# IA et science des données

Cours 8 – mardi 18 mars 2025
Arbres (suite et fin). Apprentissage non supervisé

Christophe Marsala

Sorbonne Université

LU3IN026 - 2024-2025

## Plan du cours

Apprentissage par arbres de décision (suite)
- construction
- critère d'arrêt
- discrétisation
- frontière
- conclusion

Apprentissage non-supervisé



## Mesure de désordre moyen

- Utilisation de la forme conditionnelle de l'entropie de Shannon :
  - soit $\mathbf{X}_j$ un attribut ayant pour valeurs $v_{j1}, \ldots, v_{jr}$
  - et soit $\mathbf{Y}$ la classe ayant pour valeurs $y_1, \ldots, y_q$

$$H_S(\mathbf{Y}|\mathbf{X}_j) = -\sum_{l=1}^{r} p(v_{jl}) \sum_{k=1}^{q} p(y_k|v_{jl}) \log(p(y_k|v_{jl}))$$

- $H_S(\mathbf{Y}|\mathbf{X}_j)$ : pouvoir de discrimination de l'attribut $\mathbf{X}_j$ envers la classe $\mathbf{Y}$
  - $\mathbf{X}_j$ est discriminant pour $\mathbf{Y}$ si pour toute valeur $v$ de $\mathbf{X}_j$, la connaissance de la valeur $v$ permet d'en déduire une valeur unique $y$ de $\mathbf{Y}$





## Construction de l'arbre : algorithme classique (catégoriel)

- Créer une pile $\mathcal{P}$ et y stocker la base d'apprentissage
- Tant que $\mathcal{P}$ n'est pas vide : prendre l'ensemble $\mathcal{E}$ en haut de $\mathcal{P}$
  - calculer $H(\mathbf{Y})$ pour $\mathcal{E}$
  - si le critère d'arrêt est atteint alors créer une feuille
  - sinon, pour les exemples de $\mathcal{E}$
    1. calculer $H(\mathbf{Y}|\mathbf{X}_j)$ pour tous les attributs $\mathbf{X}_j$
    2. choisir l'attribut $\mathbf{X}_j$ qui maximise $I_S(\mathbf{X}_j, \mathbf{Y})$
       si aucun attribut n'apporte un gain $\rightarrow$ critère d'arrêt
    3. créer un nœud dans l'arbre de décision avec $\mathbf{X}_j$
    4. partitionner $\mathcal{E}$ en sous-ensembles avec les valeurs de $\mathbf{X}_j$
    5. mettre les sous-ensembles obtenus dans $\mathcal{P}$





## Critère d'arrêt de la construction de l'arbre

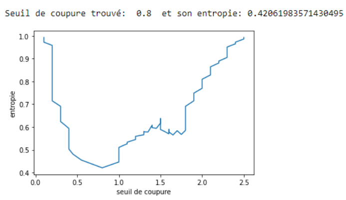
- Quelques exemples de critères d'arrêt
  - tous les exemples de la base d'apprentissage ont la même classe
  - utilisation d'une tolérance : la plupart des exemples ont la même classe
    - utilisation d'un seuil $\varepsilon \in [0, 1]$
      $\rightarrow$ arrêt si $H(\mathbf{Y}) \leq \varepsilon$
  - le gain d'information est nul ou négatif : $I_S(\mathbf{X}_j, \mathbf{Y}) \leq 0$
  - trop peu d'exemples dans l'ensemble traité
  - cas catégoriel : tous les attributs ont été utilisés une fois
- Création d'une feuille de l'arbre de décision
  - la classe majoritaire est utilisée pour étiqueter la feuille





## Traitement des attributs numériques

- $X_j$, attribut numérique
  - utilisation d'une valeur de coupure $v_j$
  - construction de 2 intervalles : $]-\infty, v_j[$ et $[v_j, +\infty[$
  - on note : $\{\mathbf{X}_j, v_j\}$ cette décomposition
- On détermine la valeur $v_j$ qui minimise $H(Y|\{\mathbf{X}_j, v_j\})$
  - phase de discrétisation : recherche exhaustive
  - on ensuite traite l'attribut comme un attribut catégoriel




Un même attribut numérique peut intervenir plusieurs fois dans l'arbre final avec des seuils de coupure différents

