Semaine IMT Grand-Est

Introduction à l'apprentissage automatique, la science de l'intelligence artificielle

Séance 2

Limites fondamentales de l'apprentissage, problèmes de partitionnement

Frédéric Sur

`https://members.loria.fr/FSur/enseignement/IMT_GE/`

Plan

Problème 1 : la malédiction de la dimensionnalité

ou *fléau de la dimension*
a.k.a. *curse of dimensionality*

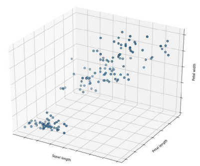
Expression inventée par Richard Bellman (années 1950)

→ plusieurs aspects liés, souvent en contradiction avec l'intuition que l'on développe en dimension 2 ou 3.

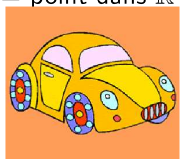
Problème : si les observations dépendent d'un grand nombre de variables, comment tirer parti des relations entre les variables pour prédire ou partitionner ?

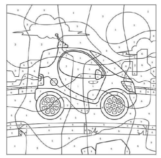
Exemples...

« facile » : dans $\mathbb{R}^3$

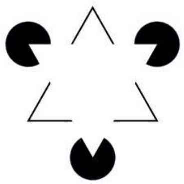


plus difficile : image numérique = point dans $\mathbb{R}^{2\cdot 10^7}$

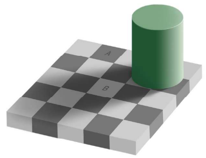




encore plus difficile :



Triangle de Kanizsa

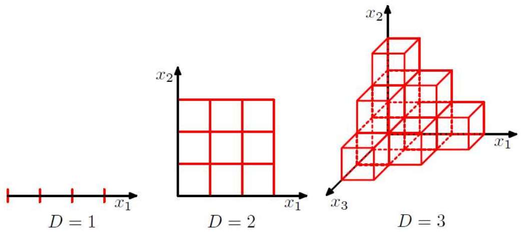


Échiquier d'Adelson

Dalmatian dog

La malédiction de la dimensionnalité : exemple 1

Nombres d'éléments pour discrétiser le cube unité



Source : C. Bishop, *Pattern Recognition and Machine Learning*, 2006.

En dimension D, le nombre d'éléments de côté $1/n$ nécessaire pour discrétiser un cube de côté 1 est : n^D

$\rightarrow$ conséquence pour l'estimation d'une densité de probabilité ?

La malédiction de la dimensionnalité : exemple 2

Explosion combinatoire

$\rightarrow$ quelle est la taille moyenne des élèves-ingénieurs ayant obtenu C en TCS *analyse de données* ?

$\rightarrow$ quelle est la taille moyenne des élèves-ingénieurs ayant obtenu C en TCS *analyse de données*, A en *statistique*, E en *mathématiques I*, Fx en *mathématiques II* ?

La malédiction de la dimensionnalité : exemple 3

Problèmes de régression

On veut faire de l'interpolation sur $\mathbb{R}^D$:

$f(x_1, \dots, x_D) = w_0 + \sum_{i=1}^D w_i x_i$
$\rightarrow D + 1 = \mathcal{O}(D)$ paramètres à estimer

$f(x_1, \dots, x_D) = w_0 + \sum_{i=1}^D w_i x_i + \sum_{i=1}^D \sum_{j=i}^D w_{ij} x_i x_j$
$\rightarrow D + 1 + D(D+1)/2 = \mathcal{O}(D^2)$ paramètres à estimer

$f(x_1, \dots, x_D) =$
$w_0 + \sum_{i=1}^D w_i x_i + \sum_{i=1}^D \sum_{j=i}^D w_{ij} x_i x_j + \sum_{i=1}^D \sum_{j=i}^D \sum_{k=j}^D w_{ijk} x_i x_j x_k$
$\rightarrow \mathcal{O}(D^3)$ paramètres à estimer

Est-ce une approche réaliste ?

La malédiction de la dimensionnalité : exemple 4

Volume de l'hypersphère unité dans $\mathbb{R}^D$:

$$V_D = \frac{\pi^{D/2}}{\Gamma(D/2+1)} \underset{d \to \infty}{\sim} \frac{1}{\sqrt{\pi D}} \left(\frac{2\pi e}{D}\right)^{D/2} \xrightarrow[D \to \infty]{} 0$$

Par comparaison, le volume de l'hypercube circonscrit est : 2^D
distance du centre aux coins : $\sqrt{D}$

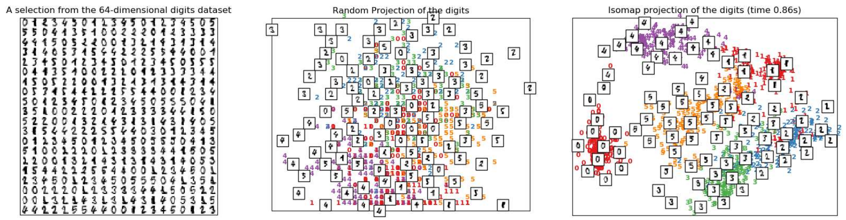
$\rightarrow$ concentration « dans les coins » du cube
$\rightarrow$ plus la dimension est grande, « plus il y a de place »

Exemple : l'image « la plus proche » d'une image de voiture est-elle une image de voiture ?

