

A.Larhlimi

Aujourd'hui

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

« Machine Learning »

- Un domaine scientifique hybride:
  - Statistique
  - Intelligence artificielle
  - « Computer science »
  - Traitement du signal

- Utilisant des techniques généralistes:
  - Optimisation numérique
  - Hardware
  - Gestion de base de données

Pourquoi le « Machine Learning »?

- Thème à la mode: Intelligence Artificielle, « deep learning », « big data »…
- Raison épistémologique

- On ne sait pas modéliser les problèmes complexes … mais on dispose d'exemples en grand nombre représentant la variété des situations
- « Data driven » vs. « Model Based »

- Raison scientifique
  - L'apprentissage est une faculté essentielle du vivant

- Raison économique
  - La récolte de données est plus facile que le développement d'expertise

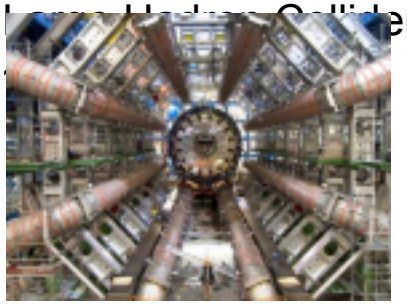
Domaines techniques utilisant du ML

- ML comme outil de conception
  - Vision & Reconnaissance des formes
  - Traitement du langage
  - Traitement de la parole
  - Robotique
  - « Data Mining »
  - Recherche dans BDD
  - Recommandations
  - Marketing…
- ML comme outil explicatif
  - Neuroscience
  - Psychologie
  - Sciences cognitives

Données = carburant du ML

DKRZ (Climat)
500 Po

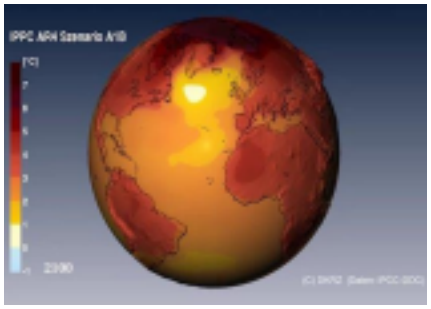
CERN /
Large Hadron Collider



Copernicus :
> 1Po/an

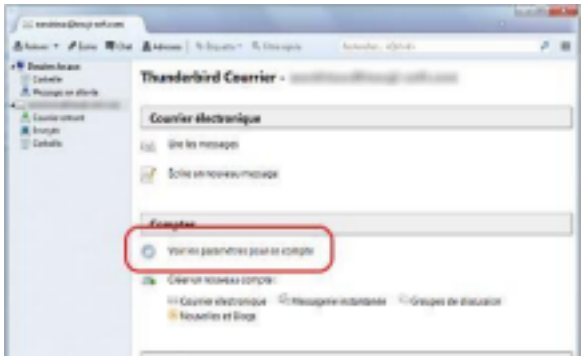
Google :
24 PetaOctets/jour

Square Kilometer Array 1376 Po/an (en 2024)



BIG DATA

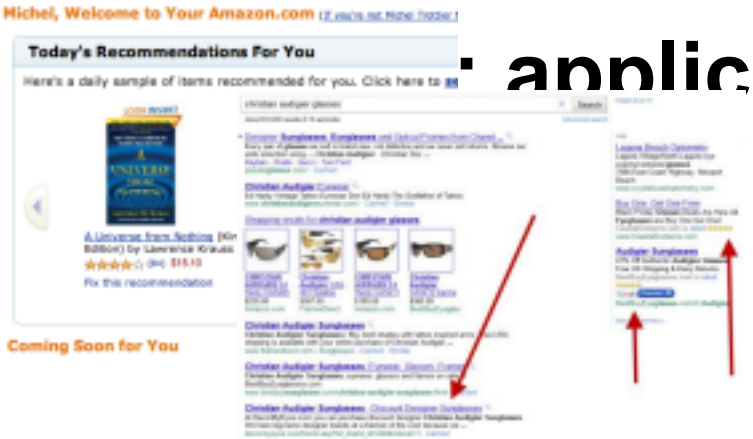
Apprentissage automatique : applications Anti-Spam (*Classifieur Bayesien*)



1997 : DeepBlue bat Kasparov
2017: Alpha GO bat Ke Jie
2019: AlphaStar champion de StarCraft

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

Apprentissage au[illegible] applications



Recommandation ciblée

(*régression logistique*)

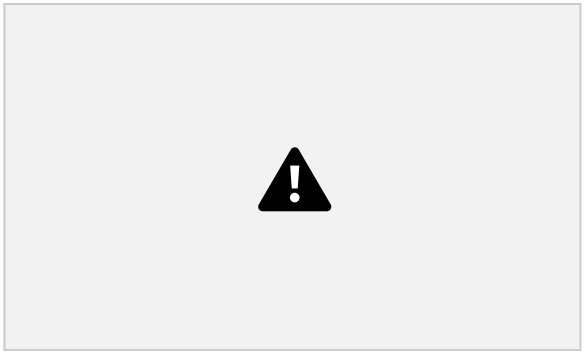
Appareil photo avec détection

de visages (*boosting*)

Apprentissage autom tions

Chat Bots
(*Réseaux de neurones*)

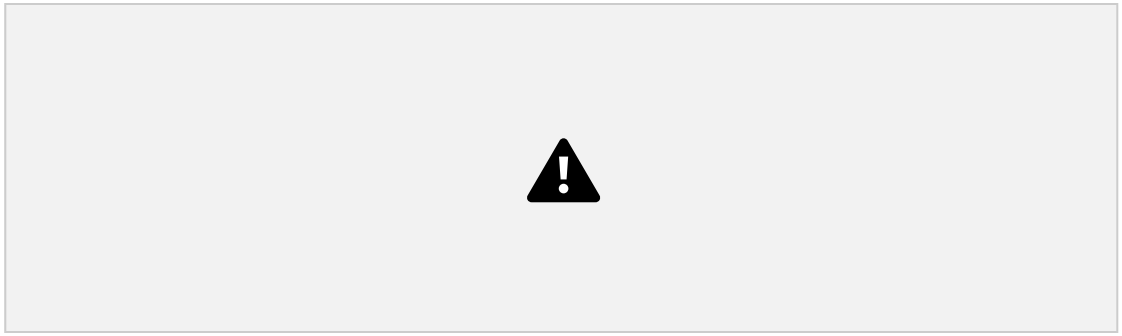
Diagnostic médical
(*Réseaux de neurones*)



Traduction multi-lingue
(Réseaux de neurones)

« Deep Learning » : le mot clé

inévitable



Données

Algorithmes

apprentissage

Moyens de calcul

Logiciels

Une rupture scientifique et technologique en

MACHINE LEARNING

Problématique

générale

Dans ce cours

L'apprentissage automatique est:

- une démarche de **conception** d'une **fonction de prédiction**

- par une modélisation ou programmation **non explicite** à partir **d'exemples** (signaux, images, texte, mesures…)

Formalisation

- Donnée à interpréter (x)
  - Mesures, texte, image, enregistrement, vidéo ou caractéristiques extraites de …
- Prédiction (y)
  - Décision, choix, action, réponse, préférence, groupe,

commande, valeur…

- Echantillons (𝒟 = { 𝒟[redacted], 𝒟[redacted]})
  - Exemples de données et de (bonnes) prédictions
  - « Base d'apprentissage »: 𝒟

Formalisation

Donnée à

Echantillons
exemples

Prédicteur

- Hypothèse forte: les échantillons contiennent toute l'information exploitable et utile
- Prédicteur = « interpolateur » à partir des données ��

Deux phases

Prédiction

Donnée

Programme ou Modèle

Inférence

Résultat

Apprentissage

données

Programme ou Modèle

Estimation Base de

Deux phases

Prédiction

$$??, ?? \mapsto ??$$

Apprentissage

$$\square\square \mapsto \square\square$$

Caractéristiques du prédicteur:
Paramètres,
Poids,
Prototypes…

Formalisation

Fonction paramétrique de prédiction

$$\hat{y} = f(x; \theta)$$

Apprentissage = trouver le θ qui optimise un critère C

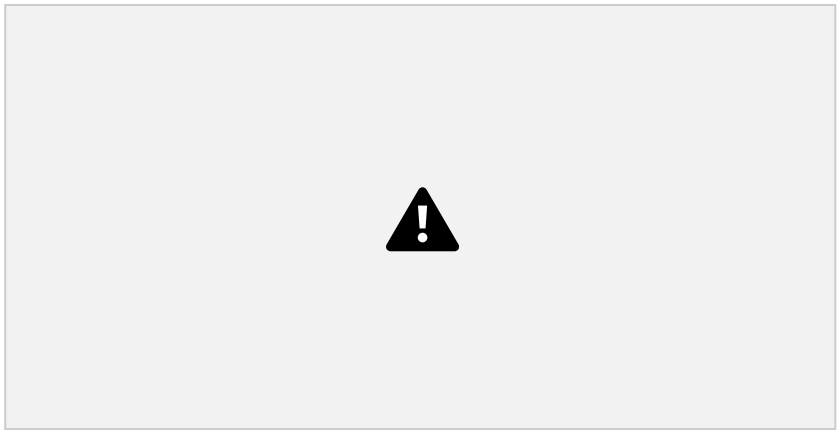
$$\theta = \arg\min_{\theta'} C(\theta, \theta')$$

A partir d’une base d’apprentissage

$$\mathcal{E} = \{ \mathcal{E}_1, \mathcal{E}_2 \}$$

Exemple: Reconnaissance de chiffres

manuscrits



- Comment définir les éléments ?

$\square\square, \square\square, \square\square, \square\square$

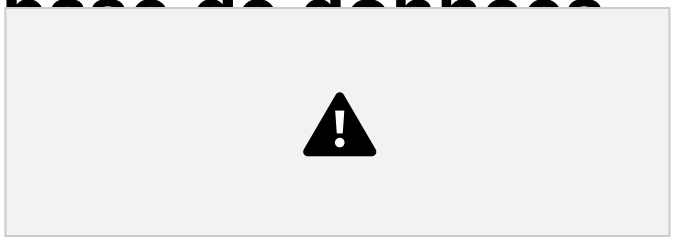
- Les fonctions d'apprentissage et de prédiction?