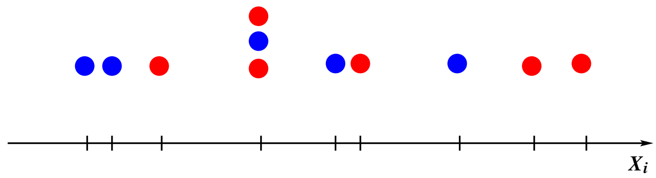
## Arbres de décision et données numériques

- Si $\mathbf{X}_j$ est un attribut numérique :
  - discrétiser $\mathbf{X}_j$ : le transformer en attribut catégoriel
    - le plus simple : discrétiser en 2 valeurs catégorielles
  - Idée : trouver une valeur de coupure $v_j$ pour $\mathbf{X}_j$
    - construction de 2 intervalles : $]-\infty, v_j[$ et $[v_j, +\infty[$
    - on note : $\{\mathbf{X}_j, v_j\}$ cette décomposition
  - trouver $v_j$ qui minimise $H(C|\{\mathbf{X}_j, v_j\})$
    - essayer toutes les valeurs possibles pour l'attribut
- Modification de l'algorithme de construction d'arbre
  1. phase de discrétisation de $\mathbf{X}_j$
  2. traiter $\mathbf{X}_j$ comme un attribut catégoriel : $\{\mathbf{X}_j, v_j\}$
- Deux possibilités
  - la discrétisation est faite avant la construction de l'arbre
  - la discrétisation est faite localement





## Arbres de décision et données numériques (2)

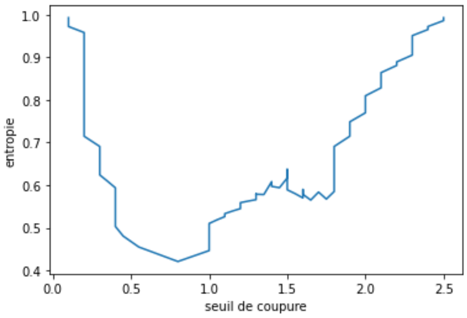







## Arbres de décision et données numériques (3)

```
Seuil de coupure trouvé:  0.8  et son entropie: 0.42061983571430495
```

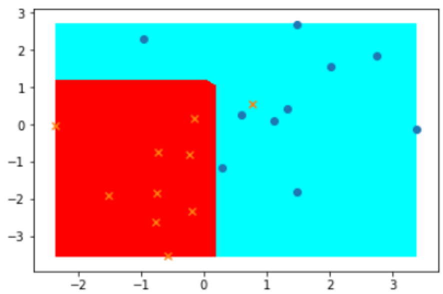







## Arbres de décision : frontière de décision (1)

- Exemple simple

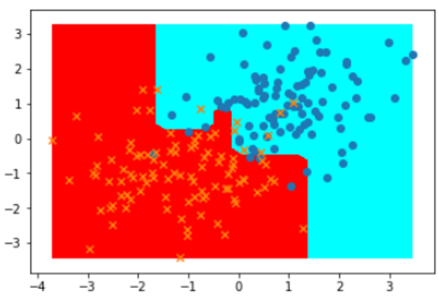






## Arbres de décision : frontière de décision (2)

- Exemple un peu moins simple







## Conclusion sur les arbres de décision

- Avantages
  - modèle d'apprentissage interprétable
    - mécanismes simples de construction
    - hiérarchie des attributs simple à comprendre
    - utilisation en classification
- Inconvénients
  - frontière construite par coupures perpendiculaires aux axes
    - pas de prise en compte de combinaisons d'attributs possibles
    - sous-apprentissage possible si le critère d'arrêt est trop lâche
    - sur-apprentissage si le critère d'arrêt est trop fort
  - lors de la construction
    - optimisation locale pour le choix d'un attribut

# Plan du cours

Apprentissage par arbres de décision (suite)

Apprentissage non-supervisé
- apprendre sans classe
- la tâche de clustering
- le clustering hiérarchique