Solution ?

Heureusement, les observations (données) vivent souvent dans un sous-espace ou une variété de dimension beaucoup plus petite que l'espace ambiant.

« Solution » pratique : réduire la dimension
→ sélection de caractéristiques pertinentes (*feature selection*), analyse en composantes principales & co...



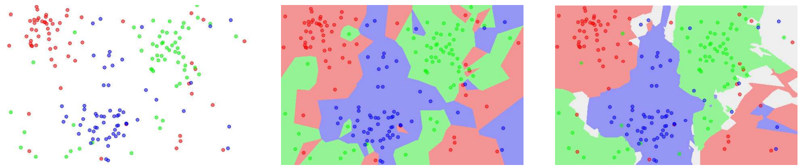
Source : `https://scikit-learn.org/stable/modules/manifold.html`

Problème : quel est ce sous-espace / cette variété ?

Problème 2 : dilemme biais-fluctuation

Cadre : apprentissage supervisé. On cherche à faire des prédictions

Question : a-t-on intérêt à avoir un modèle compliqué (précis), qui va bien représenter les observations (le jeu de données), ou un modèle simple (grossier) qui va moins bien représenter les observations mais moins en dépendre ?



dataset | 1 p.p.v. | 5 p.p.v.

→ tout le problème est qu'un prédicteur « appris » sur un autre jeu de données représentatif devrait donner des prédictions similaires

Erreur et risque de prédiction

Objectif : faire une prédiction y à partir d'une observation x.

Notations et hypothèses :

- base d'entraînement : $(x_i, y_i)_{1\leq i\leq N}$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{N}$ ou $\mathbb{R}^{d'}$
  réalisation d'un N-échantillon i.i.d. (X,Y) (loi **inconnue**)
- $\mathcal{H}$: famille de prédicteurs (*modèle*)
  Exemple : $\mathcal{H} = \{h : x \mapsto a \cdot x + b,\ (a,b) \in \mathbb{R}^d\}$
  $h(x)$: prédiction à partir de l'observation x
- coût d'une erreur : $\ell(y_i, h(x_i))$ (*loss function*)
  ex. régression : $\ell(y_i, h(x_i)) = (y_i - h(x_i))^2$
  ex. classification : $\ell(y_i, h(x_i)) = 0$ si $h(x_i) = y_i$ et $= 1$ sinon
- risque empirique : $R_e(h) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, h(x_i))$
  risque moyen de prédiction : $R_p(h) = E_{(X,Y)}(\ell(Y, h(X)))$ (**inconnu**)
  → idéalement, on voudrait trouver f dans $\mathcal{H}$ qui minimise R_p
  → on se contente de chercher $\widetilde{f}$ dans $\mathcal{H}$ qui minimise R_e

Encadrement du risque

On suppose que :
– f minimise sur $\mathcal{H}$ le risque de prédiction R_p
– $\widetilde{f}$ minimise sur $\mathcal{H}$ le risque empirique R_e
→ $\widetilde{f}$ « calculable », f idéal inconnu

Question : $R_p(\widetilde{f})$ est-il éloigné de $R_p(f)$?

Propriété

$$R_p(f) \leq R_p(\widetilde{f}) \leq R_p(f) + 2\max_{h\in\mathcal{H}} |R_p(h) - R_e(h)|$$

Preuve :
– par définition de f : $R_p(f) \leq R_p(\widetilde{f})$
– $R_p(\widetilde{f}) = R_p(\widetilde{f}) - R_e(\widetilde{f}) + R_e(\widetilde{f}) - R_e(f) + R_e(f) - R_p(f) + R_p(f)$
mais par définition de $\widetilde{f}$: $R_e(\widetilde{f}) - R_e(f) \leq 0$
et : $R_p(\widetilde{f}) - R_e(\widetilde{f}) + R_e(f) - R_p(f) \leq$
$|R_p(\widetilde{f}) - R_e(\widetilde{f})| + |R_p(f) - R_e(f)| \leq 2\max_{h\in\mathcal{H}} |R_p(h) - R_e(h)|$

Source : `http://www.di.ens.fr/~mallat/CoursCollege.html`

Dilemme biais-fluctuation (1) *(biais-variance)*

$$R_p(f) \ \leq R_p(\widetilde{f}) \ \leq R_p(f) + 2 \max_{h \in \mathcal{H}} |R_p(h) - R_e(h)|$$

On aimerait bien que $R_p(f)$ (biais) soit petit et que $R_p(\widetilde{f})$ ne soit pas trop éloigné de $R_p(f)$.