$\square\square \mapsto \square\square$

$\square\square, \square\square \mapsto \square\square$

Etape 1: choix de la base de données

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



- Il faut la construire:

• Recueil de données existantes •

Expérimentations (photos, mesures…)

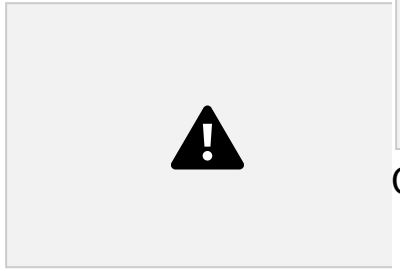
Etape 2: mise en forme des données

Image (2D)

Extraction de caractéristiques

grande dimension, bruitée, hétérogène

Pré-traitement



Profils

Cavités

0
0
0
0
0
0
1
1

0
0
0
1
0
0
0
1
1

[Dine et al., 2017]
https://doi.org/10.1007/978-3-319-46568-5_17

petite dimension, homogène, « propre »

Occupation
de zones

⬚⬚ = Vecteur

Etape 3: choix de l'approche

- Quel type de fonction et de problème d'apprentissage?
  - Classification
  - On connait 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

- Nature fonctionnelle des prédicteurs?
  - Arbres de décision, SVM, Réseaux de neurones…

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 un mesure de qualité (« reward »)

Types de prédictions

- **Classification**

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

- **Régression**
  - Prédiction de température, de cours de bourse
  - Localisation d'objet dans image
  - Commande
- **Structure**
  - Graphe des articulations d'une personne
- **Regroupement**
  - Photos dans base de données personnelle
- **Texte**
  - « C'est un chat qui saute sur une table. »

Nature fonctionnelle 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…)

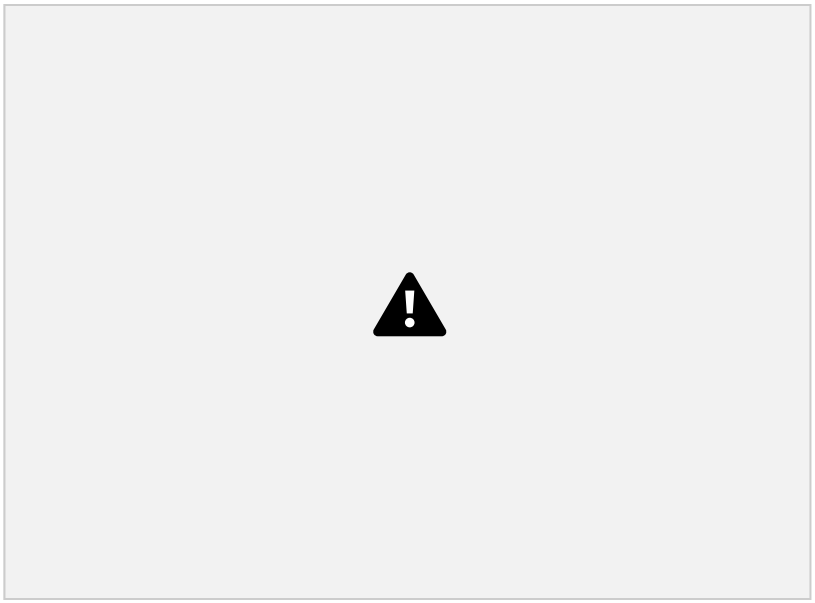
• Etc.

Etape 4: optimisation

Apprentissage =

- définir un espace fonctionnel et un critère paramétrique (coût, énergie…)
- appliquer un optimiseur et régler ses paramètres •

vérifier que l'apprentissage se passe bien • évaluation de la capacité de généralisation • convergence

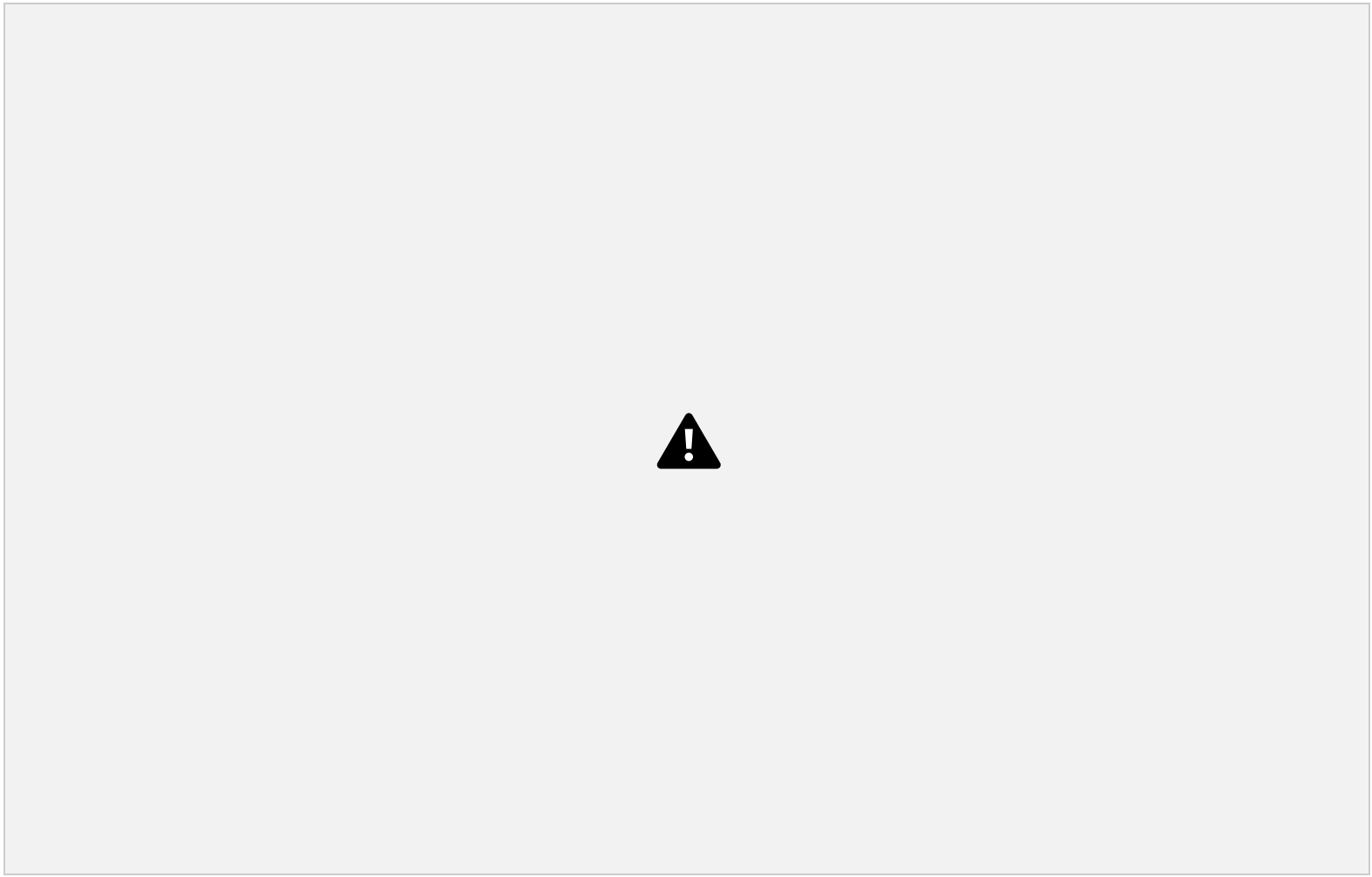


Optimisation

- **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

Etape 5: évaluation



Métriques d'évaluation

- 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

Résumé des étapes de conception

1. Constituer des bases de données
2. Préparer les données: Analyser, visualiser, prétraiter, transformer, extraire
3. Concevoir le modèle (type de prédicteur, principe d'apprentissage)
4. Définir un critère et Optimiser (l'apprentissage

proprement dit)

5. Evaluer

EXTRACTION DE CARACTÉRISTIQUES

Travailler avec des données

Deux activités complémentaires:

Préparer les données

- Etape coûteuse mais indispensable
- Objectif: rendre possible l'apprentissage avec des données: •

Propres, homogènes, recalées, calibrées, organisées, facilement accessibles, renseignées…

- « Data engineering » (un nouveau métier!)

Transformer les données

- Objectif: Extraire l'information des données, leurs caractéristiques (« features »), construire leur forme

Extraction de caractéristiques

- « Feature extraction » en anglais
- Données brutes pas exploitables directement:

- Bruitées
- Grandes dimensions (image, enregistrement)
- Information utile noyée

- Etape critique de la « reconnaissance des formes » • Caractéristiques trop simples: pas assez d'information, confusion • Caractéristiques trop riches: complexité, bruit, grande variabilité Compromis difficile à régler entre expressivité, invariance, robustesse, taille, coût de calcul…

- Deux cas de figure:
  - On sait ce qui est important et pourquoi (expertise « métier »)

    modélisation
  - On ne sait pas décrire ce qui est important

    on l'apprend!

$$\{\exists_L,\ldots aaAy =\in_1$$

Chaîne de prédiction générique = Fy

$\mathbf{x})($

capteur

Prétraitement
extraction de
caractéristiques

Algorithme de
prédiction

mesures, images, texte…

^{N}
$\in RI$

caractéristiques

$^{d}\mathbf{x}\in R$

sources

$\{\ \}_{K}$
$= {}_{1},\ldots ssS$

valeur/décision/action

*C'est un **3***

$\{\nexists aaAy_{L},\ldots =\in_{1}$

1 – Construire une représentation

(forme) = $Fy\ \mathbf{x})($

capteur Prétraitement	extraction de caractéristiques	Algorithme de prédiction

mesures, images

N
$\in RI$

prédiction/décision/action

sources
$\{\}_K$
$= {}_1,\ldots ssS$

caractéristiques
$^d \mathbf{x} \in R$

Extraction d'information « métier »

capteur

2 - Prédire

Prétraitement
extraction de
caractéristiques

$\{\}aaAy_{L},\ldots =\in_{1}$

$= Fy\ \mathbf{x})($

Algorithme
de
prédiction

mesures, images

^{N}

$\in RI$

caractéristiques

$^{d}\ \mathbf{x}\in R$

sources

$\{\ \}_{K}$

$= {}_{1},\ldots ssS$

prédiction/décision/action

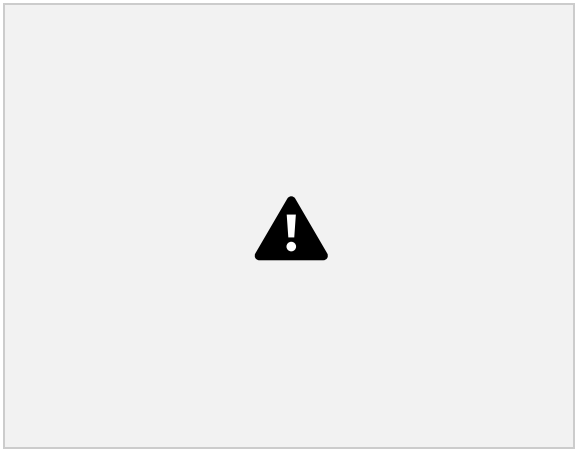
Etape « machine learning »

Exemples de caractéristiques en image

• Deux grandes classes: forme ou texture

• Forme

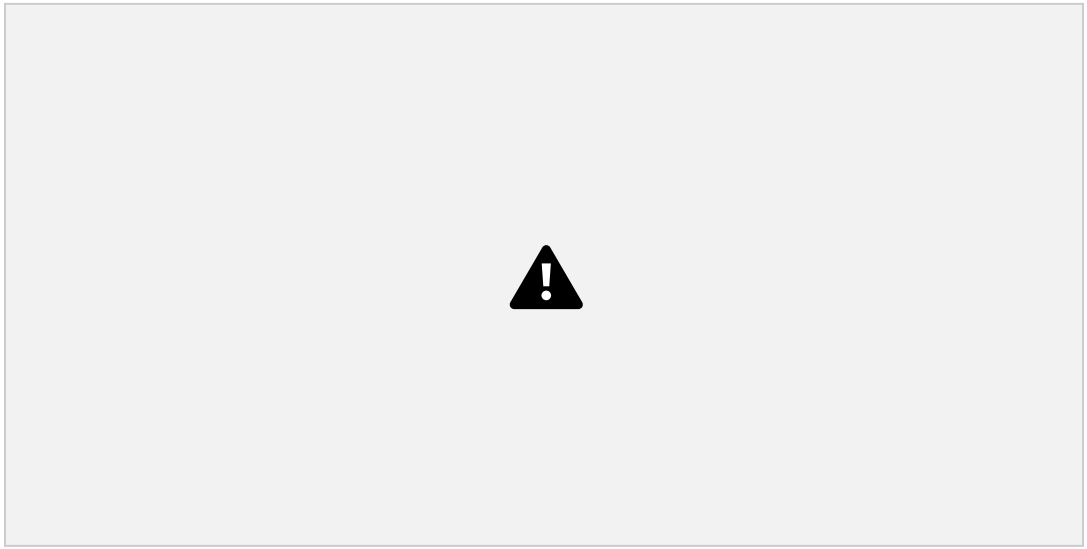
• Dépend d'une étape de séparation du fond (segmentation, saillance) • Caractéristiques structurelles ou