# Exemple d'apprentissage supervisé : classifier des emails

- Objectif : construire un modèle pour détecter un spam
  - à partir d'un ensemble d'emails déjà reçus
- Description pour chaque email : valeurs d'attributs
  - statistiques sur son contenu
  - pourcentage de mots qui sont référencés comme caractéristiques de spam (par exemple : *money*,...), utilisation d'un dictionnaire
  - nombre moyen de lettres consécutives en majuscules
  - syllabes présentes, ...
- Base d'apprentissage : constituée d'emails déjà reçus
  - 4601 emails répartis en 2 catégories : 1813 spams, et 2788 non spams
  - Par exemple : spam (1) / non spam (-1)
    - 0,0.64,0.64,0.32,0,0,0,0,0,0,0.64,$\cdots$,0.778,0,3.756,61,278 $\longrightarrow$ 1
    - 0.49,0,0.49,0.49,0,0,0,0,0,0,0.99,$\cdots$,0.091,0,1.214,5,51 $\longrightarrow$ -1





# Exemple d'apprentissage non supervisé : groupes d'emails

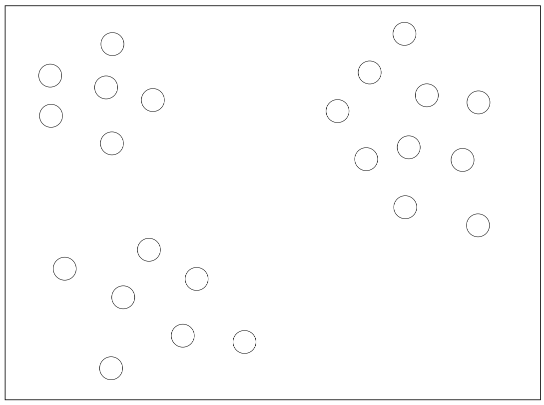
- Objectif : grouper des emails selon leur ressemblance
- Description pour chaque email : statistiques sur son contenu
  - pourcentage de mots pertinents (selon le domaine)
  - nombre moyen de lettres consécutives en majuscules
  - ...
- Aucune classe fournie
  - des classes seront mises en évidence à l'issue de l'apprentissage
- Par exemple :
  - email 1 : 0,0.64,0.64,0,0.32,0,0,0,0,0,0,0.64,0,0,0,0.32,0,1.29,1.93,0,0.96,0,0,0,0,0,0,0,0,0,0,0,0, 0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.778,0,0,3.756,61,278
  - email 2 : 0.49,0,0.49,0,0.49,0,0,0,0,0,0,0.99,0,0,0,0,0,0,0.99,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0.49,0,0,0,0,0,0,0,0,0.99,0.49,0,0,0,0,0,0,0.091,0,0,1.214,5,51





# Un petit exemple

- Un ensemble de données quelconque : combien de groupes ?







# L'apprentissage non supervisé

1. Phase d'apprentissage
   - construction d'un modèle
   - utilisation d'un ensemble de données d'apprentissage
     - créer des groupes homogènes selon les descriptions
     - mettre en évidence des ressemblances
   - évaluation du modèle obtenu
     - groupes homogènes ?
2. Phase de test
   - validation du modèle
   - utilisation sur un ensemble de données de référence
     - vérifier que les groupes restent homogènes
3. Phase d'utilisation
   - mise en œuvre du modèle
   - utilisation sur des données quelconques





# La classification en apprentissage non supervisé

- Classification : trouver des classes de descriptions
- Un ensemble de données sans classe connue
  - on recherche à faire des regroupements de descriptions similaires
  - on souhaite mettre en évidence des classes, des catégories
- But : former des groupes de données qui se ressemblent
  - clustering : faire des groupes parmi les données
  - cluster : ensemble de données regroupées ensemble
- Exemple :
  - le clustering hiérarchique
  - l'algorithme des $K$-moyennes

## Le clustering hiérarchique

- But : obtenir des groupes d'exemples
- Idée : grouper petit à petit les exemples qui se ressemblent
- Question : Comment mesurer la ressemblance entre 2 exemples ?
- On possède un espace de représentation des exemples
  - calculer des distances entre les exemples
  - deux exemples se ressemblent d'autant plus qu'ils sont proches
- Mesurer une distance : fonction d: $\mathbf{X}^d \times \mathbf{X}^d \rightarrow \mathbb{R}^+$
  - séparation : $\forall \mathbf{x}, \mathbf{y} \in \mathbf{X}^d$, d$(\mathbf{x}, \mathbf{y}) = 0$ ssi $\mathbf{x} = \mathbf{y}$
  - symétrie : $\forall \mathbf{x}, \mathbf{y} \in \mathbf{X}^d$, d$(\mathbf{x}, \mathbf{y})$ =d$(\mathbf{y}, \mathbf{x})$
  - inégalité triangulaire : $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{X}^d$, d$(\mathbf{x}, \mathbf{y})$+d$(\mathbf{y}, \mathbf{z}) \geq$d$(\mathbf{x}, \mathbf{z})$
- À connaître : distance ultramétrique
  - on remplace l'inégalité triangulaire par $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{X}^d$, $\max($d$(\mathbf{x}, \mathbf{y})$,d$(\mathbf{y}, \mathbf{z})) \geq$d$(\mathbf{x}, \mathbf{z})$