- $R_p(f)$: erreur d'approximation « idéale » : plus petite erreur moyenne pouvant être atteinte par un prédicteur de la famille $\mathcal{H}$
  $\rightarrow$ pour minimiser $R_p(f)$, on a intérêt à avoir un « gros » $\mathcal{H}$
- $\max_{h \in \mathcal{H}} |R_p(h) - R_e(h)|$: erreur de fluctuation sur $\mathcal{H}$ entre risque moyen de prédiction et risque empirique
  $\rightarrow$ à N fixé, plus $\mathcal{H}$ est « gros », plus la fluctuation est grande
  $\rightarrow$ comme $R_e(h)$ est un estimateur sans biais convergent de $R_p(h)$, on a intérêt à avoir N « grand »

Dilemme biais-fluctuation (2) *(biais-variance)*

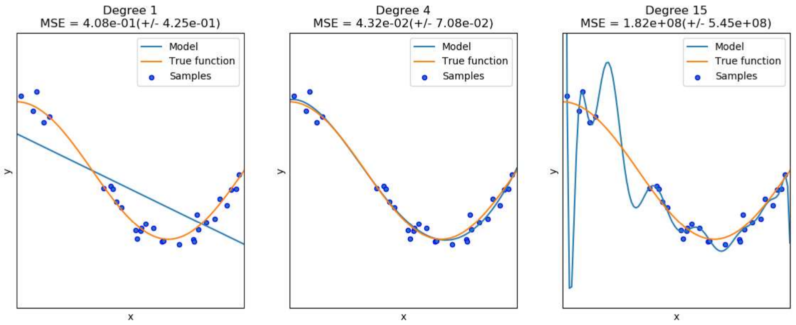
$$\underbrace{R_p(f)}_{\text{biais}} \ \leq R_p(\widetilde{f}) \ \leq \underbrace{R_p(f)}_{\text{biais}} + \underbrace{2 \max_{h \in \mathcal{H}} |R_p(h) - R_e(h)|}_{\text{fluctuation}}$$

Conséquences :

1. il faut trouver un compromis entre :
   - un modèle peu complexe ne pouvant pas bien expliquer les données ($\mathcal{H}$ trop restrictif), qui donnerait une fluctuation faible mais un biais grand
   - et un modèle très complexe ($\mathcal{H}$ gros) collant potentiellement bien aux données ($R_e(\widetilde{f})$ faible) tel que le biais est faible mais tel que $R_p(\widetilde{f})$ est potentiellement très éloigné de $R_p(f)$ (car fluctuation grande)
2. pour utiliser des modèles complexes, on a intérêt à avoir beaucoup de données
   car $\forall h$, $R_e(h) \rightarrow R_p(h)$ si $N \rightarrow +\infty$, donc fluctuation faible

Illustration (source : scikit-learn)

Régression par polynomes de degré d
$\rightarrow$ minimisation de R_e sur $\mathcal{H}_d = \{x \mapsto \sum_{i=0}^{d} a_i x^i\}$
remarque : $\mathcal{H}_1 \subset \mathcal{H}_4 \subset \mathcal{H}_{15}$



$\mathcal{H}_1$ $\mathcal{H}_4$ $\mathcal{H}_{15}$

sous-apprentissage *under-fitting*

sur-apprentissage *over-fitting*

Cf. rasoir d'Ockham, recherche de parcimonie

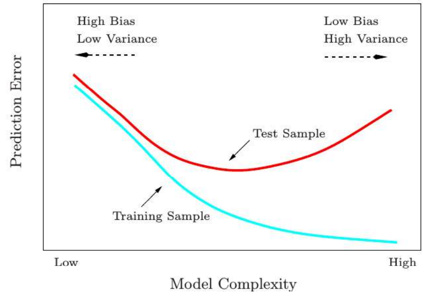
Comment sélectionner un modèle ?

Les paramètres d'un modèle sont estimés par minimisation du risque empirique. Comment choisir un modèle ? (ex : quel $\mathcal{H}_d$?)

Problème : le modèle minimisant l'erreur de prédiction sur la base d'entraînement (risque empirique) n'est pas le meilleur

$\rightarrow$ on va calculer l'erreur de prédiction sur une *base de test* représentative des observations, indépendante de la *base d'entraînement*

On observe :



Source : Hastie, Tibshirani, Friedman, *The elements of statistical learning*, 2008

Erreur d'apprentissage et erreur sur la base de test

Rappel : bases de test et d'observation sont supposées représentatives et indépendantes

On remarque :

- erreur d'apprentissage << erreur de test : **sur-apprentissage**
- erreur d'apprentissage $\simeq$ erreur de test, et erreurs « grandes » : **sous-apprentissage**
- erreur d'apprentissage $\simeq$ erreur de test, et erreurs « petites » : **OK**