• Texture

• Globales et/ou locales

• Plus difficiles à associer à un objet précis



$\{\exists aaAy\ _{L},\ldots =\in\ _{1}$

Prédire & extraire en même temps =

$Fy\ \mathbf{x})($

capteur
Prétraitement

extraction de
caractéristiques

Algorithme de
prédiction

mesures, images

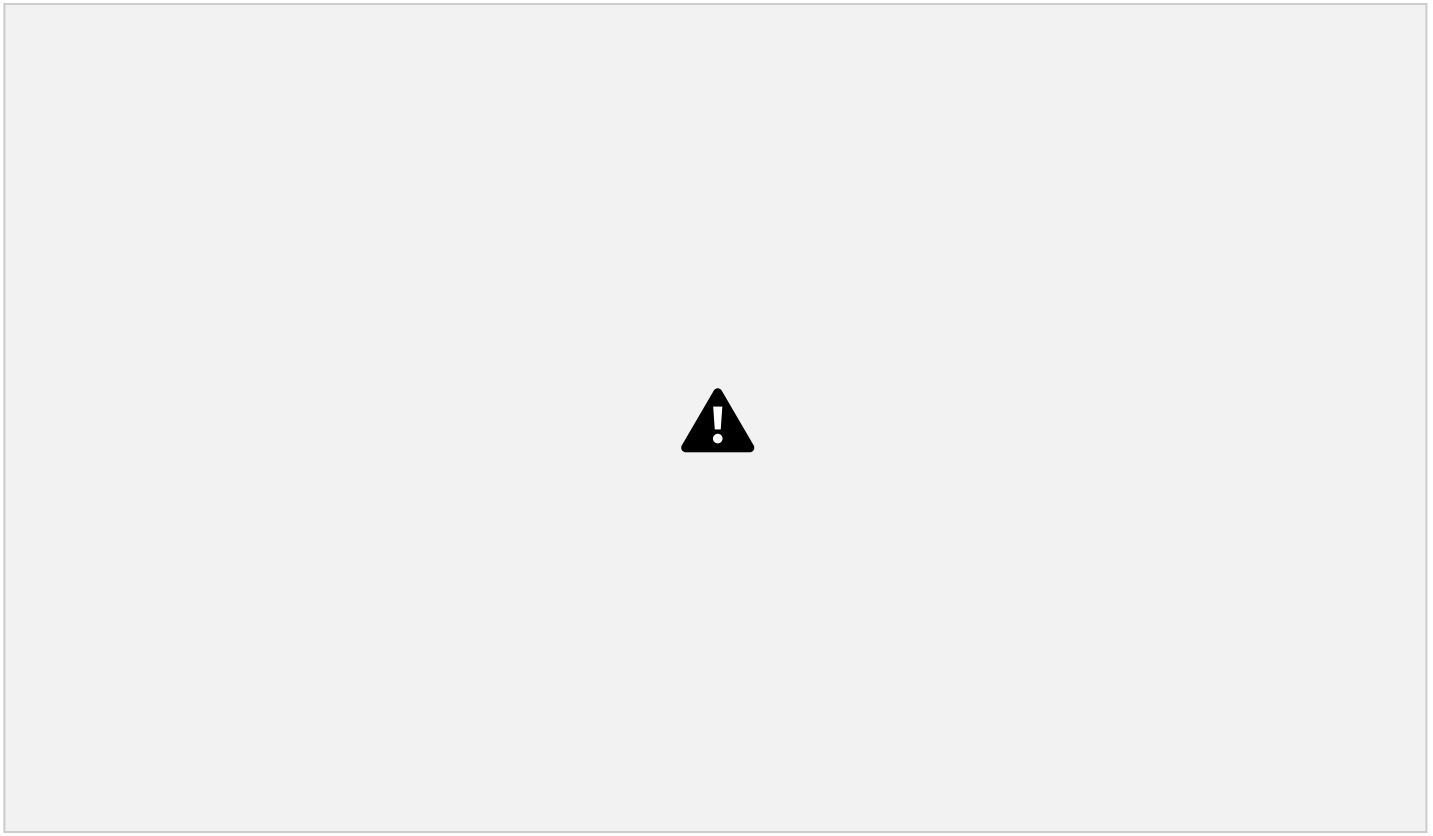
^{N}
$\in RI$

sources
$\{\ \}_K$
$= {}_1,\ldots ssS$
prédiction/décision/action

caractéristiques
$^{d}\ \mathbf{x} \in R$

Etape « machine learning » moderne

« Deep features »



On peut apprendre les caractéristiques image
Réseaux convolutifs (cours DL)

UN CONCEPT CENTRAL: LA GÉNÉRALISATION

Exemple : régression polynomiale

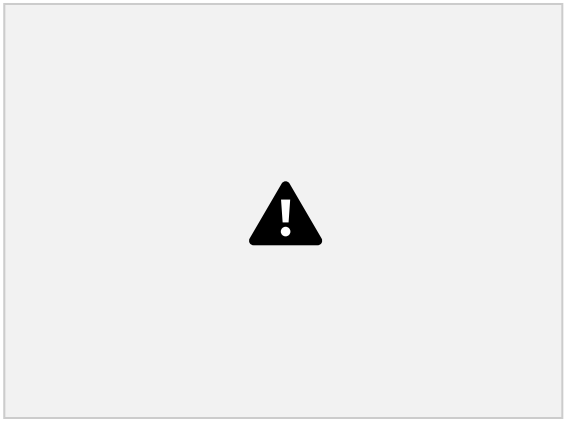
carré entre les points vrais et le polynôme estimé.

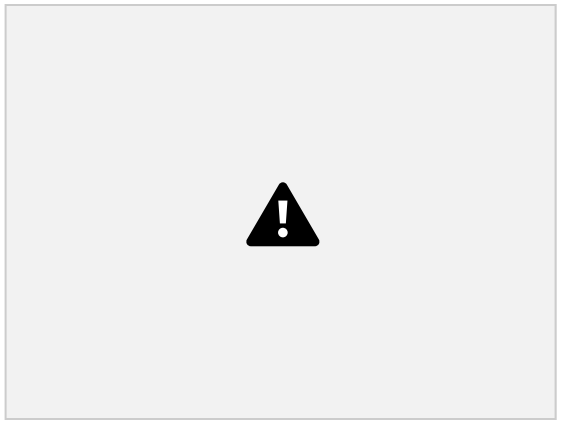
- La courbe verte est la véritable fonction à estimer (non polynomiale)

- Les données sont uniformément échantillonnées en ⬚⬚ mais bruitées en ⬚⬚.

- L'erreur de régression est mesurée par la distance au

from Bishop



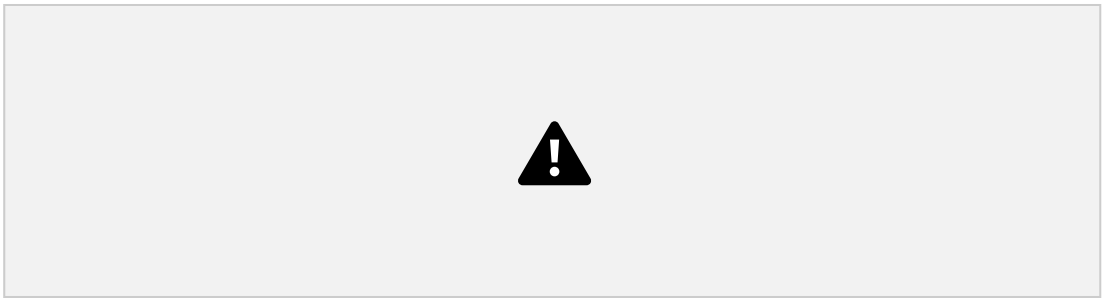


Modèles linéaires généralisés (⬚⬚$\epsilon\mathbb{R}$)

$$_{110\ 22}\, xwxwxwwxFy\ \mathbf{w,\ \varphi\varphi\ ++++==\ \varphi}\ _{MM})(\ )()(\ \ldots$$

$$)($$

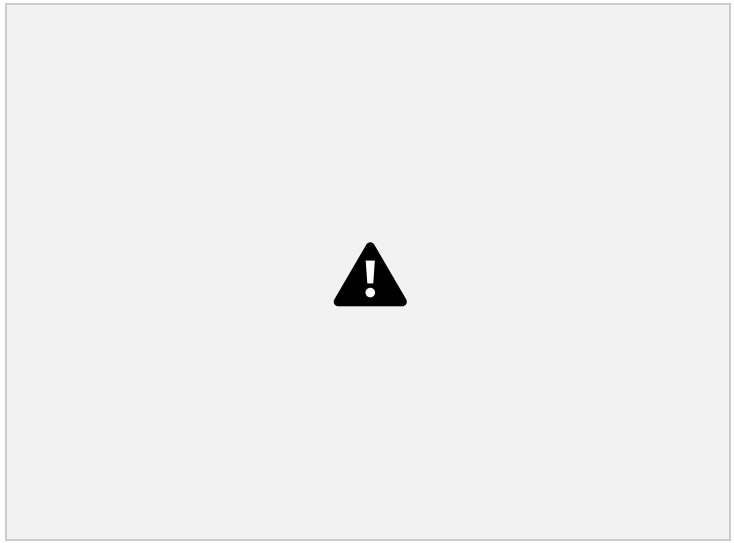
- Prédiction utilise des **fonctions de base** encodant les données source (« features »): ⬚⬚ ⬚⬚ = ⬚⬚$^{⬚⬚}$
- **Apprentissage** = Maximum de Vraisemblance:

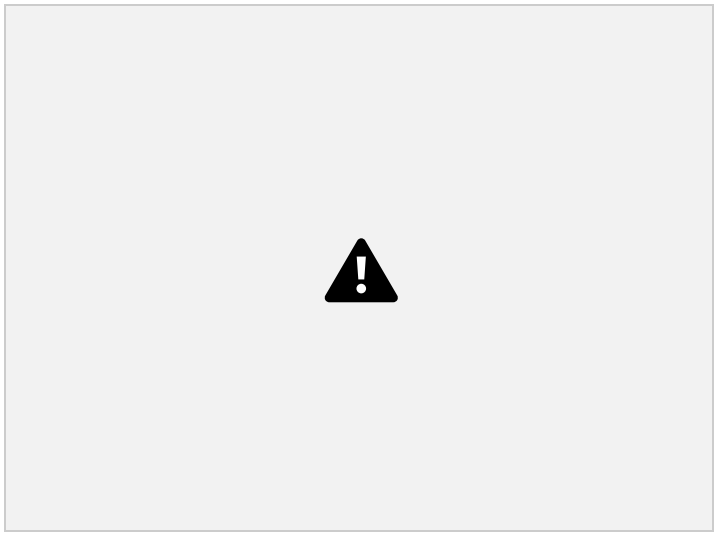


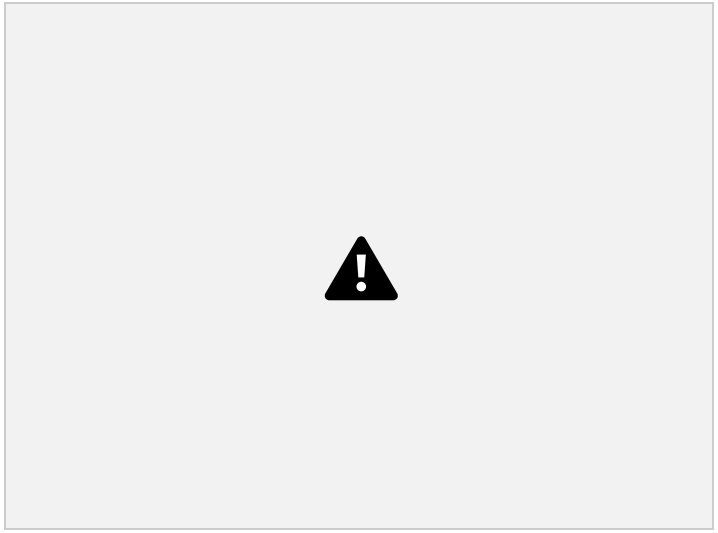
- Que vaut cet apprentissage?