## Mesurer la distance entre 2 clusters

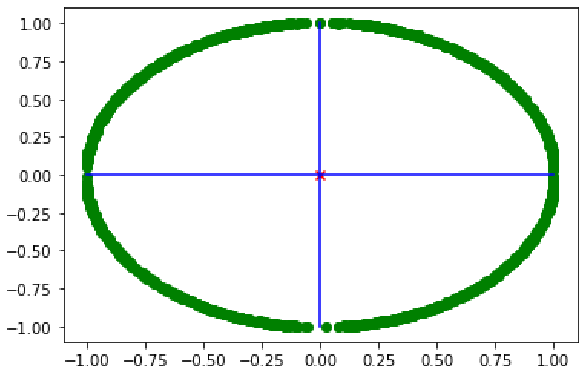
- Utiliser une distance entre 2 exemples : d$(\mathbf{x}_1, \mathbf{x}_2)$
  - Euclidienne, Manhattan, Minkowski, "infinie",...





## Distances et géométrie

- Distance euclidienne

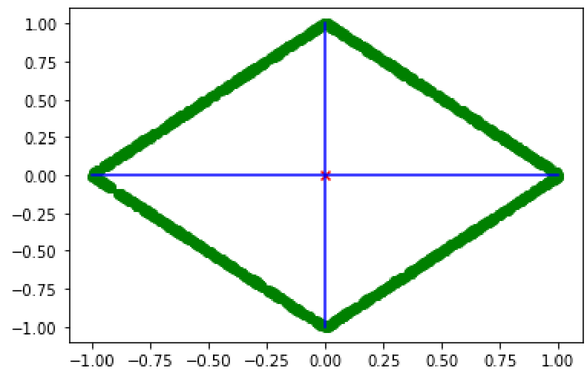







## Distances et géométrie

- Distance de Manhattan

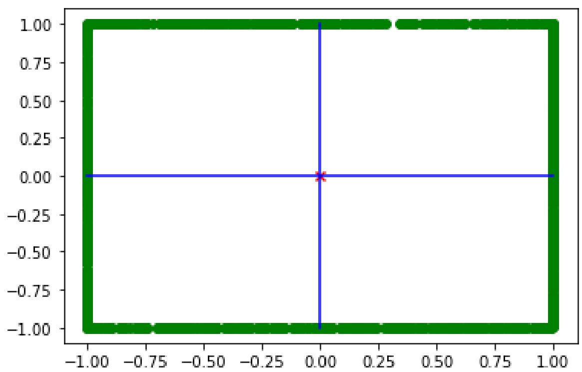







## Distances et géométrie

- Distance infinie

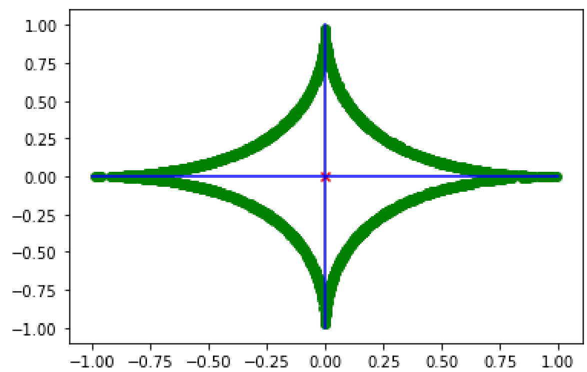







## Distances et géométrie

- Distance de Minkowski $(p = 0.5)$



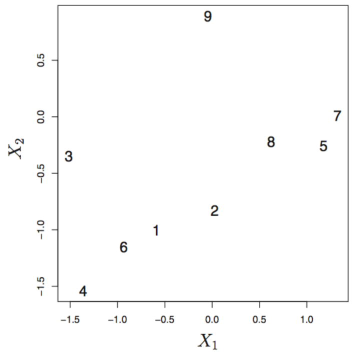
## Mesurer la distance entre 2 clusters

- Utiliser une distance entre 2 exemples : $\mathrm{d}(\mathbf{x}_1, \mathbf{x}_2)$
  - Euclidienne, Manhattan, Minkowski, "infinie",...
  - étape de normalisation nécessaire
- Distances entre 2 clusters $A$ et $B$ : $dist(A, B)$
  $A = \{\mathbf{x}_1^A, \mathbf{x}_2^A, \ldots, \mathbf{x}_{n_A}^A\}$ et $B = \{\mathbf{x}_1^B, \mathbf{x}_2^B, \ldots, \mathbf{x}_{n_B}^B\}$
  A) complete linkage
  B) average linkage
  C) simple linkage
  D) centroid linkage
- Centre de gravité (centroid) d'un cluster