En pratique avec un jeu de données $(x_i, y_i)_{1 \leq i \leq N}$

Approche 1 : on en met une partie de côté pour servir de base de test, c'est la validation *holdout*
Limite : fluctuation d'échantillonnage vs. taille base d'apprentissage

Approche 2 : pour exploiter au mieux le jeu de données, on peut faire de la validation croisée à K plis (K-fold cross validation)

on répète K fois : apprentissage sur $K-1$ plis, test sur K-ème pli

$K = 5$:

A	A	A	A	T	$\rightarrow$ erreur sur T : e_1
A	A	A	T	A	$\rightarrow$ erreur sur T : e_2
A	A	T	A	A	$\rightarrow$ erreur sur T : e_3
A	T	A	A	A	$\rightarrow$ erreur sur T : e_4
T	A	A	A	A	$\rightarrow$ erreur sur T : e_5

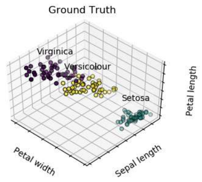
Erreur de validation croisée : moyenne des e_i

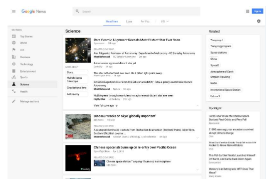
Cas particulier : $K = N$, leave-one-out cross validation (inconvénient : temps de calcul !)

Généralement, $K = 5$ ou $K = 10$.

Séance 1

Partitionnement / classification non supervisée





Observations : points dans $\mathbb{R}^d$, textes...

Distance / mesure de dissimilarité entre observations d(x,y) :
– L^1, L^2, L^∞...
– distance d'édition (de Levenshtein), voir polycopié (p. 13 et 76) :
`https://members.loria.fr/FSur/enseignement/RO/`

Mesure de dissimilarité entre classes (*cluster*) $D(\mathcal{A}, \mathcal{B})$:

$D(\mathcal{A}, \mathcal{B}) = \min\{d(x,y), x \in \mathcal{A}, y \in \mathcal{B}\}$ (*single linkage*)

$D(\mathcal{A}, \mathcal{B}) = \max\{d(x,y), x \in \mathcal{A}, y \in \mathcal{B}\}$ (*complete linkage*)

$$D(\mathcal{A}, \mathcal{B}) = \frac{n_\mathcal{A} n_\mathcal{B}}{n_\mathcal{A} + n_\mathcal{B}} ||m_\mathcal{A} - m_\mathcal{B}||^2 \quad (Ward)$$

$$= \sum_{x \in \mathcal{A} \cup \mathcal{B}} ||x - m_{\mathcal{A} \cup \mathcal{B}}||^2 - \sum_{x \in \mathcal{A}} ||x - m_\mathcal{A}||^2 - \sum_{x \in \mathcal{B}} ||x - m_\mathcal{B}||^2$$

Les classifications hiérarchiques : rappels

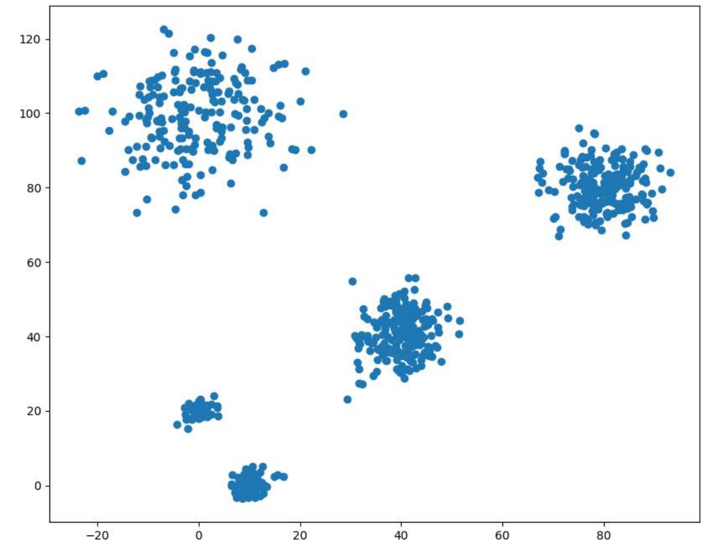
Algorithme :

1. initialisation : chaque observation dans une classe différente ;
2. jusqu'à ce qu'il ne reste qu'une classe, fusionner les deux plus proches au sens de D.

Sortie : le dendrogramme
= arbre binaire de classification, où hauteur des classes proportionnelle à dissimilarité des classes filles.