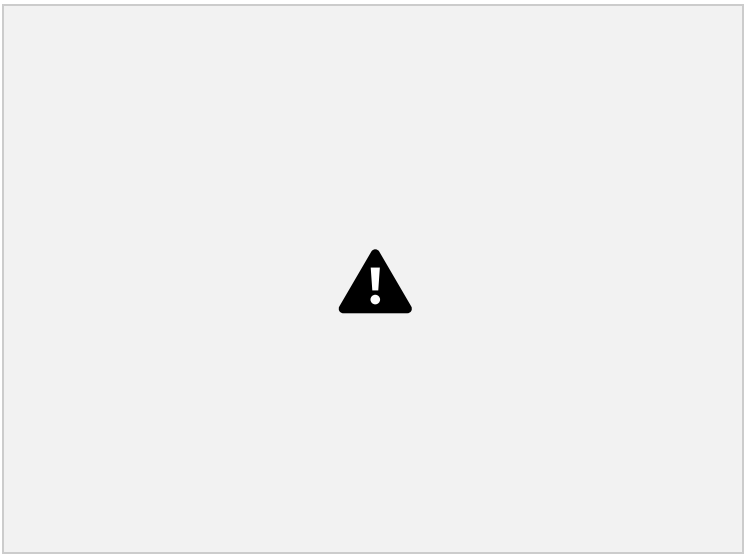
Régression polynomiale: ⬚⬚ ⬚⬚ = ⬚⬚$^{⬚⬚}$

from Bishop









“Training vs. Test”

Erreur de régression est calculée sur des données $\{ \mathbf{x}_i, y_i \}_{i=1}^{N}$:

$$E_{RMS} = \sum_i \left(y_i - \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) \right)^2$$

Mais quelles données?

- Données d'apprentissage (Training):
  - c'est un moyen de modélisation

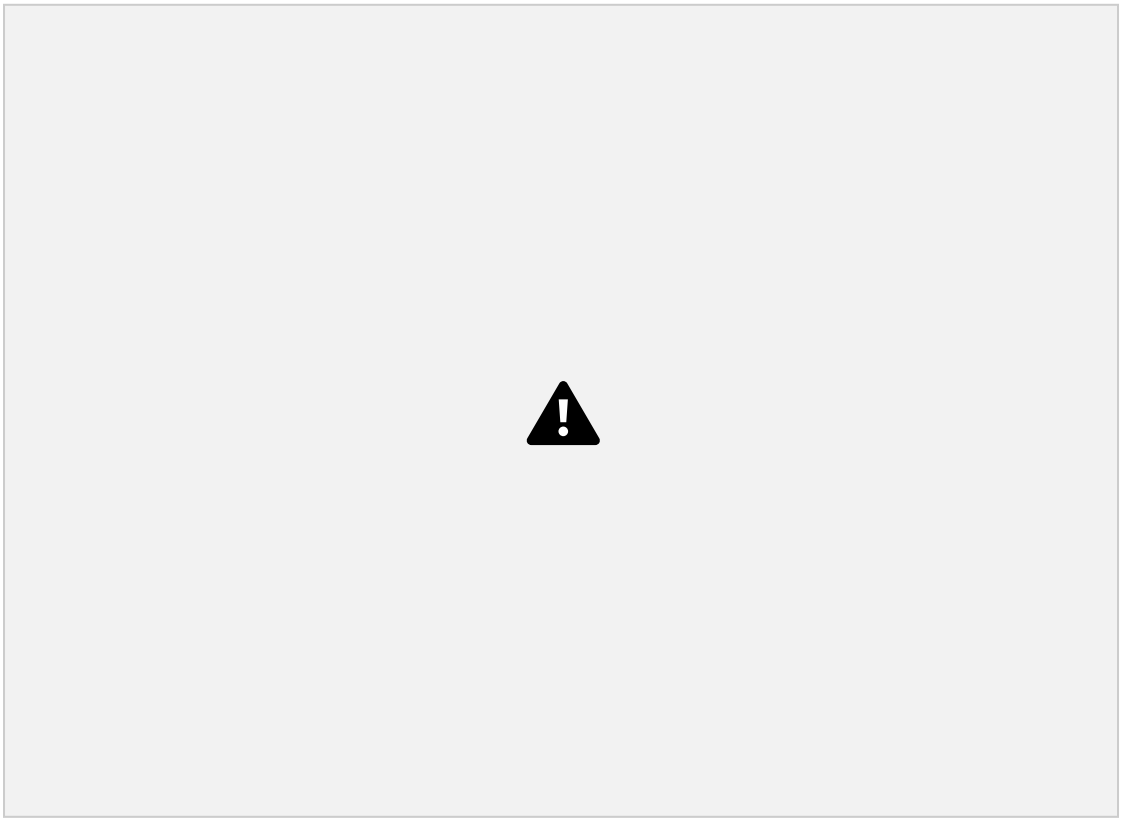
- Données opérationnelles (Test):
  - c’est la situation réelle
  - Celles pour lesquelles on veut une bonne prédiction

Comportement des erreurs

$$2 \quad N \quad {}^{T}yx \quad {}_{=}E\ \mathbf{ww}\ \varphi \quad {}_{\mathrm{RMS}}))(.()(\quad = \sum -$$

$_1 ii$

i



Prédicteur pas assez précis

Bon régime

Sur apprentissage = « Overfitting »

Erreur de généralisation

• La mesure de bon fonctionnement est l'erreur sur des données nouvelles

généraliser ≠ mémoriser (par cœur)

• Problème: les données nouvelles sont par nature inconnues! (sinon, elles seraient utilisées)

 Il est nécessaire de faire des hypothèses sur leur nature et sur le modèle de prédiction.

Deux phénomènes à contrôler (éviter)

• **Simplisme:** modélisation trop grossière pour rendre compte de la variété des données

  • Erreur d'apprentissage et de test importantes

• **Sur-apprentissage (« Overfitting »):** modèle trop complexe se spécialisant sur les données d'apprentissage

- Ecart entre erreur d'apprentissage et erreur de test

Construire son chantier d'apprentissage

- Préparer les données
  - Simplifier/compléter/formater/homogénéiser/calibrer…
- Diviser en deux ensembles:
  - **Apprentissage** ("Train") pour optimiser les paramètres du modèle.
  - **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.

DEUX APPROCHES ÉLÉMENTAIRES

Modélisation bayésienne

Plus proches voisins

Théorie Bayésienne de la décision

- On considère les données ⬚⬚, ⬚⬚ comme des variables aléatoires.
- On les modélise par des lois de probabilités:
  - ⬚⬚ ⬚⬚ , ⬚⬚ ⬚⬚ : lois a priori (ou marginales)
  - ⬚⬚ ⬚⬚, ⬚⬚ : loi jointe
  - ⬚⬚ ⬚⬚ | ⬚⬚ : vraisemblance conditionnelle
  - ⬚⬚ ⬚⬚ | ⬚⬚ : loi a posteriori
  - Classification: ⬚⬚ ∈ 1,2 ... ⬚⬚ est une étiquette • On

cherche à prédire une unique étiquette ��* à partir de ��

$$\text{}\square\square \mapsto \square\square^*$$

- Théorie de la décision démontre que le meilleur choix est:

$$\square\square^* = \arg\max$$

$$\square\square\square\square \square\square \mid \square\square$$

Théorie Bayésienne de la décision

- Deux questions:
  - Comment calculer ☐☐ ☐☐ | ☐☐ = apprentissage
  - Comment trouver le max = prédiction
- « Astuce »: utiliser la loi de Bayes

$$P(C \mid x_1, \ldots, x_n) = \frac{P(x_1, \ldots, x_n \mid C)\, P(C)}{P(x_1, \ldots, x_n)}$$

- On connait en général la fréquence d'occurrence des classes C
- On sait plus facilement calculer la **vraisemblance**: $P(x_1, \ldots, x_n \mid C)$
  - « Si je sais dans quelle classe je suis, je sais décrire le comportement/distribution de mes données »
- Le max sur C ne dépend que de $P(x_1, \ldots, x_n \mid C)$

et $P(C)$

$$C^* = \arg\max_C P(x_1, \ldots, x_n \mid C)\, P(C)$$

Approche Bayésienne multivariée

- Calcul de la loi conditionnelle: Modèle multivarié
- Par ex. modèle gaussien décrivant $\mathbf{x} = \mathbf{x}_1, \mathbf{x}_2 \ldots$

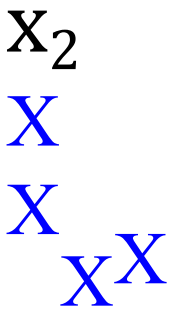
$$\mathbf{x}_i \in \mathbb{R}^{\mathbf{x}}: \mathrm{P}\ \mathbf{x}; \mathbf{x}, \mathbf{x} = 1$$

$$(2\mathbf{x})^{\mathbf{x}/2} \exp -\frac{1}{2}(\mathbf{x} - \mathbf{x})'\mathbf{x}^{-1}(\mathbf{x} - \mathbf{x})$$