## Exemple : méthode par agglomération




(source : "An introduction to statistical learning", G. James, D. Witten, T. Hastie, R. Tibshirani)





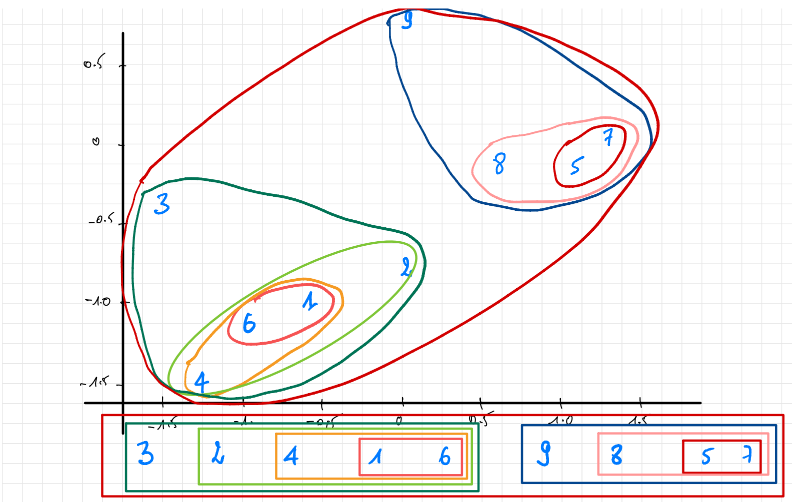
## Algorithme : clustering hiérarchique (version ascendante)

- Soit $\mathbb{E}$ un ensemble d'éléments (exemple ou groupe d'exemples)

1. calculer les distances entre chaque élément de l'ensemble
2. fusionner en un seul groupe les 2 éléments les plus proches : ce groupe remplace les 2 éléments dans l'ensemble $\mathbb{E}$
3. recommencer en 1) jusqu'à ce qu'il ne reste qu'un seul groupe unique dans $\mathbb{E}$

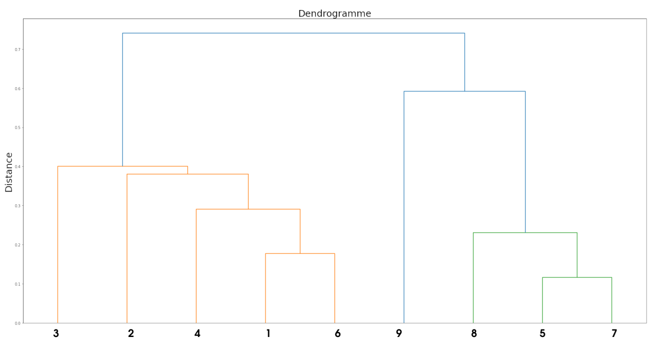
- Au départ : $\mathbb{E}$ est initialisé avec $\mathbf{X} \in \mathbb{R}^{n \times d}$
  - chaque exemple forme un groupe à lui tout seul
- Au final : $\mathbb{E}$ contient un groupe avec tous les exemples de $\mathbf{X}$
- Cet algorithme permet de construire un dendrogramme








## Exemple de dendrogramme final








## Conclusion sur le clustering hiérarchique

- Algorithme très efficace sur des jeux de données assez réduit, sinon ça devient vite peu lisible
- Le nombre de classes à trouver n'est pas défini : il est estimé par l'étude du dendrogramme
- Les calculs sont très coûteux ! ($\geq o(n^2)$)