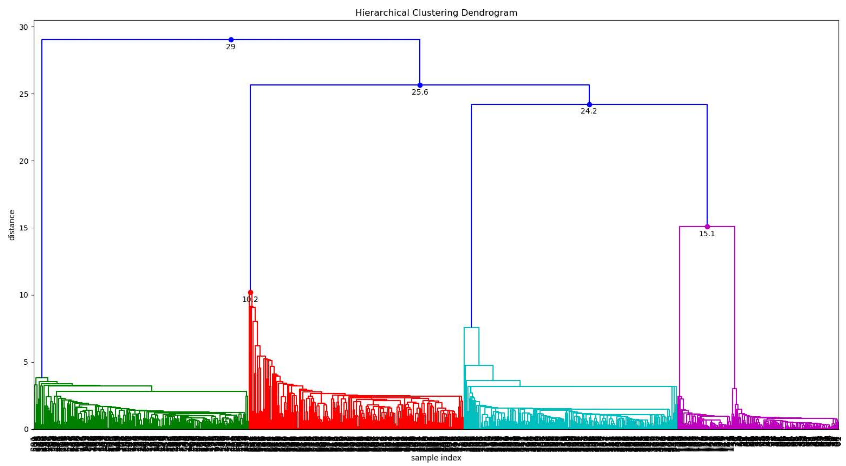
Classification : hauteur-seuil dans le dendrogramme.

Exemple 1 : données à partitionner

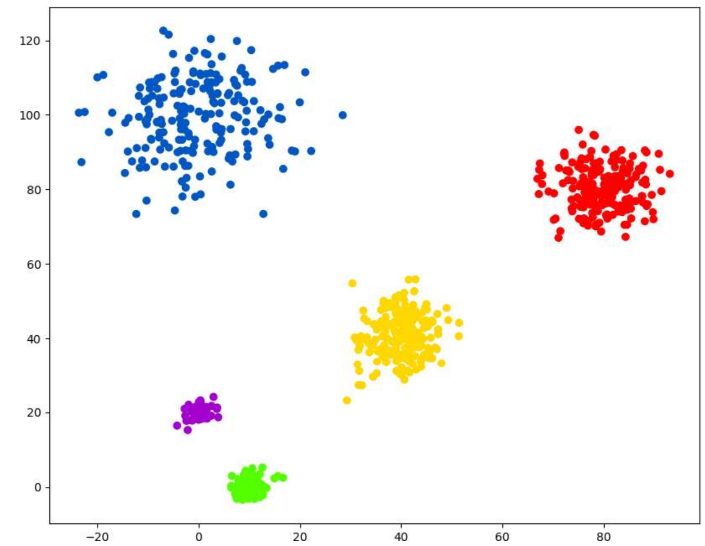


Source :
`https://joernhees.de/blog/2015/08/26/scipy-hierarchical-clustering-and-dendrogram-tutorial/`