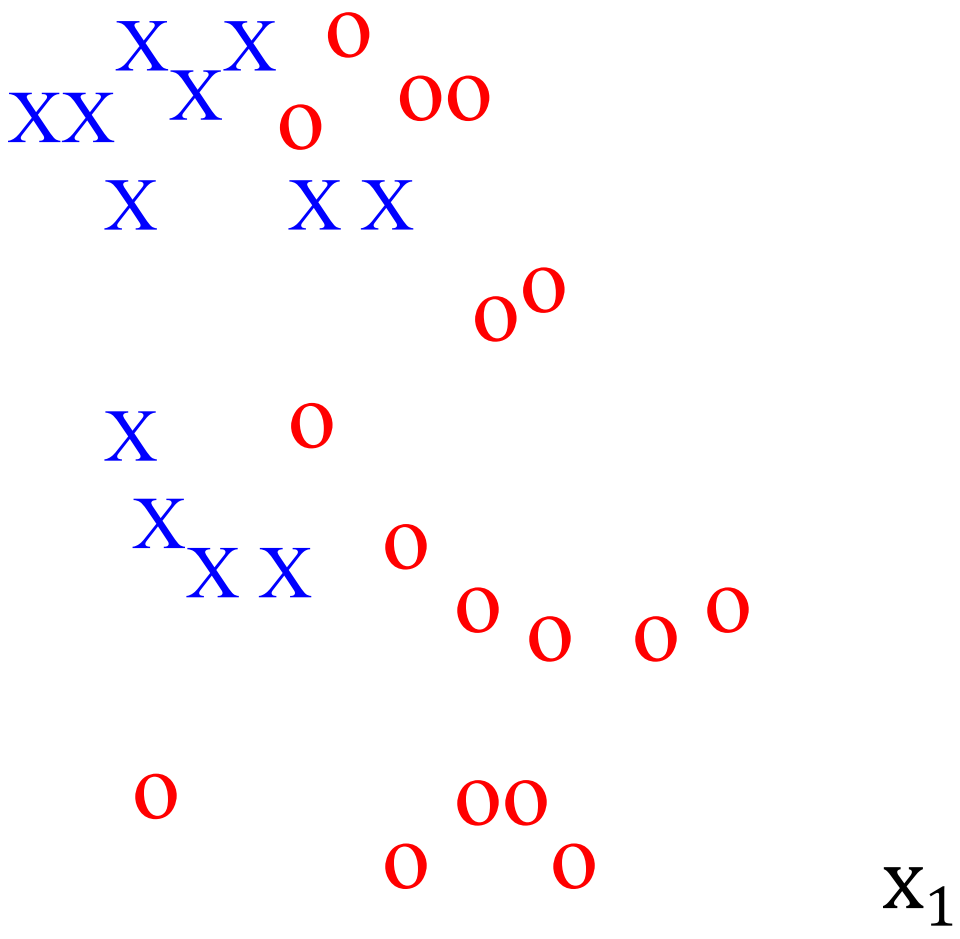
- Permet de décrire les corrélations entre dimensions.
- Mais demande de connaître la forme des distributions + limitation à petites dimensions.
- Si modélisation gaussienne et deux classes, la prédiction se réduit à calculer une fonction de degré 2

Approche gaussienne multivariée

P ◆◆; ◆◆■, ◆◆■



P ◆◆; ◆◆■, ◆◆■



Séparatrice = Forme quadratique

$$(\text{□□} - \text{□□}_1)'\Sigma_1{}^{-1}(\text{□□} - \text{□□}_1) \text{ - } \text{□□} - \text{□□}\|'\Sigma_2{}^{-1}\,\text{□□} - \text{□□}\| \geqslant$$
$$\text{□□□□□□□□}$$

Approche Bayésienne Naïve

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

$$\text{□□ □□}\|, \text{□□}\|\ldots \text{□□}\|\text{□□} = \text{□□ □□}\|\text{□□}\|\ldots \text{□□}\|, \text{□□ □□ □□}\|\ldots \text{□□}\|\text{□□}$$
$$= \text{□□ □□}\|\text{□□ □□ □□}\|\ldots \text{□□}\|\text{□□}$$
$$= \text{□□ □□}\|\text{□□ □□ □□}\|\text{□□ ... □□ □□}\|\text{□□}$$

• On calcule la vraisemblance globale dimension par dimension Problème 1D, modèles plus faciles à estimer (gaussien, binomial, histogrammes, mélange de gaussiennes…)

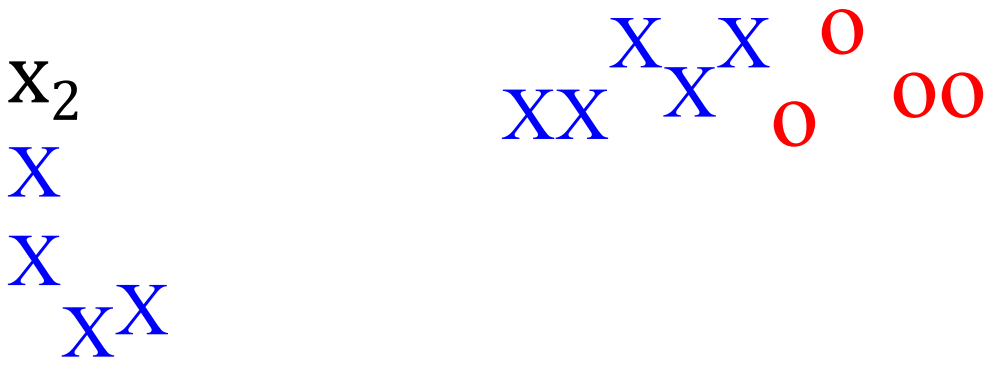
 Permet de traiter des problèmes de plus grande dimension

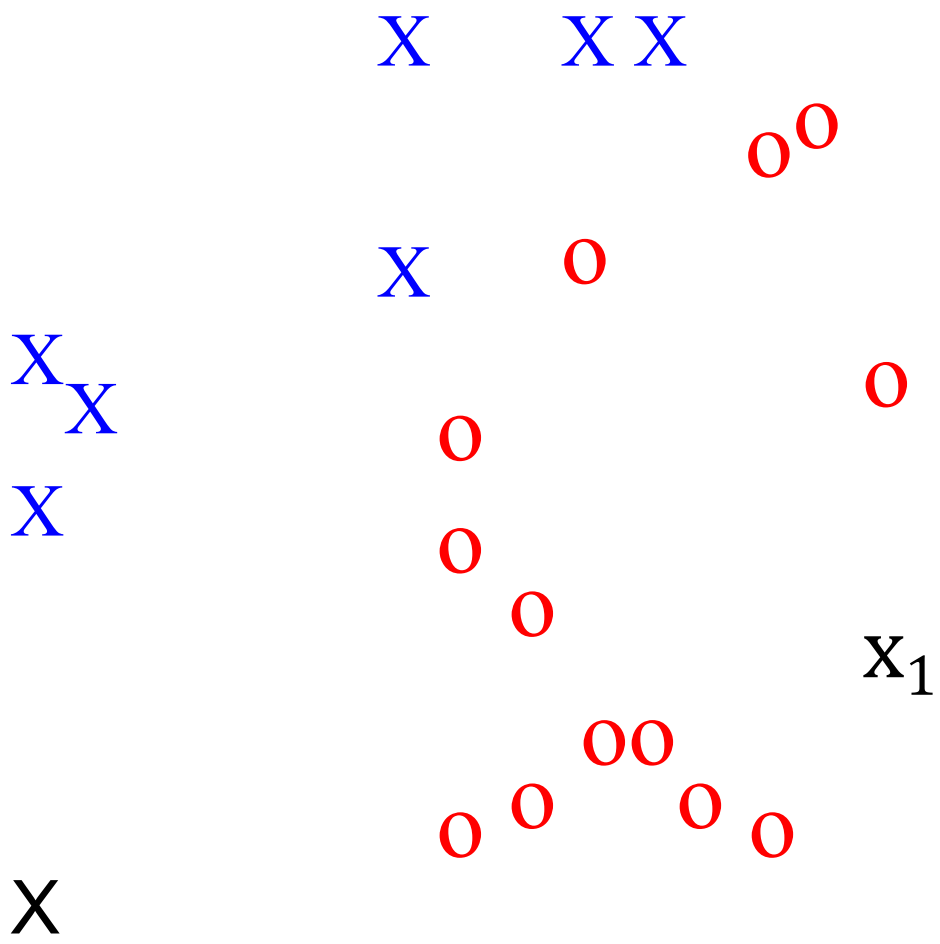
- En pratique, on calcule plutôt la log-vraisemblance pour des questions de stabilité numérique

$$\log p(x \mid y) = \log \prod_{i=1}^{d} p(x_i \mid y)$$

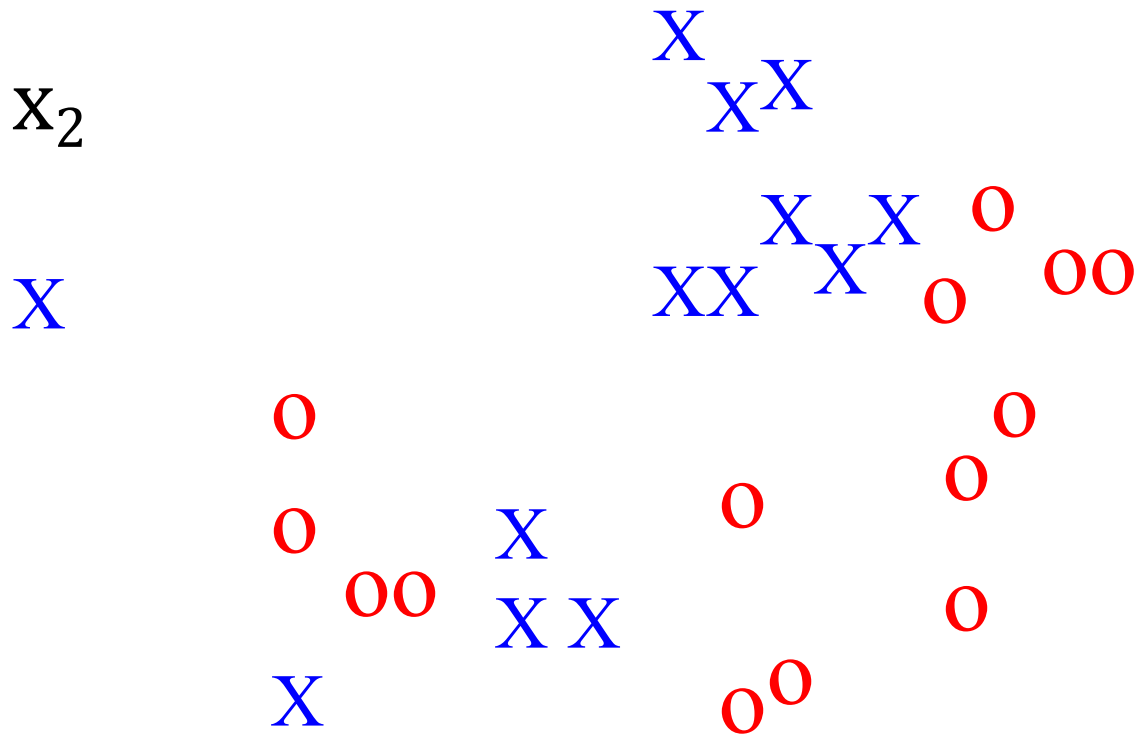
$$y^* = \arg\max_y \log p(y) p(x \mid y) + \log p(y)$$