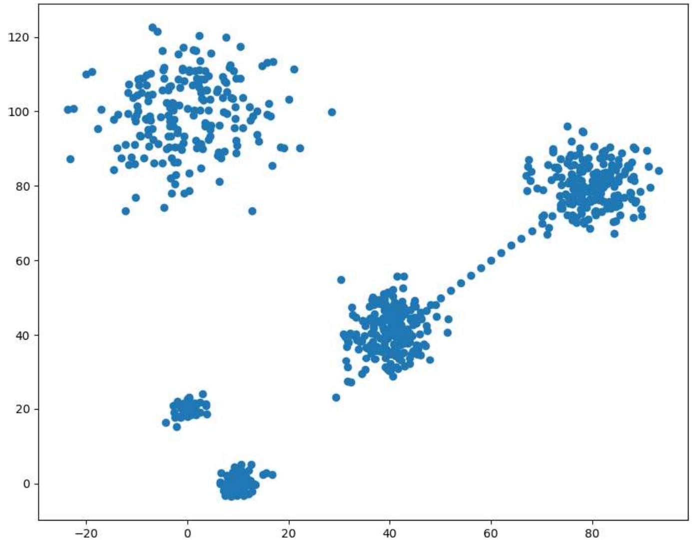
Exemple 1 : single-linkage



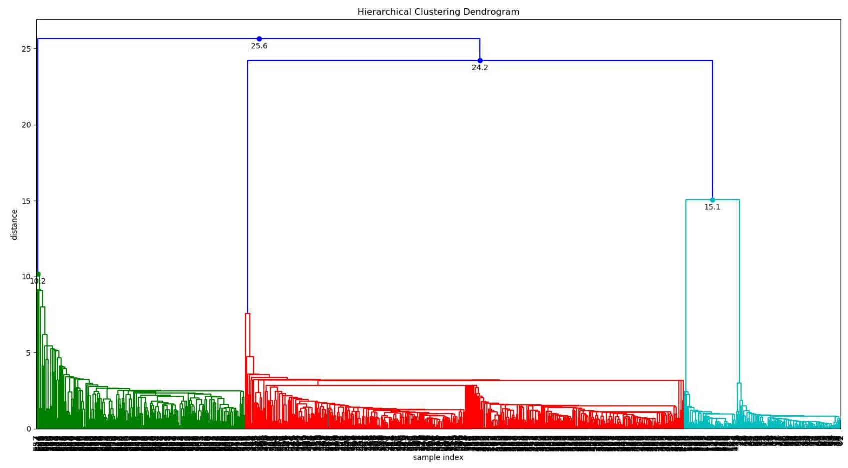
Exemple 1 : classification à 5 groupes



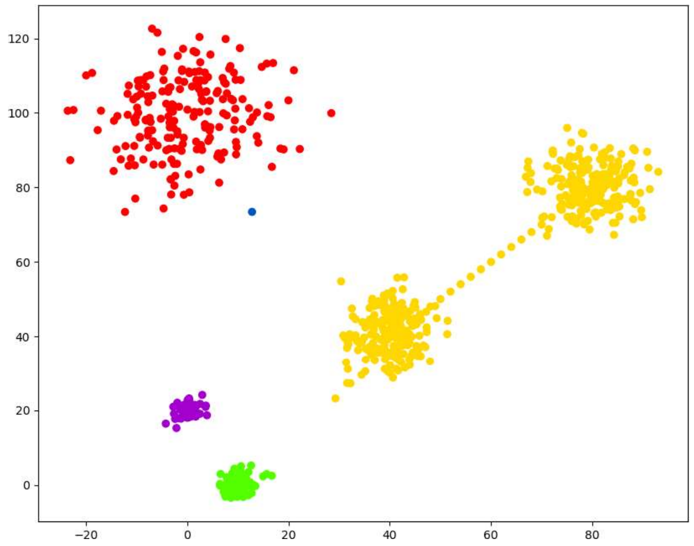
Exemple 2 : données à partitionner



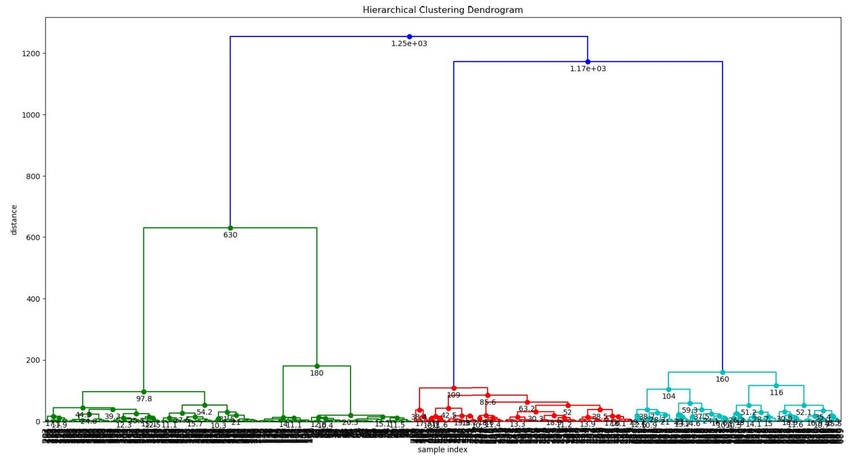
Exemple 2 : single-linkage



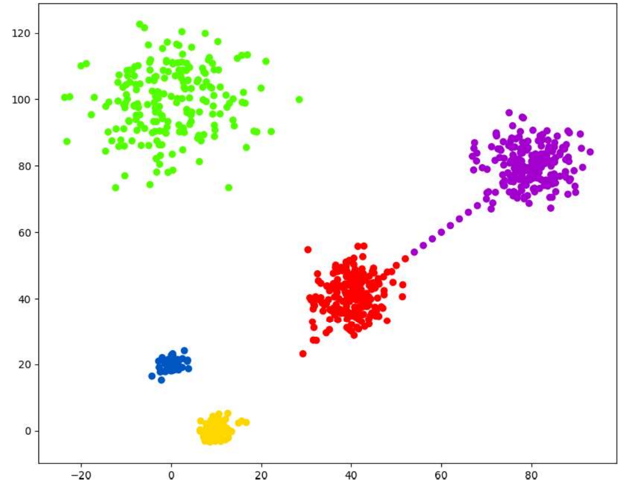
Exemple 2 : classification à 5 groupes



Exemple 2 : Ward



Exemple 2 : classification à 5 groupes



Discussion classification hiérarchique

- Quelle métrique de dissimilarité d entre observations ?
- Quelle métrique de dissimilarité D entre groupes ?
- Quel nombre de groupes ?
- Quels choix de métriques selon la distribution des observations ?
- Complexité algorithmique $\mathcal{O}(N^2 \log(N))$ (« lent »)
  Occupation mémoire $\mathcal{O}(N^2)$

K-moyennes (K-*means*)

On cherche une partition $(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K)$ de $(x_i)_{1 \le i \le N}$ minimisant

$$E(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K) = \sum_{j=1}^{K} \sum_{x \in \mathcal{C}_j} ||x - m_j||^2$$