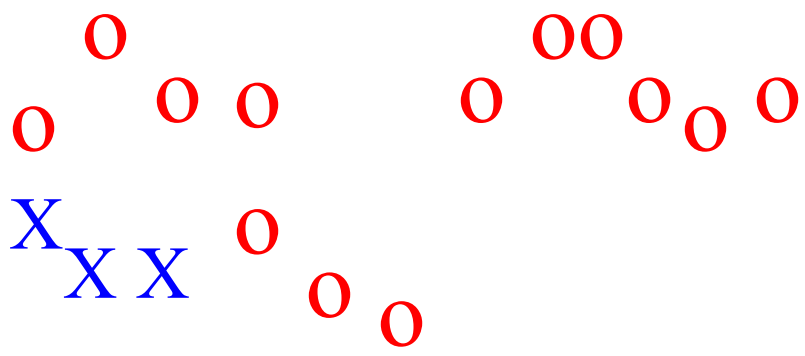
Approche bayésienne naïve

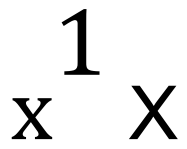



x
x x
o o
x
o
x
x
o
o
x
o
o
x1
o o
o
o
o
o
x

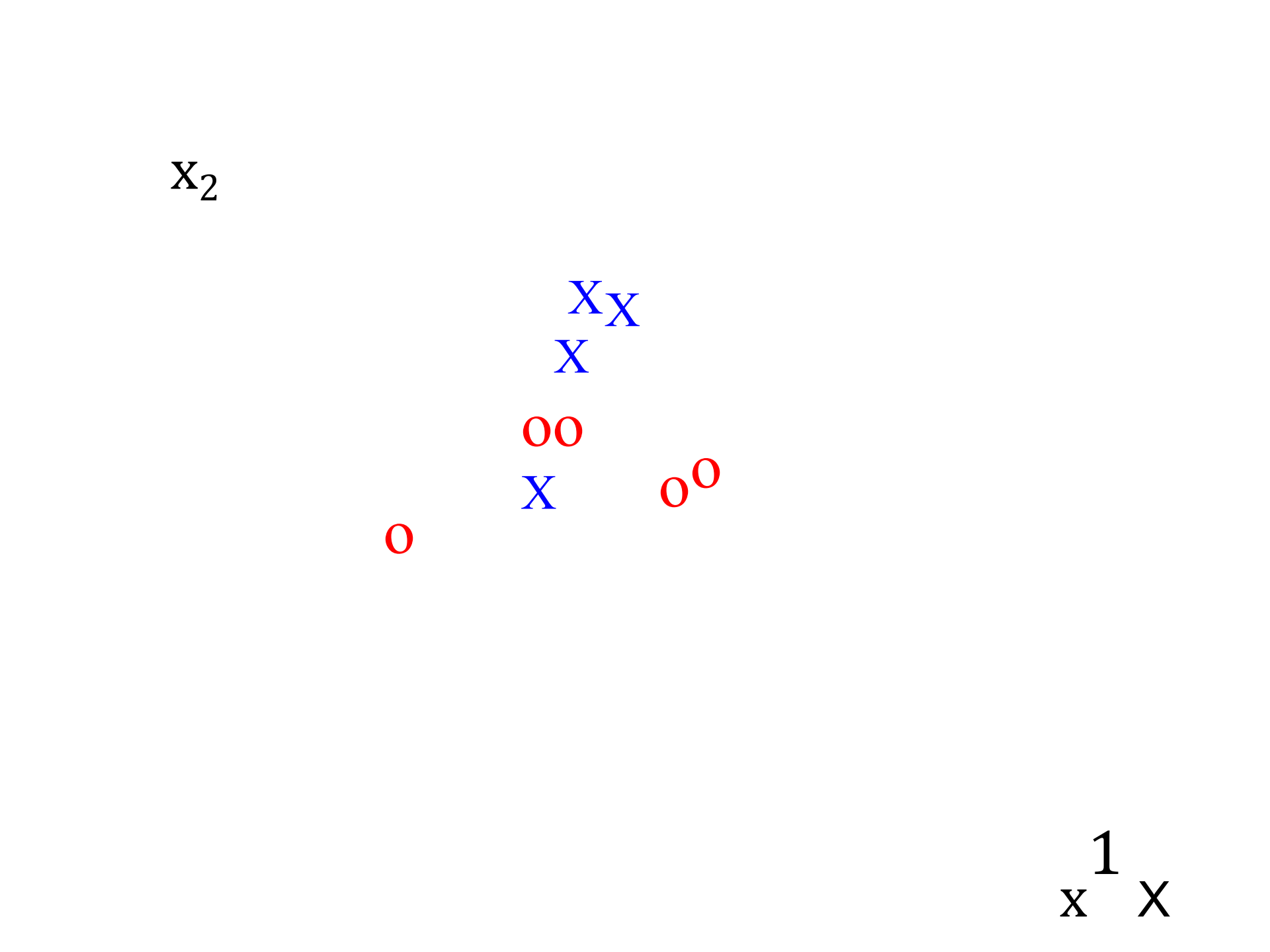
Approche bayésienne naïve vs. multivariée







Approche bayésienne naïve vs. multivariée


x_2
x_1

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

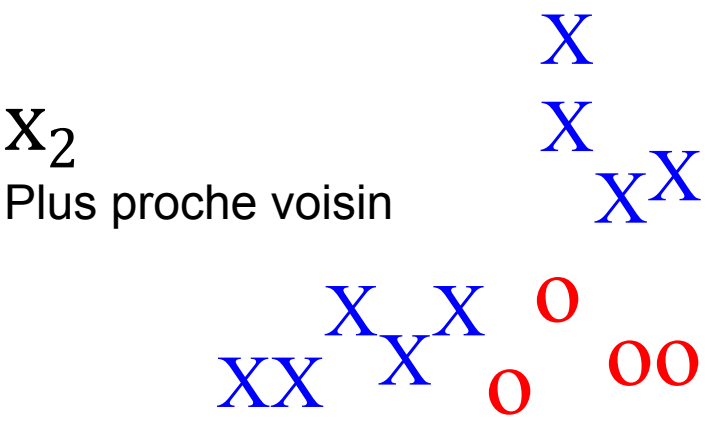
- Apprentissage
  - Estimation de lois paramétriques simples
- Prédiction
  - Calcul de log-vraisemblance et max sur hypothèses

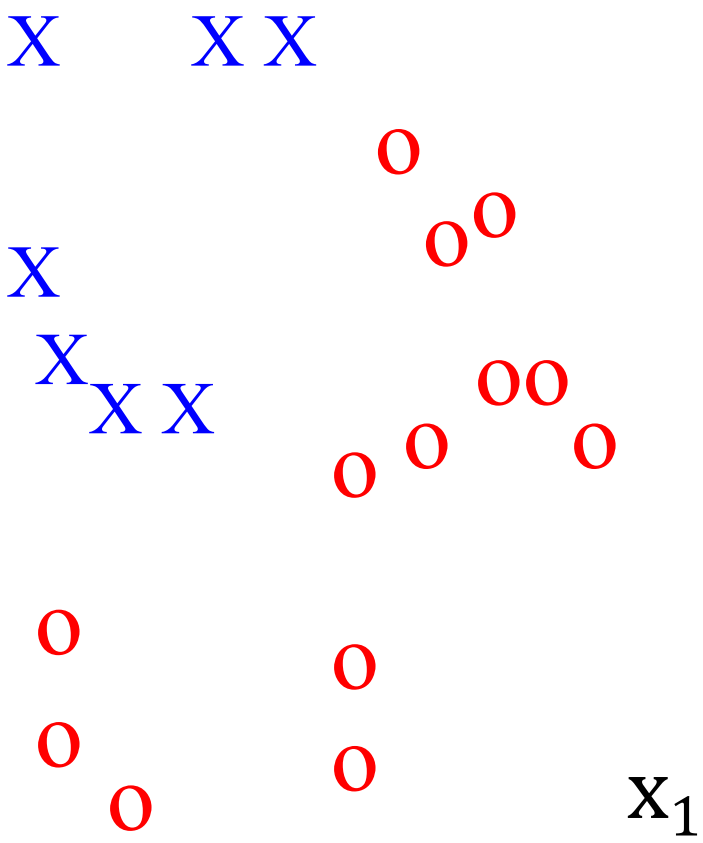
- Quand l'utiliser? (limitations)
  - Petits problèmes bien modélisés (gaussien multivarié) • Caractéristiques non corrélées (bayésien naïf, mais ça peut aussi marcher si c'est corrélé)

Classification ppv

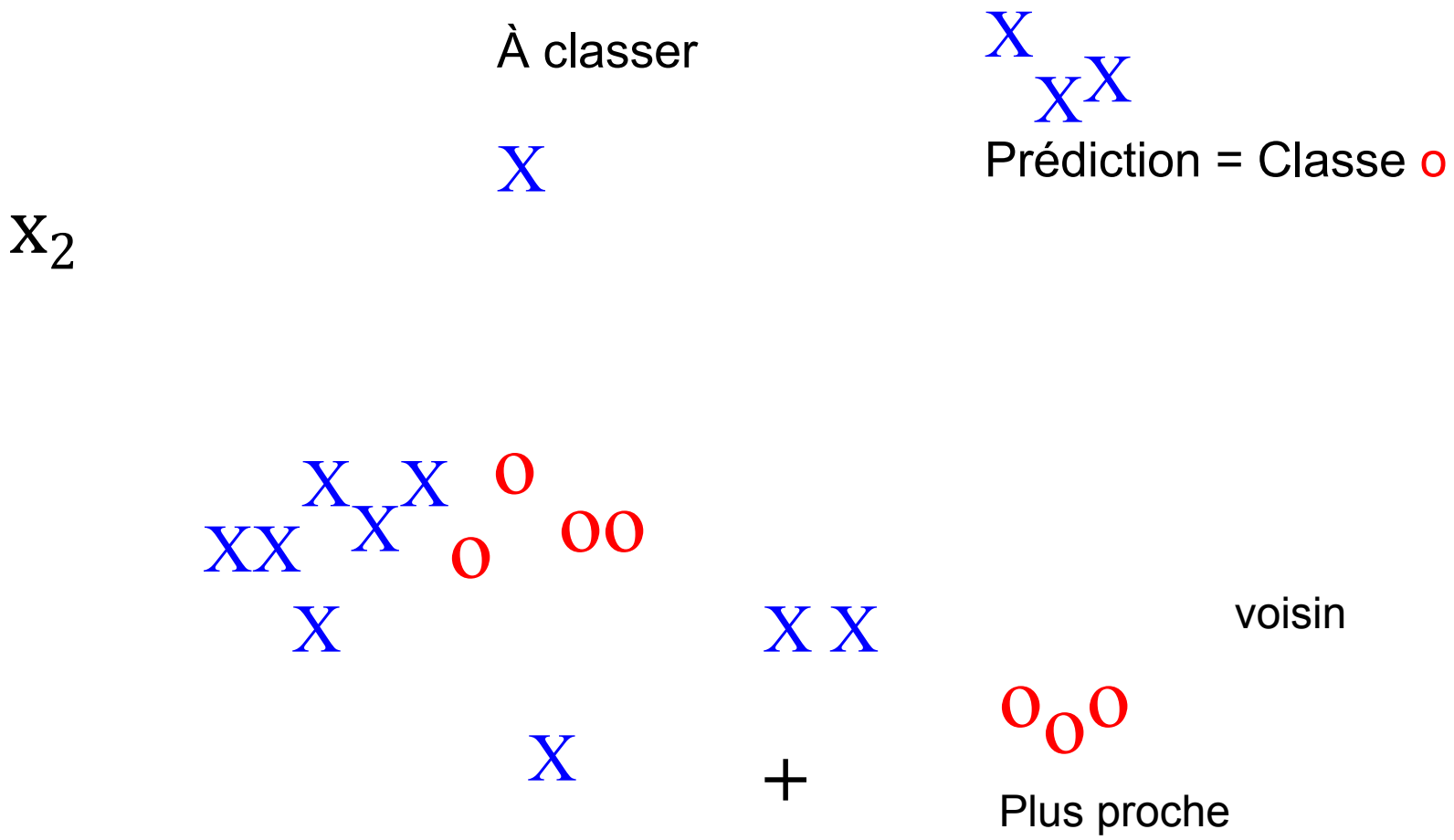
Prédiction = Classe **X**
À classer

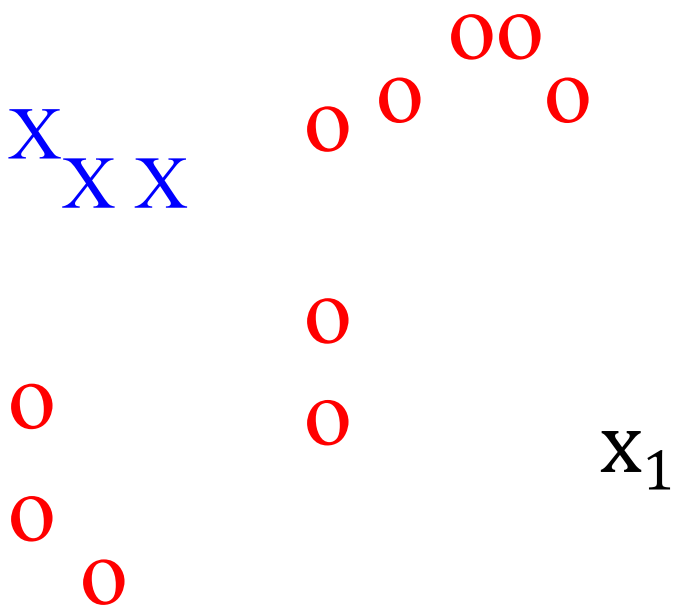
+





Classification ppv


À classer
Prédiction = Classe o
x_2
voisin
Plus proche



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 $d(x, x')$
  - 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 $\mathcal{L} = x_1, x_2, x_3, x_4, \ldots x_n, x_{n+1}$
  - Recherche de l'échantillon le plus proche: $x^* = \arg\min_{x_i} d(x, x_i)$
  - Assigne comme prédiction l'annotation du plus proche: $y^* = y_{x_i^*}$

Régression PPV

Valeur estimée

$= R\ \mathbf{x})($

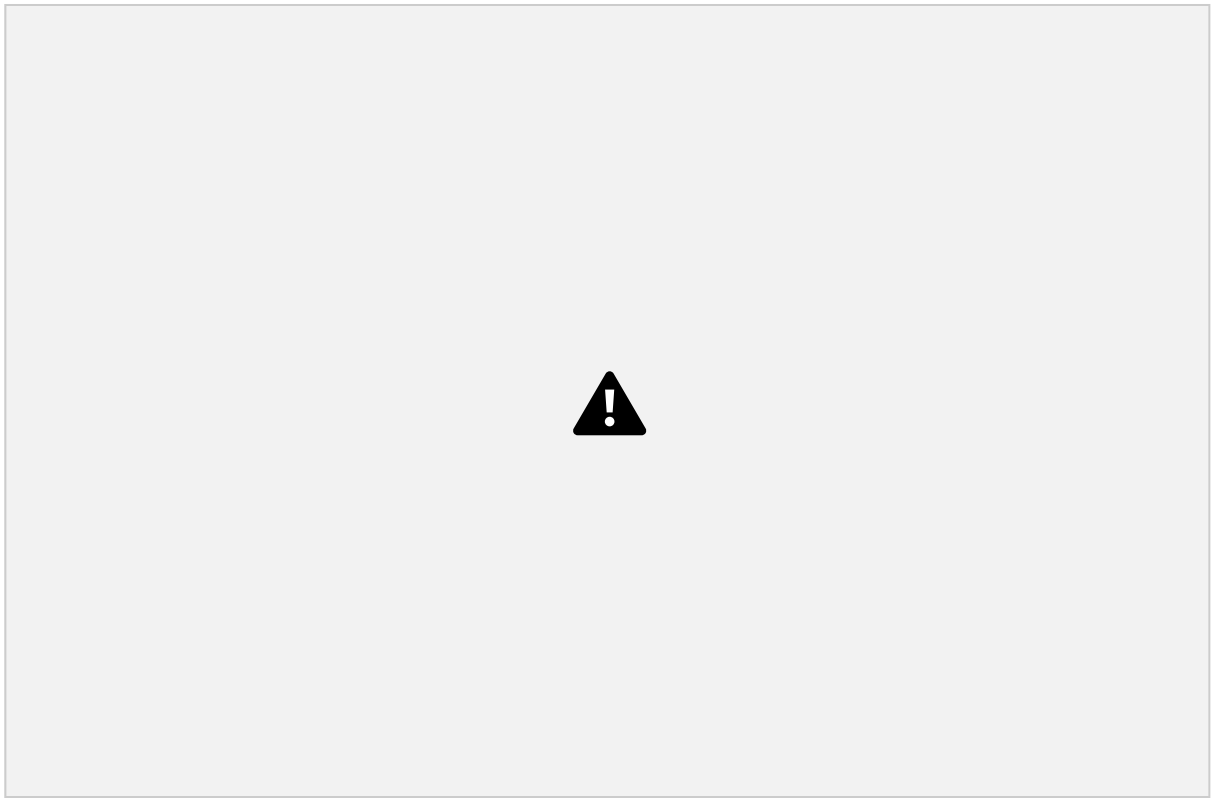
y

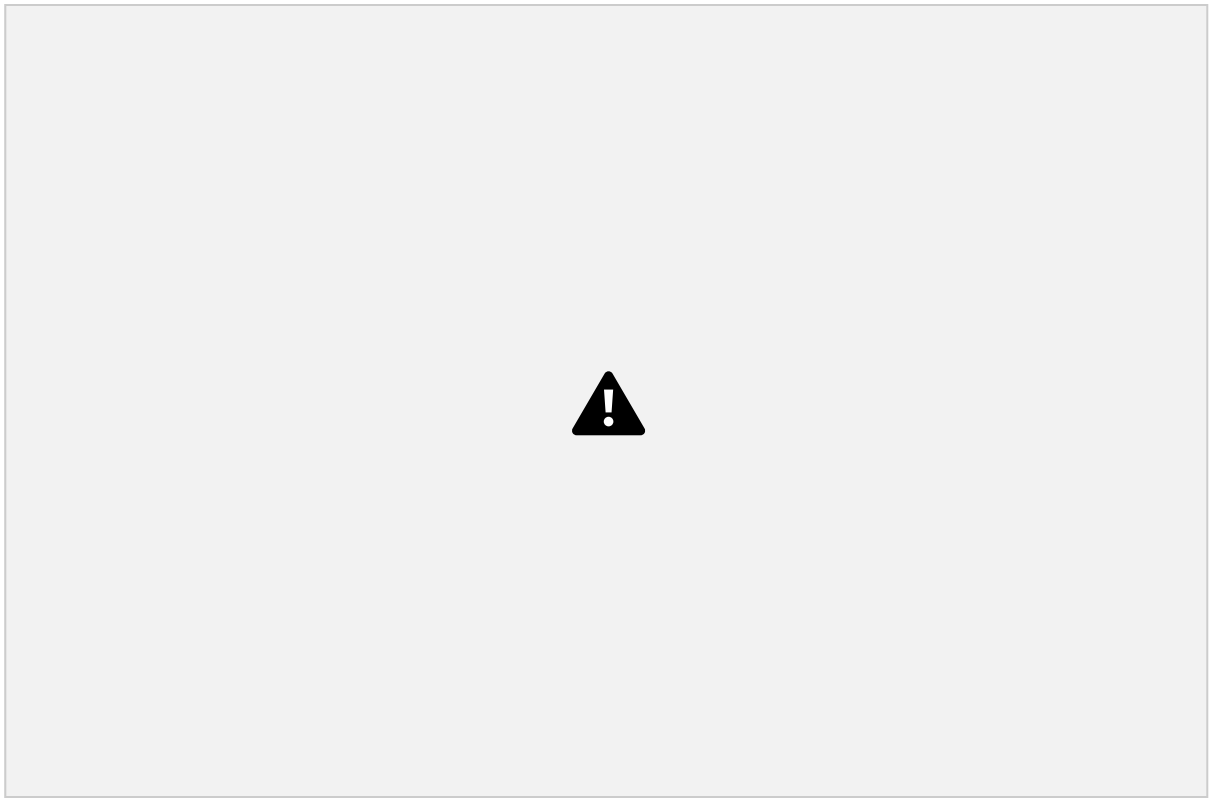
$\mathbf{x}$

À estimer Plus proche voisin

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:

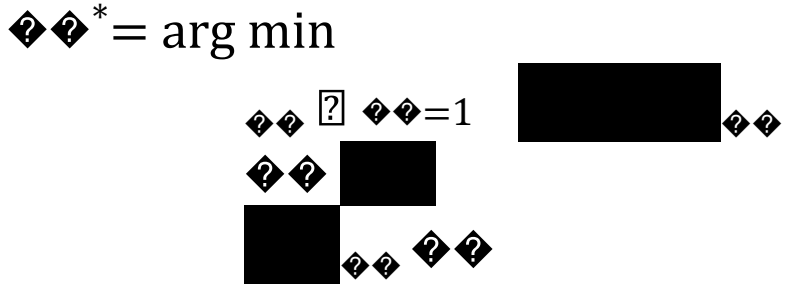
$$\text{[redacted]}_1 \leq \text{[redacted]}_2 \leq \cdots \leq \text{[redacted]}_{\text{[redacted]}}$$

- On choisit les [redacted] plus proches
- On prédit en choisissant la classe recueillant le plus de votes

$$\text{[redacted]}^* = \arg\min_{\text{[redacted]}} \sum_{\text{[redacted]}=1}^{\text{[redacted]}} \text{[redacted]}$$