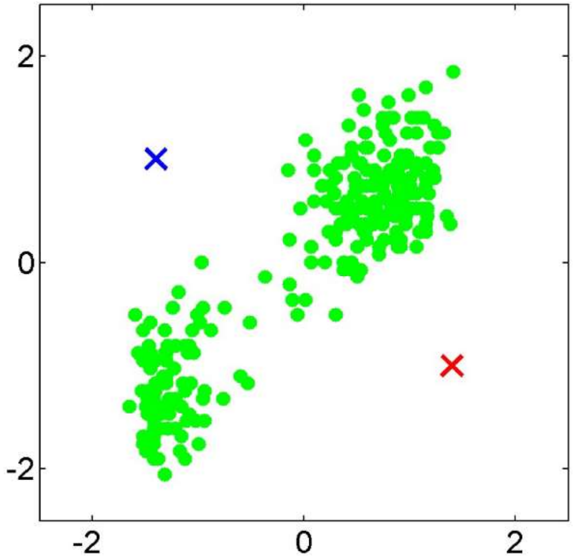
où m_j est la moyenne des $x \in \mathcal{C}_j$.
(E : *inertie* dans `sklearn`)

Algorithme (Lloyd) :
Initialisation : choix aléatoire de K « moyennes » m_j
Puis on itère :

- pour tout $1 \le j \le K$, on redéfinit $\mathcal{C}_j$ comme l'ensemble des x plus proche de m_j que des autres moyennes
- étant donnée une partition $(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K)$, on calcule les moyennes m_j

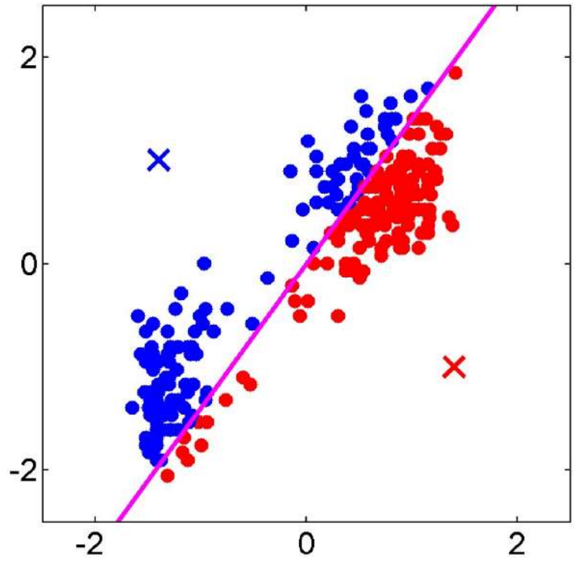
$\rightarrow$ on peut démontrer que cet algorithme permet la convergence en un nombre fini d'étapes vers un **minimum local** de E

Illustration de l'algorithme



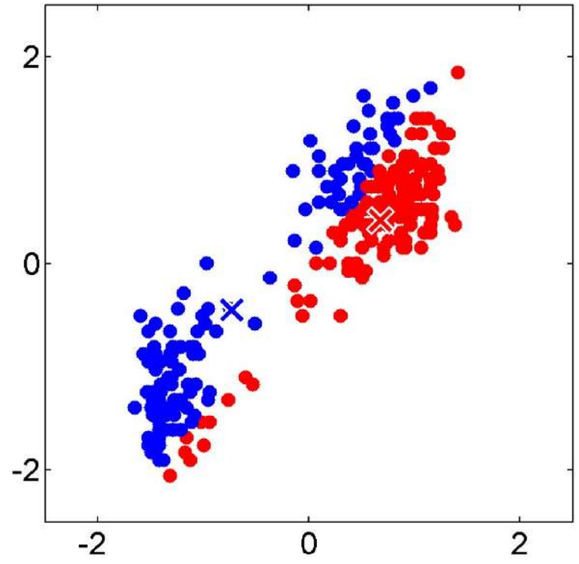
https://people.eecs.berkeley.edu/~jordan/courses/294-fall09/lectures/clustering/slides.pdf

Illustration de l'algorithme



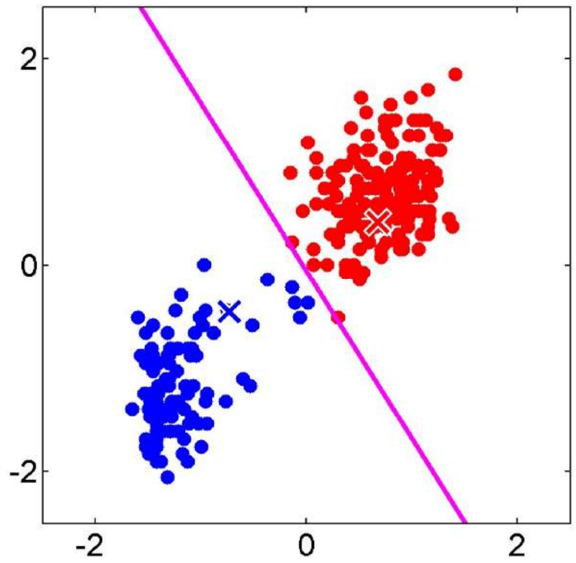
https://people.eecs.berkeley.edu/~jordan/courses/294-fall09/lectures/clustering/slides.pdf