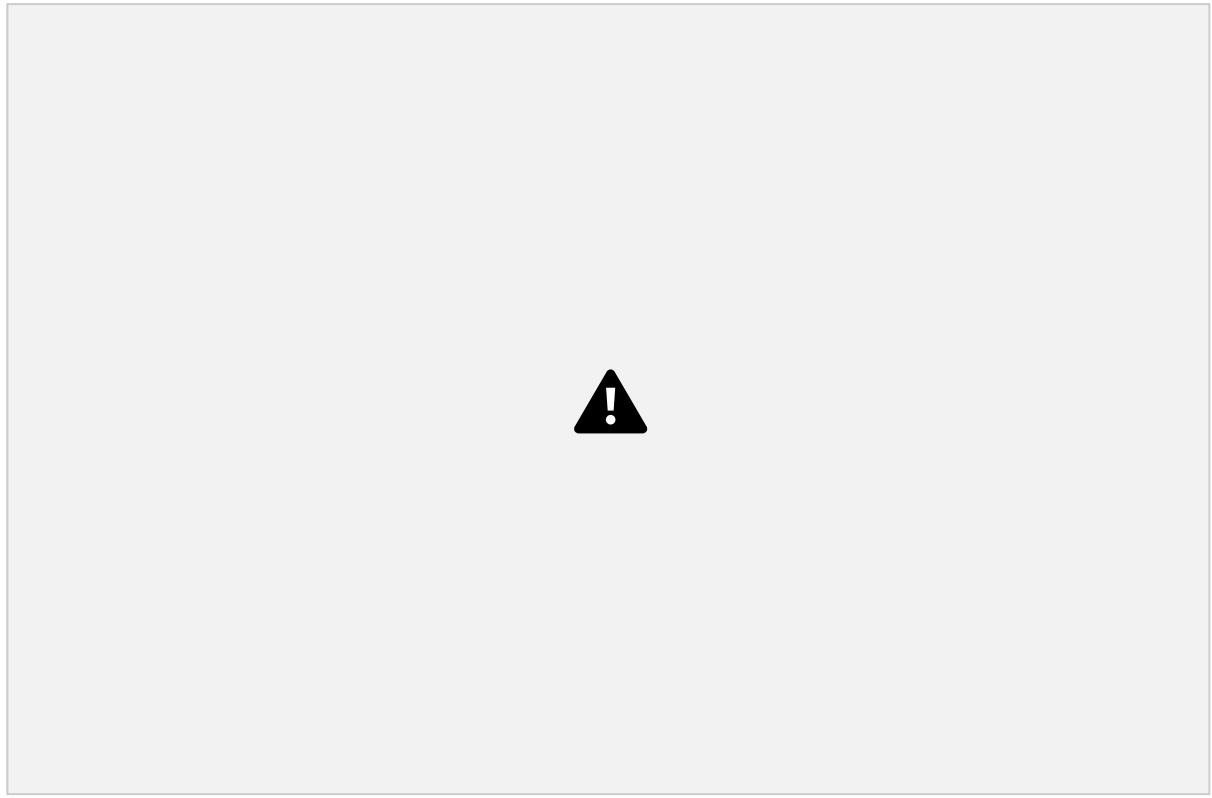
Où [redacted] est la fonction de Kronecker (elle vaut 1 si égal, 0 sinon) • Si pas de max (ambiguïté sur la prédiction) on ne décide pas! • On peut aussi pondérer les votes:

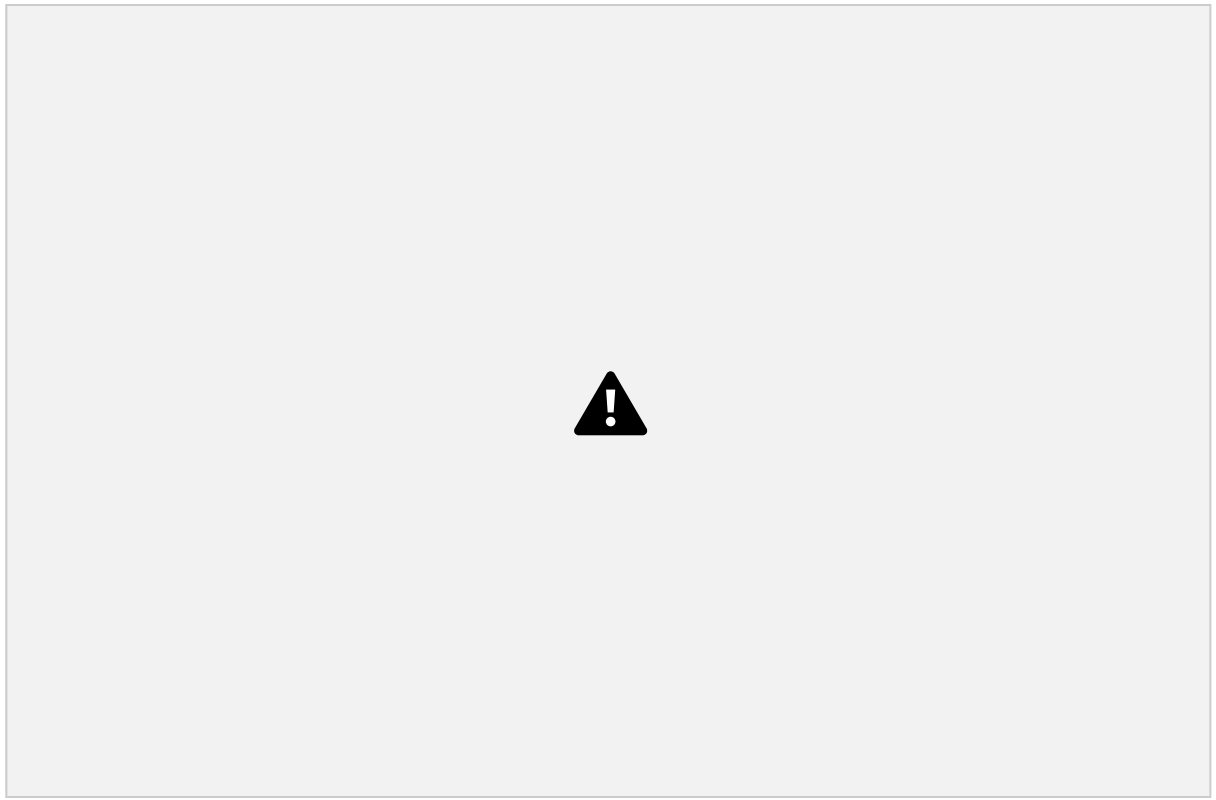
[redacted]

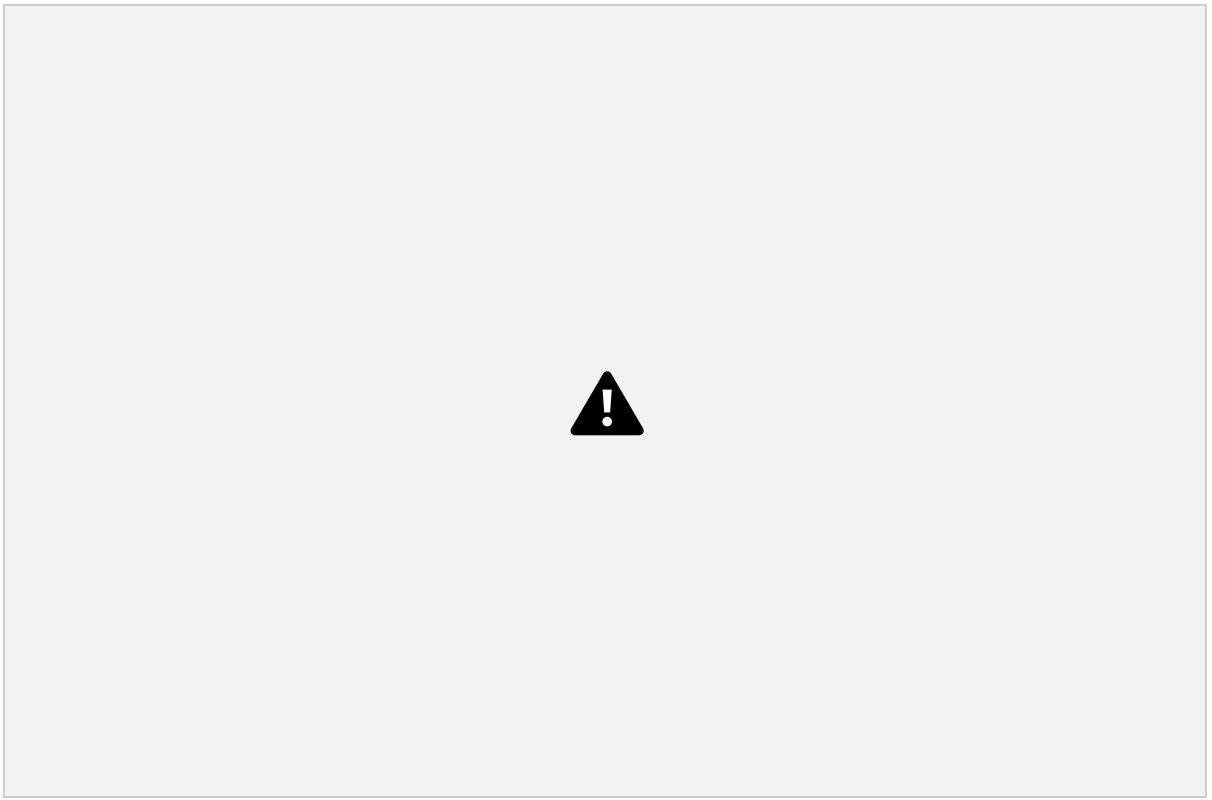


Fonction de classification 5 ppv Données

bruitées Régions isolées mauvaise régularité des prédictions







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

Propriétés statistiques

Bornes statistiques asymptotiques (�� → ∞)

$$\text{??} \leq \text{??}_{\text{??????}} \leq \text{??}\ 2 - \frac{\text{????}}{\text{??} - 1}$$

Où ?? est l'erreur théorique optimale (Bayes), ?? est le nombre de classes et $\text{??}_{\text{??????}}$ est l'erreur des k-ppv.

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

Coût de la prédiction du k-ppv

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

$$⬛_1 \leq ⬛_2 \leq \cdots \leq ⬛_{⬛}$$

- Pour N et d grands, coût important de la recherche exhaustive ⬛ ⬛. 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 $\square\square$ de l'espace de représentation augmente, les points sont tous aussi proches ou aussi loin.
- On peut montrer, pour une distribution quelconque de $\square\square$ points tirés de manière indépendante dans $[0,1]^{\square\square}$, que:

$$\lim_{d \to \infty} \frac{dist_{max}(d) - dist_{min}(d)}{dist_{min}(d)} = 0$$

- Ce n'est plus vrai si les distributions sont structurées…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 → apprentissage non supervisé

Comportement des PPV

- Avantages
  - Schéma flexible, facile à mettre en œuvre, dépendant de la définition d'une similarité entre données.
  - Bonnes propriétés statistiques (�� → ∞)

- 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

« 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

A retenir

- « Programmer à partir des données »
  - Deux phases: apprentissage et prédiction
  - Plusieurs variétés de prédicteurs et d'apprentissage
- Démarche générique:
  - Constitution d'une base d'apprentissage
  - Analyse préliminaire des données + préparation
  - Conception du modèle

- Optimisation
- Evaluation

- Objectif principal: minimiser l'erreur de généralisation •

  Train vs. Test

- Deux approches élémentaires:
  - Modélisation bayésienne
  - Plus proches voisins

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, (https://www.microsoft.com/en us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf). • Jerome H. Friedman, Robert Tibshirani et Trevor Hastie, The Elements of Statistical Learning: Data Mining, Inference, and Prediction (https://web.stanford.edu/~hastie/ElemStatLearn/). • 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
  - Kitti: http://www.cvlibs.net/datasets/kitti/
  - Waymo: https://waymo.com/open

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)
• International Conference on Learning Representations (ICLR) • Uncertainty in Artificial Intelligence (UAI)
• International Joint Conference on Artificial Intelligence (IJCAI) • International Conference on Neural Networks (ICNN) • 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) •
…

Cours & tutoriaux

- Des MOOC (Français et Anglais)
- Des tutoriaux associés aux conférences (orientés recherche)
- Des cours en français:
  - https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle

- https://www.college-de-france.fr/site/stephane-mallat/_course.htm

- Des « cheat sheets »
  - https://stanford.edu/~shervine/teaching/

Logiciels

- Environnement génériques: Matlab, ScikitLearn

• Environnements Deep Learning: Tensor Flow, Pytorch, mxnet… • Beaucoup de codes sur GitHub

Apprentissage