Illustration de l'algorithme



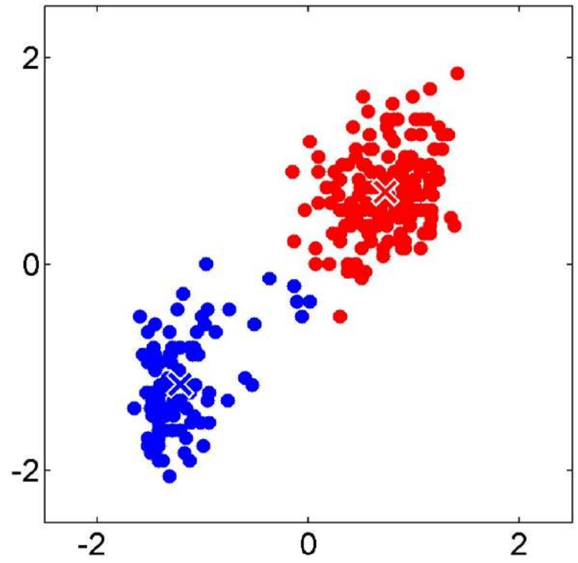
https://people.eecs.berkeley.edu/~jordan/courses/294-fall09/lectures/clustering/slides.pdf

Illustration de l'algorithme



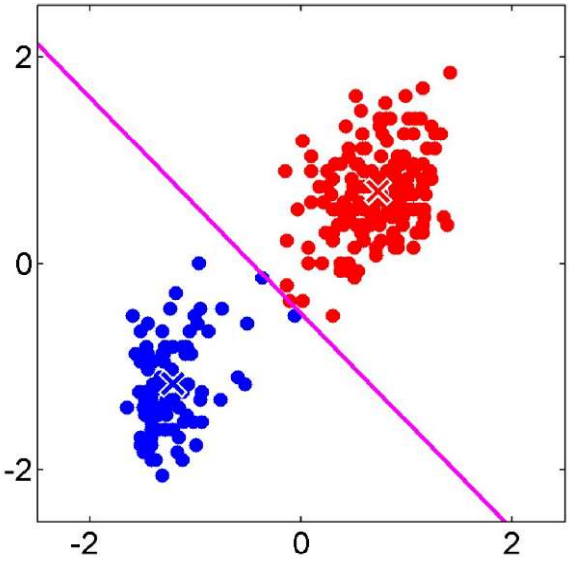
https://people.eecs.berkeley.edu/~jordan/courses/294-fall09/lectures/clustering/slides.pdf

Illustration de l'algorithme



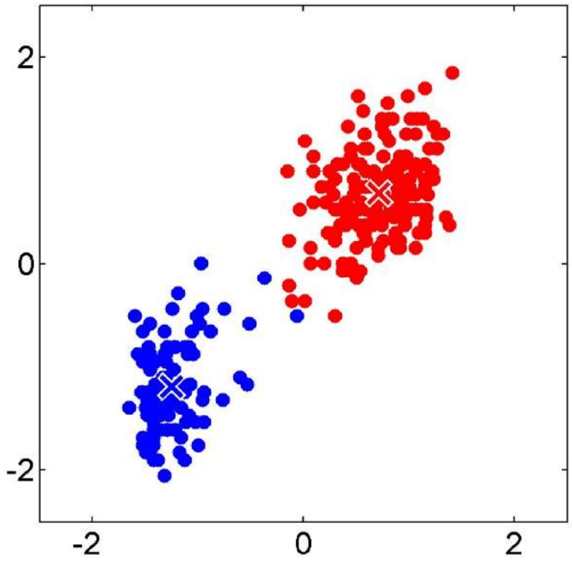
https://people.eecs.berkeley.edu/~jordan/courses/294-fall09/lectures/clustering/slides.pdf

Illustration de l'algorithme



https://people.eecs.berkeley.edu/~jordan/courses/294-fall09/lectures/clustering/slides.pdf

Illustration de l'algorithme



https://people.eecs.berkeley.edu/~jordan/courses/294-fall09/lectures/clustering/slides.pdf

Discussion K-moyennes

- Choix de K ?
  $\rightarrow$ variations du minimum de E en fonction de K ?
  (cf. *elbow plot*)
- Convergence vers un minimum local de E
  $\rightarrow$ plusieurs exécutions avec initialisations différentes
- Adapté à toute distribution des observations ?
- « Rapide » en pratique

Plan

Conclusion - Résumé

Difficultés fondamentales de l'apprentissage :

- malédiction de la dimensionnalité
- dilemme biais / variance,
  sous-apprentissage vs. sur-apprentissage

Sélection de modèle : l'outil de la validation croisée

En TP : problèmes de partitionnement